

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

Département d'Automatique



Mémoire de fin d'études

En vue de l'obtention du diplôme d'Ingénieur d'Etat en Automatique

Thème :

**Étude et réalisation de la supervision de
l'alimentation du siège OTA
Utilisation de la solution SIEMENS :
STEP7 et WinCC**

Réalisé par :

AYAB Ahmed

KHIAR Ilyas

Soutenu devant le jury composé de:

Dr. O.STIHI **Président**

Mr. M.CHAKIR **Examineur**

Pr. E.M.BERKOUK **promoteur**

Promotion : **juin 2014**

ENP 10 Avenue Hacén Badi B.P 182 , El-Harrache 16200, Alger , ALGERIE

REMERCIEMENTS

Nous remercions Dieu le tout puissant qui nous a donné le courage et la volonté de réaliser ce travail.

Nous tenons à exprimer nos vifs remerciements à notre promoteur Pr.BERKOUK de l'Ecole Nationale Polytechnique pour nous avoir encadré durant notre projet de fin d'études et nous conseillé tout le long de notre travail.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre projet.

Nos sincères remerciements aux ingénieurs d'ORASCOM TELECOM ALGERIE qui nous ont conseillé et éclairé sur notre travail tout le long de notre projet.

Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Automatique qui nous ont encadrés auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.

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

Dédicaces

Je dédie ce modeste travail à mes chers parents qui ont fait de moi ce que je suis.

A mes frères et mes sœurs, pour qui j'ai toujours tenu à donner le meilleur de moi-même.

A ma grand-mère, qui m'a toujours soutenu.

A mon binôme Ilyas pour ces deux années de travail pleines de souvenirs, ainsi qu'a

toute sa famille que je remercie pour son hospitalité et sa gentillesse.

A mes amis Salah, Zaki, Anis, Hamada, Djalil, Zinou, Mohamed, Labib, Abdallah et Abdnour, qui ont toujours été là pour moi

A mes amis et camarades de l'Ecole Nationale Polytechnique, et toute la promotion Automatique de l'année 2014.

Aux anciens Automaticiens qui m'ont servis d'exemples

A tous ceux qui me sont chers, et qui me portent dans leurs cœurs.

AHMED

Dédicace :

dédions ce modeste mémoire à :

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère ...

A mon père, école de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.

Que dieu les gardes et les protège.

A nos deux sœurs et mon seul et unique frère «Billel».

A nos binôme : « Ahmed » pour son aide, son temps, son assistance et soutien.

A tous mes amis : Hamada, Zino,..

Au Pr. BERKOUK El Madjid qui, en tant que, nos professeur encadreur, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'il a bien voulu nous consacrer.

Aux Ingénieurs (Djezzy) : KHALDI A. Errahmen et MEZIANI Mostafa qui nous ont guidés sur le droit chemin tout au long du travail et nous ont aspiré les bons pas.

Sans leurs aides, ce travail n'aura pas abouti.

Aussi aux membres du jury pour l'honneur qu'ils nous ont fait en évaluant notre modeste travail.

Et à Tous ceux qui croient qu'

« *Il n'est jamais trop tard pour mieux faire.* »

A ceux qui ont attendu avec patience les fruits de leurs efforts : nos professeurs.

Et finalement, à tout ce qui ont contribué de près ou de loin à la réalisation de ce travail.

Merci infiniment.

Ilyas

ملخص:

ويستند العمل المقدم في هذه المذكرة أساسا على استخدام الآلي. SIEMENS PLC, عملنا هو دراسة و أنجاز نظام رصد و تحكم في تزويد مقر OTA. باستخدام برنامج البرمجة STEP7، وبرنامج تصميم واجهات المستخدم الرسومية WinCC مرن.
الكلمات المفتاحية : SIEMENS PLC؛ STEP7، WinCC، الإشرافية، التنبيه.

RESUME :

Le travail présenté dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables SIEMENS. Notre travail est l'étude et la réalisation d'un système de supervision de l'alimentation du siège OTA, par l'utilisation du logiciel de programmation STEP7, et le logiciel de conception des interfaces graphiques WinCC flexible.

Mots clés : automates programmables SIEMENS ; STEP 7 ; WinCC, supervision, alarme.

Abstract :

The work presented in this project is essentially based on the utilization of SIEMENS Programmable Logic Controllers. Our work is the study and the realization of a supervisory system for the OTA station, using the programming software STEP7 and the graphical interfaces conceptor WinCC flexible.

Keywords : Programmable Logic Controllers, SEIMENS, STEP7, WinCC, supervision, Alarm.

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I: Les automates programmables industriels

I.1.Introduction .	3
I.2.Définitions générales .	3
I.3.Architecture générale des API	4
I.3.1. Aspect extérieur	4
I.3.2. Structure interne	5
I.3.2.1. Unité centrale	5
I.3.2.2. Les modules d'entrées/sorties	6
I.3.2.3. Module d'alimentation	7
I.3.2.4. Modules de communication	7
I.3.2.5. Les auxiliaires	7
I.4. Traitement du programme dans l'automate	8
I.5. Les langages de programmation	8
I.5 .1 . Les langages graphiques	8
I.5.1.1.Le GRAFCET (SFC)	8
I. 5.1.2 Le Langage FBD	10
I. 5.1.3 Le langage LD	11
I. 5 .2 . Les langages textuels	12
5.2.2. Le langage IL	13
I. 6.Présentation de quelques gammes SIMATIC	14
I.6.1.SIMATIC S7	14
I .6. 2. SIMATIC M7	15
I .6. 3 .SIMATIC C7	16
I. 7. Critères de choix d'un automate programmable	16
I. 8 .Description de l'automate S7- 314C-2 DP	17
I.8 .1. Module d'alimentation	17
I. 8 .2. La CPU	18

I. 8.3. Le module E/S digitales intégré (DI24/DO16)	23
I. 8.4. Le module E/S analogiques intégré (AI5/AO2)	25
I.10. Conclusion	28

Chapitre II: STEP7, logiciel de programmation des API Siemens

II.1.INTRODUCTION	29
II.2.Présentation du logiciel STEP	29
II.2.1.Gestionnaire de projets SIMATIC Manager	29
II.2.2.Editeur de la configuration matérielle	29
II.2.3.Editeurs de programmes des Différents langages de programmation	29
II.2.5.Paramétrage de l'interface PG/PC	30
II.2.6.Le simulateur de programmes PLCSIM	30
II.3.Création d'un projet SIMATIC Manager	30
II.3.1.Configuration matérielle	32
II.3.2 Edition de la table des mnémoniques	38
II.3.3.Edition des programmes	40
II.4 .Différents langages disponibles dans le STEP7	43
II.4.1.CONT	44
II.4.2.LIST	45
II.4.3.LOG	45
II.5.Simulation des modules	46
II.5.1.Ouverture de l'application PLCSIM	46
II.5.2.Chargement du programme	46
II.5.3.Configuration du PLCSIM	46
II.5.4. Exécution du programme	47
II.5.5.Etat de fonctionnement de la CPU	48
II.6. Conclusion :	48

Chapitre III: Le logiciel de conception des interfaces Homme/Machine WinCC flexible

III .1. Introduction	49
III .2. Présentation de WinCC	49
III .2.1. Description générale :	49
III .2.1. Fonctionnement de WinCC flexible	50
III .2.2. Eléments de l'interface utilisateur de WinCC flexible	50
III .3. Etapes de conception d'un projet sous WinCC flexible	52
III .3.1. Intégration du projet WINCC flexible à STEP7 managé	52
III .3. 2. Création de la liaison entre le projet HMI et l'API	52
III .3. 3. Définition des variables	53
III .3. 4. Création des vues	54
III .3. 4.1. Les objets de l'éditeur "Vues"	54
III .3. 5. Gestion des alarmes	55
III .3. 5.1. Classes d'alarmes prédéfinies dans WinCC flexible	55
III .3. 5.2. Éditeur "Alarmes TOR"	55
III .3. 5.3. Éditeur "Alarmes Analogiques"	57
III .3. 6 Configuration de la vue courbe d'une variable analogique	58
III .3. 7 Editeur de "Scripts"	60
III .3. 7.1 : Accès aux variables	60
III .3. 7.2. Utilisation des boucles	61
III .3. 8. Editeur de "Recettes"	62
III .3. 9. Archivage des données dans WinCC flexible	65
III .3.9.1. Archive variable	65
III .3.9.2. Archive Alarmes	66
III.3.10. Fonctions système en Runtime	67
III .4 . Simulation du projet sur WinCC flexible Runtime	68
III .5. Conclusion	70

Chapitre IV : application

IV.1. Introduction	71
IV.2. Description de la station télécommunication mobile (Switch) OTA	71
IV.2. 1. Les sources d'alimentation du Switch	72
IV.2. 1. 2. Le Réseau Secours(Le Groupe Electrogène)	72
IV.2. 2. Les équipements de la salle d'énergie	73
IV.2. 2.1. L'inverseur ATS/MTS	73
IV.2. 2.2 . Le tableau générale basse tension (TGBT)	74
IV.2. 2.3 . Les redresseurs	74
IV.2. 2.4 .Les alimentation sans interruption UPS (onduleur)	75
IV.2. 2.5 . La climatisation	76
IV.2. 3 . Les équipements de télécommunication	76
IV.2. 4 .La supervision actuelle de la station OTA	76
IV.3. Développement du projet de gestion et supervision de l'alimentation du siège OTA	77
IV.3. 1. Définition du cahier des charges	78
IV.3. 1.1 .La partie supervision	79
IV.3. 1. 2. La commande	80
IV.3. 2. Définition des entrées/sorties	80
IV.3. 3. Le choix d'automate	80
IV.3. 4. Création du projet STEP 7	81
IV.3. 4.1 Configuration du matériel dans STEP7	81
IV.3. 4. 2 Ecriture du programme	82
IV.3. 5. Configuration de la communication entre le pupitre et l'automate S7 314C-2 DP	85
IV.3. 6. Création de l'interface homme machine HMI du projet par WinCC flexible	86
IV.3. 6.1 Elaboration de la supervision	86
IV.3. 6.2 Description des vues	86
IV.3. 6.3 Amélioration optionnel	94
IV.3. 7. Instrumentation	94
IV.3. 7.1 .Récupération et branchement des entrées TOR	95
IV.3. 7.2 . Les capteurs analogiques	96

IV.5. Conclusion 100

Bibliographie

Annexes

Sommaire des figures

Figure I. 1. Automate modulaire (Siemens). Figure I. 2. Automate modulaire (Modicon).	4
Figure I. 3. Automate compact (Allen-bradley)	5
Figure I. 4. Structure interne d'un API.	5
Figure I. 5. Traitement du programme dans l'automate.	8
Figure I. 6. Le GRAFCET.	10
Figure I. 7. Exemple d'un programme FBD	11
Figure I. 8: Exemple d'un programme LD	11
Figure I. 9. Exemple d'un programme ST.	12
Figure I. 10. Exemple d'un programme IL.	13
Figure I. 11. Automate S7-200	14
Figure I. 12. Automate S7-300	14
Figure I. 13. Automate S7-400	15
Figure I. 14. Automate M7.	15
Figure I. 15. Automate C7.	16
Figure I. 16. Vue générale de l'automate S7- 314C-2 DP.	17
Figure I. 17. LED de visualisation d'état de la CPU.	18
Figure II. 1 Création d'un nouveau projet (1).....	30
Figure II. 2 Création d'un nouveau projet (2).....	31
Figure II. 3. Insertion d'une station de travail	31
Figure II. 4. Choix de la station SIMATIC.	32
Figure II. 5. Editeur des configurations matérielles.....	32
Figure II. 6. Catalogue du matériel.	33
Figure II. 7. Catalogue SIMATIC 300.	33
Figure II. 8. Sélection des modules (1).	34
Figure II. 9. Sélection des modules (2).	35
Figure II. 10. Paramétrage de la CPU (1).	35
Figure II. 11. Paramétrage de la CPU (2).	36

Figure II. 12. Paramétrage du module E/S TOR.	37
Figure II. 13. Paramétrage du module E/S analogiques.	37
Figure II. 14. Editeur des mnémoniques.	38
Figure II. 15. Editeur de programme CONT, LIST, LOG.	44
Figure II. 16. Exemple d'un programme en CONT.	44
Figure II. 17. Exemple d'un programme en LIST.	45
Figure II. 18. Exemple d'un programme.	45
Figure II. 19. Simulation des modules.	47
Figure III.1. Communication entre le WinCC, l'API et le processus.	49
Figure III. 2. Eléments de l'interface de WinCC flexible.	51
Figure III. 3. Intégration du projet WINCC flexible à STEP7 manager	52
Figure III. 4. Liaison entre le pupitre et l'automate.	53
Figure III. 5. Éditeur de variable dans le Win CC flexible.	54
Figure III. 6. Création d'une vue.	54
Figure III. 7. Configuration des alarmes TOR (1).	56
Figure III. 8. Configuration des alarmes TOR (2).	56
Figure III. 9. Configuration des alarmes TOR (3).	57
Figure III. 10. Configuration des alarmes analogiques (1).	58
Figure III. 11. Insertion et configuration d'une vue courbe.	59
Figure III. 12. Editeur du script (1).	60
Figure III. 13. Editeur de script(2)	61
Figure III. 14. Editeur de script(3)	61
Figure III. 15. Eléments de la recette.	62
Figure III. 16. Les enregistrements de la recette.....	63
Figure III. 17: Vue de recette (étendue).....	63
Figure III. 18. Editeur archive variable.	66
Figure III. 19. Editeur archives alarmes (1).	67
Figure III. 20. Editeur archives alarmes (2).	67
Figure III. 21. Configuration d'une fonction système.	68
Figure III. 22. Simulation sans raccordement à l'automate.....	69

Figure III. 23. Paramétrage du pupitre.	70
Figure IV. 1. Schéma synoptique de l'alimentation du siège OTA]	71
Figure IV. 2. Poste transformateur.	72
Figure IV. 3. Le groupe électrogène.	73
Figure IV. 4. L'inverseur de source ATS/MTS.	74
Figure IV. 5. Schéma installation des redresseurs.	75
Figure IV. 6. Schéma Installation des onduleurs.	75
Figure IV. 7. Texte d'alarme de la supervision actuelle.	76
Figure IV. 8. L'organigramme du développement du projet.	78
Figure IV. 9. Paramétrage E/S Analogique.	81
Figure IV. 10. Programme pour les entrées TOR.	82
Figure IV. 11. Traitement des entrées analogiques dans l'automate.	83
Figure IV. 12. La fonction SCALE.	83
Figure IV. 13. Programme pour alarmes du dépassement de seuil.	84
Figure IV. 14. Programme du test démarrage GE.	85
Figure IV. 15. La liaison entre l'automate et le pupitre.	85
Figure IV. 16. Paramétrage de l'interface PG /PC.	86
Figure IV. 17. Vue principale de l'interface graphique.	87
Figure IV. 18. Vue poste transformateur	87
Figure IV. 19. Vue groupe électrogène	88
Figure IV. 20. Vue commutateur "ATS/MTS"	88
Figure IV. 21. Vue redresseurs	89
Figure IV. 22. Vue onduleurs	89
Figure IV. 23. Vue climatisation	90
Figure IV. 24. Vue courbes courant	90
Figure IV. 25. Vue température	91
Figure IV. 26. Vue alarmes instantanées	91
Figure IV. 27. Vue archives	92
Figure IV. 28. Vue archive courants	92
Figure IV. 29. Vue archive température	93

Figure IV. 30. Vue archive alarmes.	93
Figure IV. 31. Protection de la commande par mot de passe	94
Figure IV. 32. État du contacte sec pour les alarmes.	95
Figure IV. 33. Branchement des entrées TOR.	95
Figure IV. 34. Transformateur de courant.	96
Figure IV. 35. Branchement du capteur courant à l'automate.	97
Figure IV. 36. Sonde de température TB/TS à thermistance.	98
Figure IV. 37 .Branchement de la sortie commande GE.	98
Figure IV. 38. Automate S7-314	99
Figure IV. 39. Test des entrées analogiques.	99

Sommaire des Tableau

Tableau I. 1. Les modules d'alimentations suivantes la gamme S7-300.	18
Tableau I. 2. Positions du commutateur du mode de fonctionnement.	19
Tableau I. 3. Caractéristiques techniques de La CPU S7-314C-2DP.	23
Tableau I. 4. Caractéristiques techniques du module E/S digitale intégré.	24
Tableau I. 5. Caractéristiques techniques du module E/S analogique.	27
Tableau I. 6. Correspondance valeurs analogiques tension.	27
Tableau I. 7. Correspondance valeurs analogiques courant.	28
Tableau II. 1. Les zones mémoires.	39
Tableau II. 2. Accès aux différentes zones mémoire.	40
Tableau II. 3. Les blocs d'organisation.	42
Tableau III. 1. Boutons commande de la vue recette.	64

Symboles et abréviation

API : Automate Programmable Industriel
ATS : Automatique Transfer Switch
CPU : Central Processing Unit
EPROM : Erasable Programmable Read Only Memory
FB : Blocs fonctionnels
FBD : Fonction Bloc Diagram
FC : Fonctions
GE : Groupe électrogène
HMI : Interface Homme Machine
IL : Instruction List
LD : Ladder Diagram
LIST : Le langage de liste d'instructions
LOG : Le langage a base de logigramme
MPI : Multi Point Interface
MT/BT : Moyenne tension /Basse tension
MTS : Manuel Transfer Switch
NE : Network équipements
OB :Blocs d'organisation
OP : Pupitre opérateur
OTA : Orascom Telecom Algerie
PG : La console de programmation sur le terrain
PID: Proportional Integral Derivative
PLC :Programmable logic controllers
RAM : Random Access Memory
ROM : Read Only Memory
SFB : Bloc fonctionnel système
SFC : Bloc programme pour le langage évolué textuel
ST : Structured Text
TOR : Toute ou rien
UAL :Unité Arithmétique et Logique
VB: Visual Basic
VBS: Visual Basic Script

Introduction générale :

Dans un contexte de compétitivité et de pérennité, les entreprises du secteur de télécommunication sont obligées d'automatiser de plus en plus leur fonctionnement. La supervision de l'alimentation de l'énergie entre dans ce contexte. Cette supervision est une forme évoluée du dialogue Homme-Machine, qui consiste à surveiller l'état de fonctionnement d'un procédé dont les possibilités vont bien au delà de celles des fonctions de conduite, et de surveillance réalisée avec les interfaces.

L'entreprise ORASCOM TELECOM Algérie n'échappe pas à cette réalité, et se voit dans l'obligation de superviser ses stations en utilisant des automates programmables. Ceci se fait dans un but d'améliorer la gestion de ses ressources, de réduire les délais et d'optimiser ses dépenses.

C'est dans cette conjoncture que ORASCOM TELECOM de Dar El Beida, nous a proposé de mener une étude visant à développer une solution à base d'automates programmables dans le but de superviser la gestion de l'alimentation en énergie.

A cet effet, l'entreprise nous a remis un cahier des charges spécifique dont il fallait respecter les exigences, et sur la base duquel notre travail sera validé.

Plan du travail

L'étude qui a été menée, se décompose en quatre chapitres.

Il a été question dans le premier chapitre de présenter l'automate programmable industriel en décrivant son architecture et en citant les différents langages qui servent à le programmer. Cet automate programmable est l'un des appareils de commande des systèmes de production et d'automatisme les plus utilisés dans l'industrie.

Le deuxième chapitre a introduit un outil de programmation des automates programmables industriels, à savoir, le STEP7 qui permet la programmation et la simulation de procédures automatisées avec différents langages normalisés dont il en a été question dans le premier chapitre.

Le chapitre III a servi à présenter le logiciel de conception des interfaces Homme-Machine appelé « SIMATIC WinCC ». Ce dernier est un système de supervision de processus modulables qui offre des fonctions performantes de surveillance des automatismes.

Le chapitre IV a détaillé le projet, l'environnement dans lequel il doit être appliqué ainsi que le cahier des charges auquel il devait répondre. Après avoir réalisé des tests réels visant à valider le projet, nous avons, en dernière partie, proposé des solutions qui permettent de respecter le cahier des charges qui nous a été imposé.

Enfin, et en guise de conclusion générale, nous ferons une rétrospective de ce qui a été réalisé, et nous proposerons quelques axes d'amélioration.

CHAPITRE I :

Les automates programmables industriels

I.1.Introduction :

L'automate programmable industriel (PLC **Programmable logic controllers**) est l'un des appareils de commande des systèmes de production et d'automatisme, les plus utilisés dans l'industrie. Son apparition pour la première fois fut aux Etats-Unis dans le secteur de l'industrie automobile. Ce dernier était commandé par des systèmes d'armoires de commande à relais, qui sont complexes et difficiles à modifier.

L'industrie automobile a donc trouvé dans les PLC un moyen redoutable, moins coûteux et qui est surtout le plus facile à modifier pour remplacer ces armoires de commande. Dès lors, ils se sont répandus très rapidement dans toutes les autres industries, au point de représenter aujourd'hui l'un des appareils de commande occupant une place très importante dans les systèmes automatisés de production.

I.2.Définitions générales :

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

Un système est considéré comme automate programmable si :

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

Ceci nous amène à une définition assez lourde qui est celle de la norme européenne EN 61131-1 :

« Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en œuvre de fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties Tout-ou-Rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues ». [1]

I.3. Architecture générale des API :

L'architecture du processeur d'un automate programmable est fondamentalement la même que celle d'un ordinateur à usage général, néanmoins, il existe certaines caractéristiques très importantes qui les distinguent. Tout d'abord, contrairement aux ordinateurs, les automates programmables sont conçus pour résister aux conditions difficiles de l'environnement industriel. Un automate bien conçu peut être placé dans une zone avec d'importantes quantités: de bruit électrique, des interférences électromagnétiques, des vibrations et d'humidité sans condensation. Une deuxième distinction, est perceptible au niveau des matériels et logiciels des automates, qui sont conçus pour une utilisation ciblée et simple. Afin de faciliter l'usage de l'automate par les techniciens et par les opérateurs. Aussi ses interfaces de connexions font parties intégrantes de l'automate lui-même de ce fait il facilite sa communication aux appareils de mesure qui lui sont connectés. Cette architecture peut être divisée selon deux aspects [2].

I.3.1. Aspect extérieur :

Les automates peuvent être, suivant leur aspect extérieur, soit de type **compact** ou **modulaire**.

Le type **compact**, intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande des petits automatismes. Dans ce type on distingue les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des micro-automates.

Dans le type **modulaire**, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes, puissants, où la capacité de traitement et la flexibilité sont nécessaires [2]

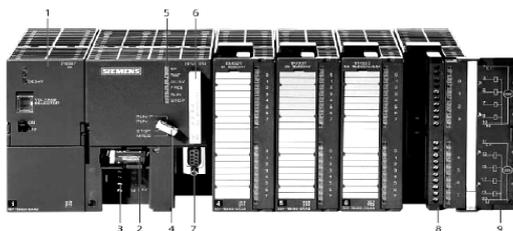


Figure I. 1. Automate modulaire (Siemens). **Figure I. 2.** Automate modulaire (Modicon).



Figure I. 3. Automate compact (Allen-bradley)

I.3.2. Structure interne :

En général, un automate programmable se constitue essentiellement d'une unité centrale, un module d'entrées/sorties, un module d'alimentation, un module de stockage, de liaisons et d'auxiliaires [3].

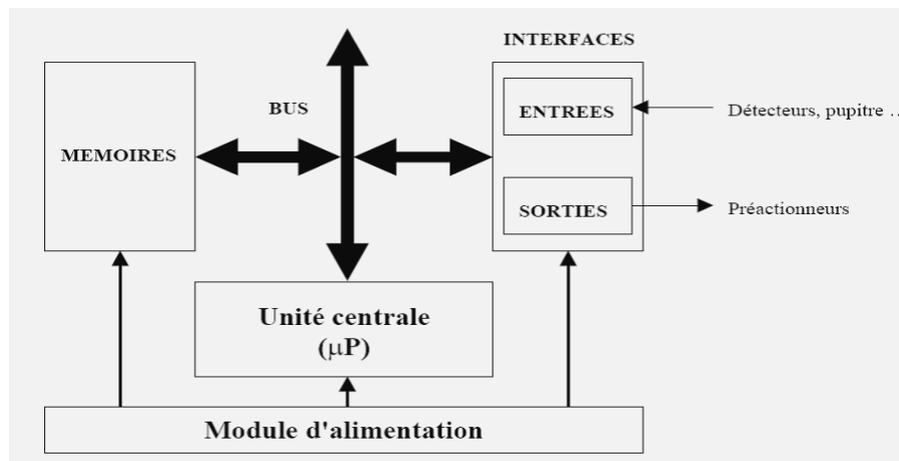


Figure I. 4. Structure interne d'un API.

I.3.2.1. Unité centrale :

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

- **Le processeur**

Le processeur est chargé d'exécuter le programme utilisateur, il doit assurer des opérations logiques et arithmétiques ainsi que des fonctions de temporisation et de comptage. Il peut être issu de la technologie câblée ou de la technologie à microprocesseur.

Il assure :

- La lecture des informations d'entrée
- L'exécution des instructions du programme mis en mémoire
- La commande ou l'écriture des sorties

En général, un microprocesseur est composé d'une Unité Arithmétique et Logique (UAL), de registres, un décodeur d'instructions, un compteur programme et une horloge. [4]

- **Les mémoires :**

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

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

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

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

I.3.2.2. Les modules d'entrées/sorties :

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions.

Plusieurs types de modules sont disponibles selon l'utilisation souhaitée :

- **Modules TOR :** l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1...) C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir, ...etc.
- **Modules analogiques :** l'information traitée est continue. Ces modules analogiques assurent la conversion analogique/numérique du signal délivré par un capteur analogique. Le signal est généralement du type tension : 0 à 10V ou -10V à +10V, ou du type courant : 4 à 20mA.

Un module peut être par exemple utilisé pour la mesure de température : la sonde résistive est reliée directement au module, lequel réalise une conversion

du courant délivré par la sonde avant l'écriture d'un mot (de "n" bits) dans la mémoire.

- **Modules spécialisés** : l'information traitée est contenue dans des mots codes sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

I.3.2.3. Module d'alimentation :

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

I.3.2.4. Modules de communication :

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

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

I.3.2.5. Les auxiliaires

Ce sont des modules indépendants tel que :

- le support mécanique (rack) : l'automate se présente alors sous forme d'un ensemble de cartes, d'une armoire, d'une grille et des fixations y correspondantes.
- Un ventilateur : il est indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée (plus de 40 °C).
- Un indicateurs d'état : il indique la présence de tension, l'exécution du programme (mode RUN), la charge de la batterie ainsi que le bon fonctionnement des coupleurs.

I.4. Traitement du programme dans l'automate :

Tous les automates fonctionnent selon le même mode opératoire :

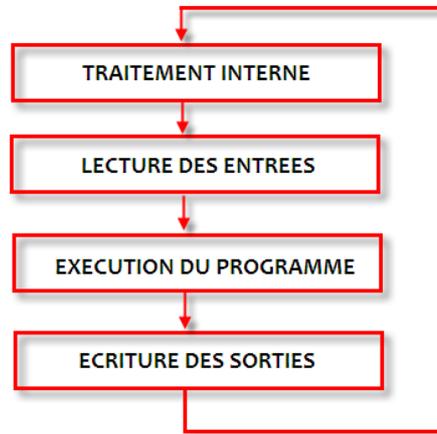


Figure I. 5. Traitement du programme dans l'automate.

- **Traitement interne** : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).
- **Lecture des entrées** : L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.
- **Exécution du programme** : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- **Ecriture des sorties** : L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

I.5. Les langages de programmation :

La norme de la Commission Électrotechnique Internationale (IEC 1131-3) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces langages peuvent être divisés en deux catégories. [5]

- Langages graphiques comme le SFC, FBD et le LD.
- Langages textuels tel que le ST et le IL.

I.5.1 . Les langages graphiques :

I.5.1.1. Le GRAFCET (SFC) :

Le langage (sequential function chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme

une suite connue d'étapes (états stables), reliées entre elles par des transitions. Une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD. Les principales règles graphiques du GRAFCET sont :

- Deux étapes ne peuvent se suivre sans qu'il n'y ait de Transition entre eux.
- Deux Transitions ne peuvent se suivre sans qu'il n'y ait d'étapes entre eux.

Les composants de base (symboles graphiques) du graphique SFC sont : [5]

- **Étapes et étapes initiales** : Une étape est représentée par un carré. Chacune des étapes est référencée par un nom inscrit dans son le cadre. Ces informations constituent le niveau "1" de l'étape. La situation initiale d'un programme SFC est représentée par ses étapes initiales. Le symbole d'une étape initiale est un cadre double.
- **Transitions** : Une transition est représentée par une barre horizontale qui croise un arc de liaison. Chacune est référencée par un nom, inscrit à côté du symbole de la transition.
- **Liaisons orientées** : Une simple ligne est utilisée pour représenter une liaison orientée entre une étape et une transition. Quand l'orientation n'est pas explicitement donnée, le lien est orienté du haut vers le bas.
- **L'envoi à une étape** : Un symbole de renvoi peut être utilisé pour représenter un arc de liaison d'une transition vers une étape, sans tracer le lien. Il doit être référencé avec le nom de l'étape de destination.
- **L'action** : représentée dans un cadre à côté de l'étape dans le GRAFCET. elle est référencée selon l'action se produisant lors de l'activation de l'étape.

Finalement le format graphique d'un programme GRAFCET est le suivant :

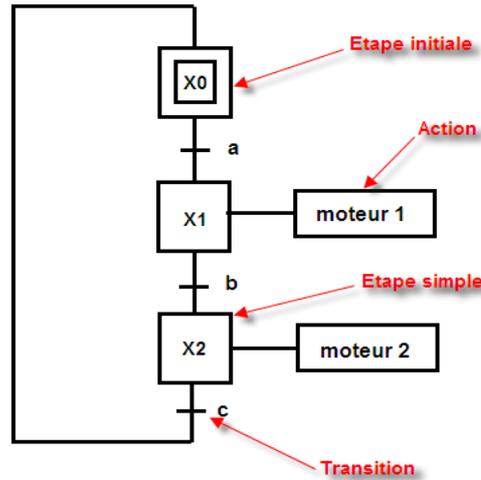


Figure I. 6. Le GRAFCET.

I. 5.1.2 Le Langage FBD :

Le diagramme FBD décrit une fonction entre des variables d'entrée et des variables de sortie. Une fonction est décrite comme un réseau de fonctions élémentaires. Les variables d'entrée et de sortie sont connectées aux boîtes fonctions par des arcs de liaison. Une sortie d'une boîte peut être connectée sur une entrée d'une autre boîte.

Les variables d'entrée du diagramme FBD doivent être connectées aux entrées des boîtes fonctions. Le type de chaque variable doit être cohérent avec le type de l'entrée de la boîte correspondante. Une entrée peut être une expression constante, une variable interne ou externe.

Les variables de sortie du diagramme FBD doivent être connectées aux sorties des boîtes fonctions. Le type de chaque variable doit être cohérent avec le type de la sortie de la boîte correspondante.

Une sortie peut être une variable interne ou de sortie, ou le nom du programme édité (pour les fonctions seulement). Quand le nom de la fonction éditée est utilisé comme sortie du diagramme, il représente une assignation de la valeur retournée par la fonction.

Par exemple pour calculer la valeur DELTA d'une équation de deuxième ordre $AX^2 + BX + C = 0$, avec $DELTA = B^2 - 4AC$, on utilise le diagramme FBD suivant :

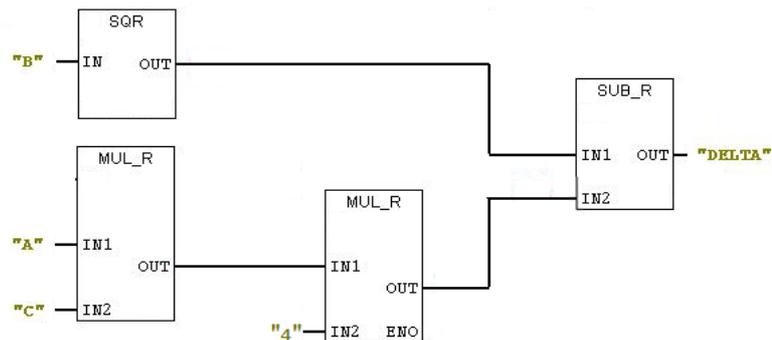


Figure I. 7.Exemple d'un programme FBD

I. 5.1.3 Le langage LD :

Le langage LadderDiagram (LD) est une représentation graphique d'équations booléennes combinant des contacts (arguments d'entrée) avec des bobines (résultats de sortie). Il permet la description de tests et la modification de données booléennes à l'aide des symboles graphiques placés dans un diagramme.

Les symboles graphiques LD sont organisés comme les éléments d'un schéma électrique à contacts. Et sont connectés à gauche et à droite aux barres d'alimentation verticales par des segments de liens.

Chaque segment de lien peut prendre l'état booléen FALSE ou TRUE. Cet état est le même pour tous les segments directement connectés ensemble.[6]

On peut réaliser comme exemple l'équation logique : $D = (A + B) \cdot \bar{C}$, par la représentation LD suivante :

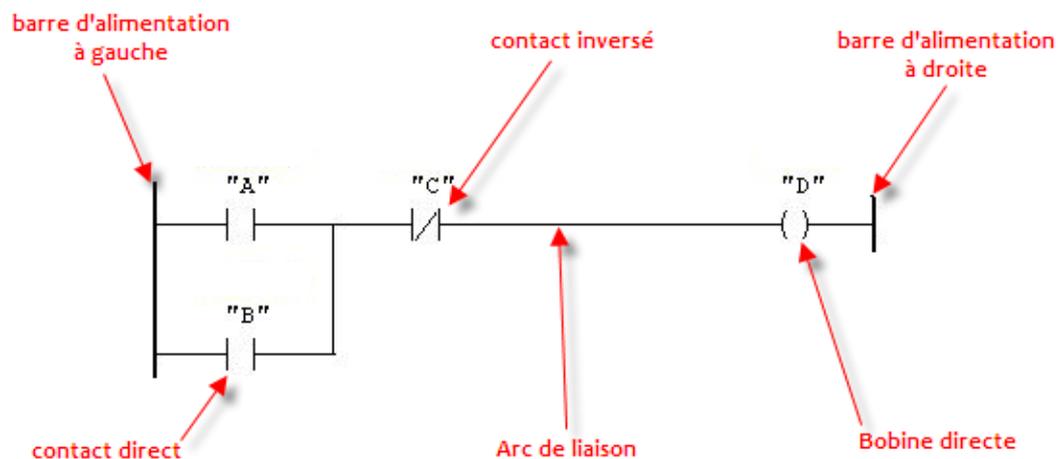


Figure I. 8: Exemple d'un programme LD

I. 5.2 . Les langages textuels :

I. 5.2.1. Le langage ST :

Le langage ST (StructuredText) est un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour implémenter des procédures complexes, difficilement modélisables avec les langages graphiques. Le langage

ST peut être utilisé pour la programmation des actions dans les étapes et les conditions associées aux transitions du SFC.

Un programme ST est une suite d'énoncés. Chaque énoncé se termine par un point-virgule (";"). Les noms utilisés par le codesource (identificateurs de variables, constantes, mots-clés du langage, etc.) sont délimités par des séparateurs passifs (espace, fin de ligne ou tabulation) ou par des séparateurs actifs, qui ont une signification bien définie et qui jouent le rôle d'opérateurs (par exemple, le séparateur ">" indique une comparaison "Plus grand que"). Des commentaires peuvent être insérés librement dans le texte de programmation. On peut citer plusieurs types d'énoncés standards du ST tels que : [6]

- Assignation (variable := expression);
- Appel de fonction ;
- Appel de bloc fonctionnel ;
- Énoncés de sélection (IF, THEN, ELSE, CASE, etc.) ;
- Énoncés d'itération (FOR, WHILE, REPEAT, etc.) ;
- Énoncés de contrôle (RETURN, EXIT, etc.) ;
- Énoncés spéciaux pour le lien avec d'autres langages, tels que le SFC.

Pour la résolution d'une équation de deuxième ordre $AX^2 + BX + C = 0$, on écrit le programme suivant en langage ST :

```

Delta := B * B - 4 * A * C;
IF Delta < 0 THEN
  NSolution := 0;
ELSE IF Delta = 0 THEN
  NSolution := 1;
  X1 := (-B) / (2 * A);
ELSE
  NSolution := 2;
  X1 := (-B + sqrt(Delta)) / (2 * A);
  X2 := (-B - sqrt(Delta)) / (2 * A);
END_IF

```

Figure I. 9. Exemple d'un programme ST.

5.2.2. Le langage IL :

Le langage IL, ou Instruction List, est un langage textuel de bas niveau. Les instructions travaillent toujours sur un résultat courant (ou registre IL). L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est à son tour stocké dans le registre.

Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une ligne nouvelle, et doit contenir un opérateur, complété éventuellement par des modificateurs et, si nécessaire pour l'opération, un ou plusieurs opérandes, séparés par des virgules (","). Une étiquette suivie de deux points (":") peut précéder l'instruction.[5]

On peut citer quelques opérateurs du langage IL :

- LD : Charger l'opérande;
- ST : Stocker le résultat courant ;
- CAL : Appeler un bloc fonctionnel ;
- JMP : Sauter une étiquette ;
- RET : Retour de fonction ;
- ADD : Addition ;
- MUL : multiplication ;
- SUB : soustraction.

Pour un exemple d'opérations différées imbriquées : résultat = a1 + (a2 * (a3 - a4) * a5) + a6, on écrit le programme suivant :

LD	a1	(* résultat := a1; *)
ADD (a2	(* ADD différé - résultat := a2; *)
MUL (a3	(* MUL différé - résultat := a3; *)
SUB	a4	(* résultat := a3 - a4; *)
)		(* exécute MUL différé - résultat := a2 * (a3-a4); *)
MUL	a5	(* résultat := a2 * (a3 - a4) * a5; *)
)		(* exécute ADD différé *)
		(* résultat := a1 + (a2 * (a3 - a4) * a5); *)
ADD	a6	(* résultat := a1 + (a2 * (a3 - a4) * a5) + a6; *)
ST	résultat	(* résultat := résultat courant *)

Figure I. 10. Exemple d'un programme IL.

I. 6. Présentation de quelques gammes SIMATIC :

Il existe plusieurs gammes de SIMATIC. On trouve trois grandes familles disponibles sur le marché :

I.6.1. SIMATIC S7 :

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

- **SIMATIC S7-200 :**

La famille S7-200 est constituée de micro-automates programmables utilisables dans des applications d'automatisations variées, Ces automates sont de type modulaire, avec possibilités d'extension jusqu'a 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.



Figure I. 11. Automate S7-200 [7].

- **SIMATIC S7-300 :**

La famille S7-300 est constituée d'automates programmables de conception modulaire utilisés pour des automatismes de moyenne gamme, et peuvent être connectés entre eux .

Cette famille a la possibilité d'extension jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et Industrial Ethernet.



Figure I. 12. Automate S7-300 [7].

- **SIMATIC S7-400 :**

cette famille est constituée d'automates programmables de hautes performances pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industrial Ethernet.

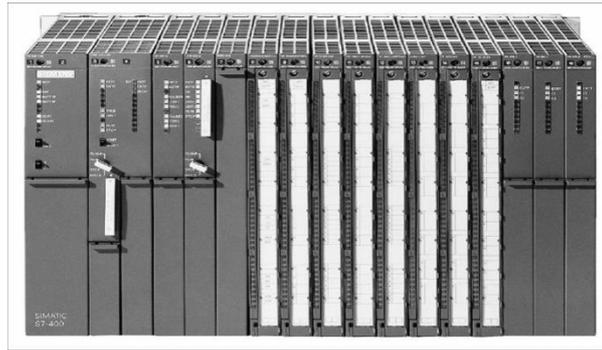


Figure I. 13. Automate S7-400 [7].

I.6. 2. SIMATIC M7 :

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

En raison de son architecture PC standard, les contrôleurs M7-300 et M7-400 constituent une extension programmable de la plate-forme d'automatisation SIMATIC. Les programmes utilisateur pour SIMATIC M7 peuvent être écrits dans un langage de haut niveau comme C et C++, les services fournis par le système d'exploitation sont extrêmement importants.



Figure I. 14. Automate M7. [7]

I.6. 3 .SIMATIC C7 :

Le SIMATIC C7 combine automate programmable et panneau operateur dans une seule unité. L'automate compte La CPU, les modules d'entrées /sorties et le panneau operateur qui est utilisé comme une interface Homme/Machine HMI.

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



Figure I. 15. Automate C7. [7]

I. 7. Critères de choix d'un automate programmable :

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

Le personnel de maintenance doit toutefois être formé sur ces matériels, et une trop grande diversité de matériels peut avoir de graves répercussions sur le fonctionnement du processus. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour pouvoir assurer les mises au point et dépannages dans des meilleures conditions.

La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

- Nombre d'entrées / sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme qui est souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes (commande d'axe, pesage ...) permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution, ...).

- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).

I. 8 .Description de l'automate S7- 314C-2 DP :

Cet automate est celui que nous avons utilisé dans notre application (Chapitre IV).

L'automate programmable S7- 314C-2 DP, c'est un automate modulaire constitué des éléments suivants :

- Un module d'alimentation ;
- Une CPU314C ;
- Un module d'entrées, sorties TOR ;
- Un module d'E/S analogiques.



Figure I. 16. Vue générale de l'automate S7- 314C-2 DP.

I.8 .1. Module d'alimentation :

Tout réseau 24V peut être utilisé pour alimenter de La CPU des S7-300. Les modules d'alimentation suivants de la gamme S7 sont prévus pour être utilisés avec le S7-300(voir tableau I.1).

Désignations	Courant de sorties	Tension à la sorties	Tension à l'entrée
	2 A	DC 24 V	AC 120/230 V

PS 307	5 A	DC 24 V	AC 120/230 V
	10 A	DC 24 V	AC 120/230 V

Tableau I. 1 Les modules d'alimentations suivantes la gamme S7-300. [8]

I. 8 .2. LaCPU :

Les CPU de la série S7-300 se composent des éléments suivants :

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

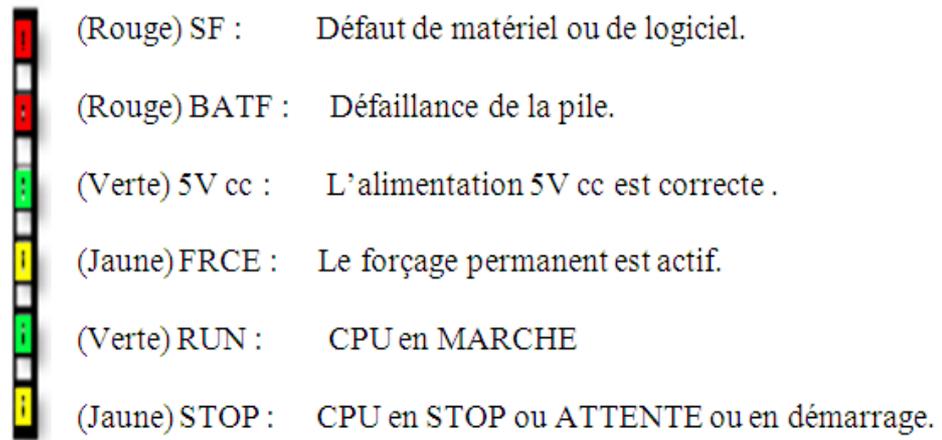


Figure I. 17.LED de visualisation d'état de la CPU.

- **Commutateur de mode de fonctionnement :**Le changement de mode se fait à l'aide d'uneclé.

Position	Signification	Explication
RUN -P	Mode de fonctionnement RUN-PROGRAM	La CPU traite le programme utilisateur. Le programme peut être modifié. Dans cette position la Clé ne peut pas être retirée.
RUN	Mode de fonctionnement RUN	La CPU traite le programme utilisateur. Le programme ne peut être modifié qu'avec légitimation par mot de passe la Clé peut être retirée.

STOP	Mode de fonctionnement	La CPU ne traite aucun programme utilisateur. La Clé peut être retirée.
MRES	Effacement Général	Position instable du commutateur, pour effectuer l'effacement général il faut respecter un ordre particulier de commutation

Tableau I. 2. Positions du commutateur du mode de fonctionnement.

- **Pile de sauvegarde ou accumulateur :**

L'utilisation de l'accumulateur ou de la pile de sauvegarde est nécessaire pour l'horlogetemps réel.La pile de sauvegarde est aussi utilisée pour :

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

L'accumulateur est rechargé à chaque mise sous tension de La CPU. Son autonomieest de quelques jours voir quelques semaines, au maximum. La pile de sauvegarde n'estpas rechargeable mais son autonomie peut aller jusqu'à une année.

- **Carte mémoire :**

La CPU S7- 314C-2 Dutilisée n'est pas dotée d'une carte mémoire mais il faut savoirque la plupart des CPU en possèdent, son rôle est de sauvegarder le programmeutilisateur, le système d'exploitation et les paramètres qui déterminent le comportementde la CPU et des modules en cas de coupure du courant.

- **Interface MPI : (interface multipoint) :**

La MPI 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. La vitesse de transmission typique est de 187,5 k bauds.

Quant aux éléments : pile, carte mémoire, interface MPI, DP seront décrits dans le tableau suivant, avec les caractéristiques techniques de la CPU :

CPU et version	
N° de référence	6ES7 314-6CG03-0AB0
Version matériel	01
Version microprogramme	V 2.6
Pack de programmation correspondant	STEP 7 à partir de V5.4 + SP3 ou STEP 7 à partir de V5.3 + SP2 avec HSP0123
Mémoires	
Intégré	96 Ko
Extensible	Non
Mémoire de chargement	Enfichable via SIMATIC microcarte mémoire (max. 8 Mo)
Conservation des données sur la microcarte mémoire (après la dernière programmation)	Minimum 10 ans
Sauvegarde	Garantie par la microcarte mémoire
Temps de traitement (µs)	
Opérations sur bit	Min. 0.1
Opérations sur mots	Min. 0.2
Opérations arithmétiques sur nombres entiers	Min. 2
Opérations arithmétiques sur nombres réels	Min. 3
Compteurs	

Compteurs S7 : Nombre Plage de comptage Rémanence Rémanence par défaut	256 0 à 999 Réglable De Z0 à Z7
Compteurs CEI : Type Nombre	SFB Illimité (suivant la mémoire de travail)
Temporisations	
Temporisation S7 : Nombre de temporisations Plage de temps Rémanence Rémanence par défaut	256 10 ms à 9990 s Réglable Pas de rémanence
Temporisation CEI : Type Nombre	SFB Illimité (suivant la mémoire de travail uniquement)
Zones de données	
Mémentos : Taille Par défaut Mémentos de cadence	256 octets De MB 0 à MB 15 8 bits (1 octet de memento)
Blocs de données : Nombre Taille	551 : de DB1 à DB551 Max. 16Ko
Blocs	
Nombre total	1024 (DB, FB, DB)
OB : Taille cycle libre alarmes horaires alarmes temporisées alarmes cycliques alarmes processus alarme DPV1 alarme de démarrage alarmes d'erreur asynchrones alarmes d'erreur asynchrones	Max. 16Ko OB1 OB10 OB20 OB35 OB40 OB 55, 56, 57 OB100 OB80, 82, 85, 87 OB121, 122

FB : Nombre Taille	1024 : de 0 à 1023 Max. 16Ko
FC : Nombre Taille	1024 : de 0 à 1023 Max. 16 Ko
Zones d'adresses (E/S)	
Périphérie totale : Entrées Sorties	1024 octets (adressage libre) 1024 octets (adressage libre)
Mémoire image des E/S : Entrées Sortie	128 octets 128 octets
Voies TOR : Voies intégrées d'entrées Voies intégrées de sorties	24 voies intégrées 16 voies intégrées
Voies analogiques : Voies intégrées d'entrées Voies intégrées de sorties	4+une entrées résistance 2
Interface de communication MPI	
Communication PG/PC(OP)	Oui
Routage	Non
Communication par données globales	Oui
Communication de base S7	Oui
Vitesse de transmission	187.7 kBauds max
Maitre DP	
Communication PG/PC(OP) Routage Communication par données globales Communication de base S7 Vitesse de transmission	Oui Oui Non Oui (uniquement blocs I) jusqu'à 12 Mbauds
Esclave DP	
Communication PG/PC(OP) Routage Communication par données globales Communication de base S7 Vitesse de transmission	Oui oui (uniquement pour une interface active) Non Non jusqu'à 12 Mbauds
Programmation	

Langage de programmation Niveaux de parenthèses Fonctions système (SFC) Blocs fonctionnels système (SFB) Protection du programme utilisateur	CONT, LIST, LOG 8 voir liste des opérations voir liste des opérations oui
E/S intégrées	
Adresses par défaut des : Entrées TOR Sorties TOR Entrées analogiques Sorties analogiques	De 124.0 à 126.7 De 124.0 à 125.7 De 752 à 761 De 752 à 755
Dimensions	
montage L x H x P (mm) Poids	120 x 125 x 130 env. 676 g
Tensions, courants	
Tension d'alimentation (valeur nominale) Plage admissible Courant absorbé (en marche à vide) Courant d'appel à l'enclenchement Courant absorbé (valeur nominale) Protection externe des conducteurs de l'alimentation (conseillée) Puissance dissipée	24VDC De 20.4 à 28.8VDC 150 mA 11 A 800mA Commutateur LS, type C min. 2 A, Commutateur LS, type B min. 4 A 14 W

Tableau I. 3.Caractéristiques techniques de La CPU S7-314C-2DP. [9]

I. 8 .3. Le module E/S digitales intégré (DI24/DO16) :

Nous avons vu que l'automate S7- 314C-2 DP possède son propre module E/S digitales, ses caractéristiques sont résumées dans le tableau suivant :

Données spécifiques au module	
Entrées :	
Nombre	24
Entrées utilisables pour les fonctions	16
Longueur de câble	1000 m
Visualisation	LED verte par voie

Sorties :	
Nombre	16
Sorties rapides	4
Longueur de câble	1000 m
Visualisation	LED verte par voie
Tensions, courants, potentiels	
Entrées :	
Tension de charge nominale	24V
Irréversibilité	OUI
Nombre d'entrées pouvant être activées Simultanément	24 jusqu'à 40°C en position horizontale 12 jusqu'à 60°C en position horizontale 12 jusqu'à 40°C en position verticale
Sorties :	
Tension de coupure inductive	L+ (-48 V)
Irréversibilité	3A jusqu'à 40°C en position horizontale 2A jusqu'à 60°C en position horizontale 2A jusqu'à 40°C en position verticale
Caractéristiques pour la sélection d'un capteur (entrées)	
Tension d'entrée :	
Valeur nominale	24 V DC
Pour le signal '1'	15 à 30 V DC
Pour le signal '0'	-3 à +5 VDC
Courant d'entrée avec le signal '1'	9 mA
Caractéristiques pour la sélection d'un actionneur (sorties)	
Tension de sortie avec le signal '1'	24VDC (-0.8V)
Courant de sortie Avec le signal '1':	
Valeur nominale	500 mA
Plage admissible	De 5 mA à 0.6 A
Courant de sortie Avec le signal '0'	0.5 mA
Plage de résistance de charge	De 48 Ohm à 4 kOhm

Tableau I. 4.Caractéristiques techniques du module E/S digitaleintégré. [9]

I. 8 .4. Le module E/S analogiques intégré (AI5/AO2) :

Les caractéristiques du module E/S analogique sont résumées dans le tableau suivant :

Donnée spécifiques au module	
Entrées : Nombre Longueur de câble	4 en mesure courant /tension 1 en mesure résistance/température 100 m
Sorties : Nombre Longueur	2 200m
Tension, courants, potentiels	
Entrée de résistance : Tension de marche à vide Courant de mesure	2,5 V De 1.8 à 3.3 mA
Sorties : Tension de charge nominale Irréversibilité	24 V Oui
Formation des valeurs analogiques	
Entrées : Principe de mesure	Codage de la valeur actuelle (approximation Successives
Temps d'intégration/conversion/résolution: Paramétrable Temps d'intégration en ms Fréquence d'entrée autorisée Résolution Constante de temps du filtre d'entrée Temps d'exécution de base	Oui 2,5 / 16,6 / 20 ms 400 Hz 12 bit 0.38 ms 1 ms
Sorties : Résolution Temps de conversion	12 bit 1 ms
Temps d'établissement : Pour la charge ohmique Pour la charge capacitive Pour la charge inductive	0.6 ms 1 ms 0.5 ms

Caractéristiques pour la sélection d'un capteur	
Etendues d'entrée/Résistance d'entrée : Tension Courant Résistance Thermomètre de résistance	$\pm 10 \text{ V}/100 \text{ k}\Omega$; 0 V à 10 V/100 k Ω $\pm 20 \text{ mA}/50 \Omega$; 0 mA à 20 mA/50 Ω ; 4 mA à 20 mA/50 Ω 0 Ω à 600 Ω /10 M Ω Pt 100/10 M Ω
Tension d'entrée autorisée (limite) : Pour la tension d'entrée Pour l'entrée de courant	Max. 30V en permanence max. 2,5 V en permanence
Courant d'entrée autorisé : Pour la tension d'entrée Pour l'entrée de courant	max. 0,5 mA en permanence max. 50 mA en permanence
Raccordement du capteur de signaux : Pour la mesure de la tension	Possible
Pour la mesure du courant : Transducteur de mesure 2 fils Transducteur de mesure 4 fils Pour la mesure de la résistance : Borne de ligne 2fils Borne de ligne 3fils Borne de ligne 4fils	Possible avec alimentation externe Possible possible, sans compensation des résistances de câble impossible impossible
Caractéristiques pour la sélection d'un actionneur	
Tension Courant	$\pm 10 \text{ V}$; 0 V à 10 V $\pm 20 \text{ mA}$; 0 mA à 20 mA ; 4 mA à 20 mA
Résistance de charge : Capacitive (sortie tension) Inductive (sortie courant)	min. 1 k Ω , max. 0,1 μF max. 300 Ω , 0,1 mH
Sortie tension : Protection contre les courts-circuits Courant de court-circuit	oui 55 mA

Sortie de courant : Tension de marche à vide	17 V
Limite des courants/tensions appliqués Del'extérieur : Tension au niveau des sorties Courant	max. 16 V en permanence max. 50 mA en permanence
Raccordement des actionneurs : Pour la sortie du tension : Raccordement 2 fils Montage à 4 fils Pour la sortie courant : Raccordement à 2fils	Possible Impossible Possible

Tableau I. 5. Caractéristiques techniques du module E/S analogique. [9]

La correspondance tensions valeurs analogiques est représentée dans le tableau qui suit :

Système			Etendue de tension	
Pourcentage	Décimale	Hexadécimal	Tension 0 à 10 V	Domaine
117.59 %	32511	7EFF	11.76 V	Domaine de dépassement
	27649	6C01		
100 %	27648	6C00	10 V	Etendue nominale
75 %	20736	5100	7.5 V	
0.0036 %	1	1	361.7 μ V	
0 %	0	0	0 V	
	-32768	8000	0 V	Impossible la tension min.d'entrées ou de sortie est limitée à 0 V

Tableau I. 6. Correspondance valeurs analogiques tension.[9]

La correspondance courant valeurs analogiques est représentée dans le tableau qui suit :

Système			Etendue de Courant	
Pourcentage	Décimale	Hexadécimal	Courant 0 mA à 20 mA	Domaine
117.56 %	32511	7EFF	23.52 mA	Domaine de
	27649	6C01		

				dépassement
100 %	27648	6C00	20 mA	Etendue nominale
75 %	20736	5100	15 mA	
0.0036 %	1	1	723.4 nA	
0 %	0	0	0 mA	
	-1	FFFF	0 mA	
	-32768	8000	0 mA	Impossible la tension min.d'entrées ou de sortie est limitée à 0 V

Tableau I. 7. Correspondance valeurs analogiques courant.[9]

I.10. Conclusion :

L'automate programmable est devenu un produit incontournable dans les systèmes de production automatisés. Cela est due à sa mobilité, la flexibilité de son architecture, sa facilité de programmation et à son adaptation rapide dans les milieux industriels.

Dans ce chapitre, on a pu connaître les multitudes de possibilité d'utilisation que nous offre l'API. Ce dernier possède une architecture complexe et modifiable selon la nécessité du projet. Ce qui optimise sont efficacité pour n'importe quelle application, vue les possibilités de personnalisation.

Ceci dit la personnalisation de l'API et d'autant software que hardware. Pour le software l'API possède une palette de langages différents, chacun d'eux propose une meilleure approche que les autres par rapport à la nature de configuration du projet. Donc on pourra dire que le software rajoute à la personnalisation que propose déjà le hardware.

Toutefois le choix de l'API n'est pas relié qu'à son architecture et ses langages de programmation, d'autres choix indépendants de sa technologie doivent être pris en comp

CHAPITRE II :

STEP7, logiciel de programmation des API Siemens

II.1. INTRODUCTION :

Les API ont pour fonctions de remplir des tâches décommande en élaborant des actions en suivant une algorithmique appropriée à partir d'informations données par des capteurs.

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

II.2. Présentation du logiciel STEP :

Le STEP7 est un logiciel de base pour la configuration et la programmation des systèmes d'automatisation SIMATIC S300 et S400. Il fait partie de l'industrie logicielle SIMATIC. En effet il assiste dans toutes les phases du processus de création de la solution d'automatisation, il permet la programmation et la simulation de procédures automatisées avec différents langages normalisés, il inclut 6 applications :

II.2.1. Gestionnaire de projets SIMATIC Manager :

Le gestionnaire de projets SIMATIC gère toutes les données relatives à un projet, il démarre automatiquement les applications requises pour le traitement des données sélectionnées.

II.2.2. Editeur de la configuration matérielle :

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

II.2.3. Editeurs de programmes des Différents langages de programmation :

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

- Editeur de programme CONT, LIST, LOG qui correspondent respectivement aux langages normalisés LD Ladder Diagramme, IL Instruction List, FBD Fonction Bloc Diagramme).
- D'autres langages de programmation peuvent être inclus sous forme de logiciels additionnels :
- Editeur de programmes SCL qui correspond au langage normalisé ST Structure Text.

- Editeur de programmes Graph qui correspond au GRAFCET.

II.2.4. Interface de paramétrage du régulateur PID :

Cette interface permet de paramétrer le régulateur PID du projet d'automatisation. Il permet aussi une visualisation graphique du comportement de la régulation.

II.2.5. Paramétrage de l'interface PG/PC :

Cette interface a pour but de configurer la communication avec l'automate en termes de vitesse de transmission dans le réseau Profibus ou MPI en vue d'une transmission de projets.

II.2.6. Le simulateur de programmes PLCSIM :

Ce logiciel permet d'exécuter et de simuler un programme S7, la simulation étant réalisée complètement au sein du logiciel step7, il n'est pas nécessaire qu'une liaison PC/automate soit établie. Cette interface simple permet de visualiser et de forcer les différents paramètres présents dans le programme.

II.3. Création d'un projet SIMATIC Manager :

Nous allons expliquer les différentes étapes pour créer un nouveau projet. On démarre le programme SIMATIC Manager. Ce programme est l'interface graphique qui permet la manipulation du projet et l'accès aux autres programmes de STEP7.

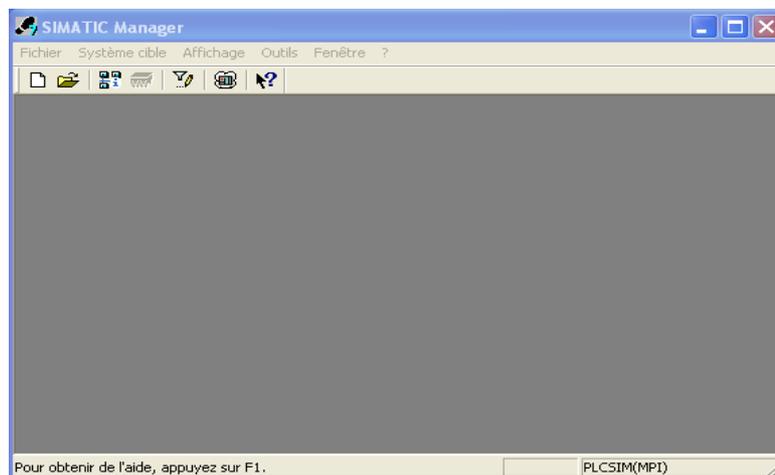


Figure II. 1 Création d'un nouveau projet (1)

Un clic sur « nouveau projet » permet de créer un projet d'automatisation.

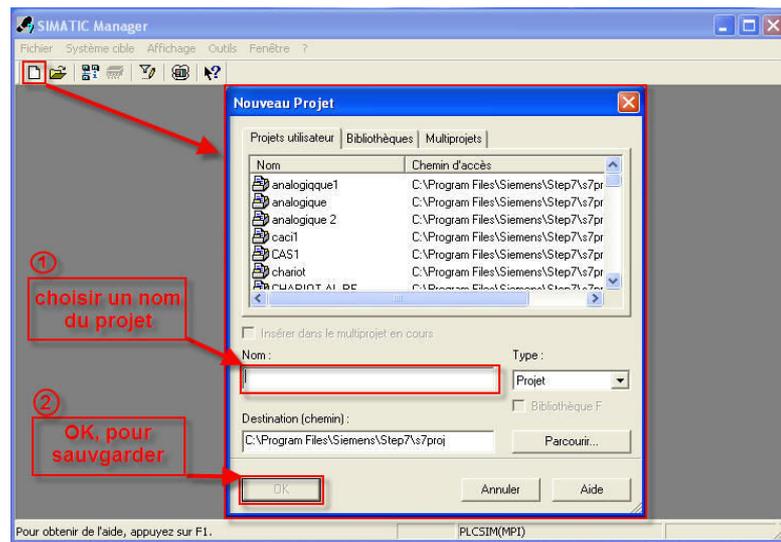


Figure II. 2Création d'un nouveau projet (2)

Une fois le nom du projet est choisi et sauvegardé, la fenêtre suivante apparaît :

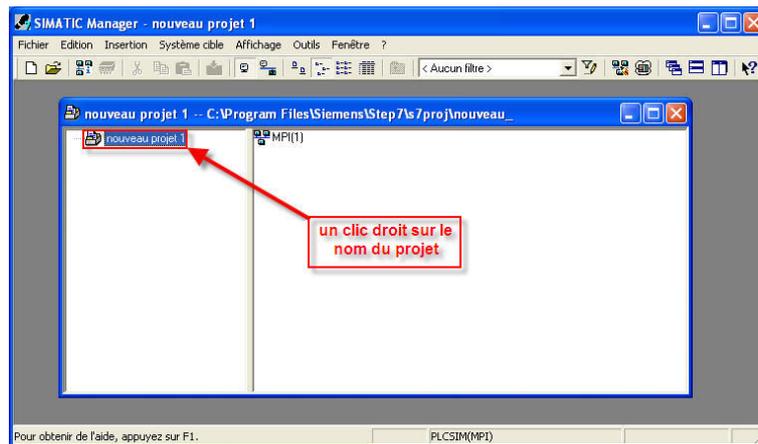


Figure II. 3.Insertion d'une station de travail

Il faut d'abord choisir une station de travail SIMATIC, cette station représente une configuration matérielle S7 comportant un ou plusieurs modules programmables.

Il existe différents types de station SIMATIC disponibles:

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

Un clic droit sur le projet en cours nous permet de choisir la station SIMATIC adéquate. Pour notre exemple on a choisi une station SIMATIC 300 :

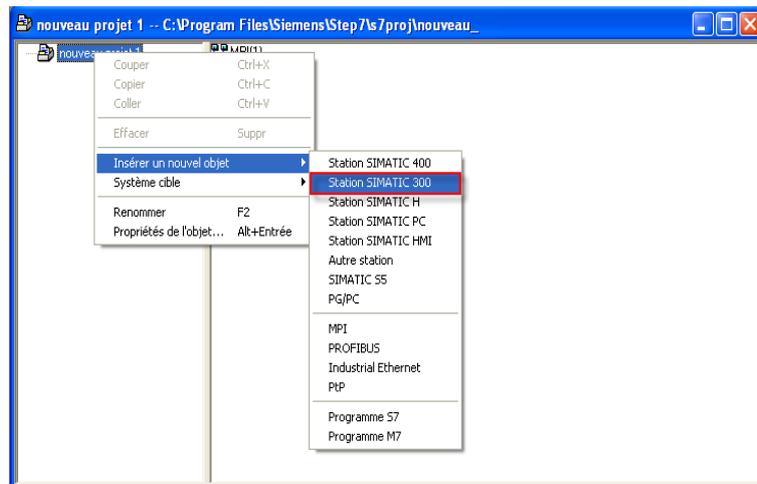


Figure II. 4. Choix de la station SIMATIC.

II.3.1. Configuration matérielle :

Pour la configuration matérielle, nous utilisons un catalogue des modules pour définir la CPU et tous les modules contenus dans sa commande. On démarre la configuration matérielle par double clic sur la station, puis sur 'Matériel', la fenêtre suivante apparaît :

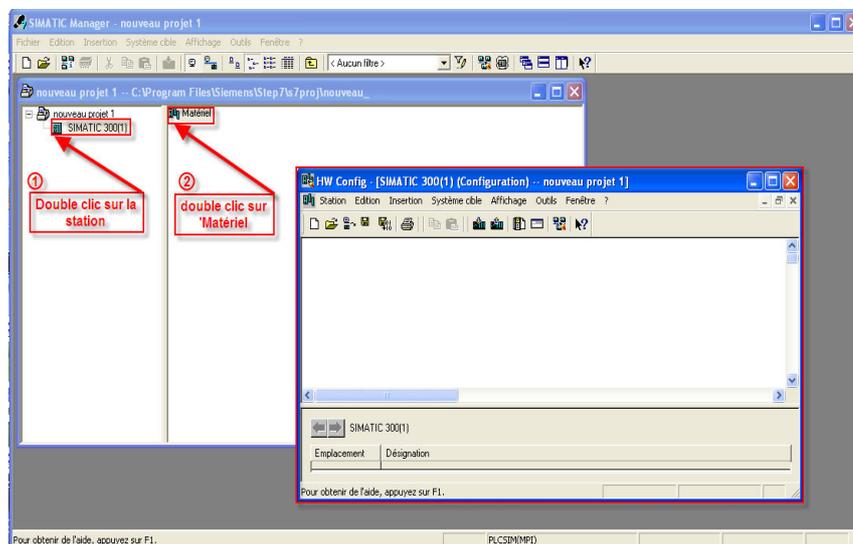


Figure II. 5. Editeur des configurations matérielles.

Un clic sur l'objet Catalogue Hardware nous permet de choisir un de ces répertoires suivants :

- PROFIBUS-DP, SIMATIC 300, SIMATIC 400 et SIMATIC PC Based Control, ainsi que tous les supports et composants modules d'interface nécessaires à la configuration de notre installation.

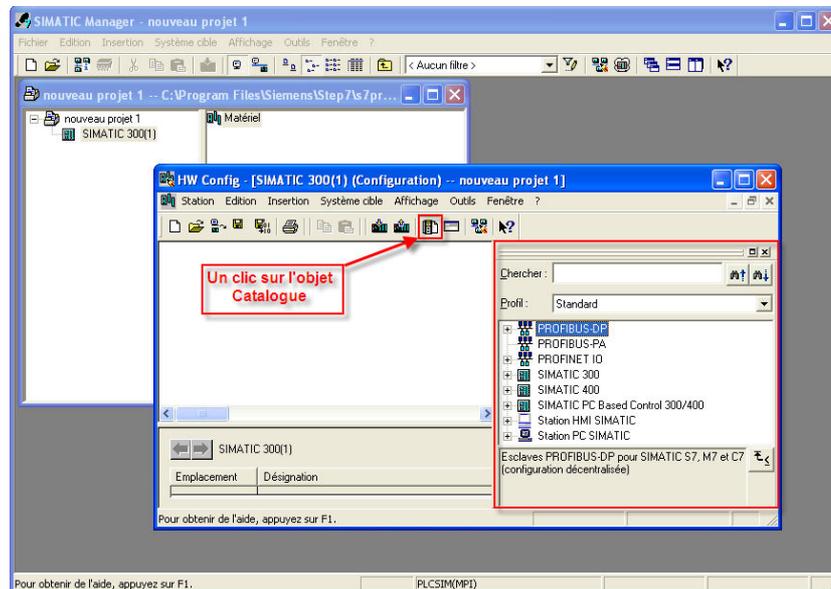


Figure II. 6. Catalogue du matériel.

Après le choix de SIMATIC 300 dans le catalogue, un clic sur RACK-300 nous permet de choisir le Profilé support qui contiendra tous les différents modules par la suite.

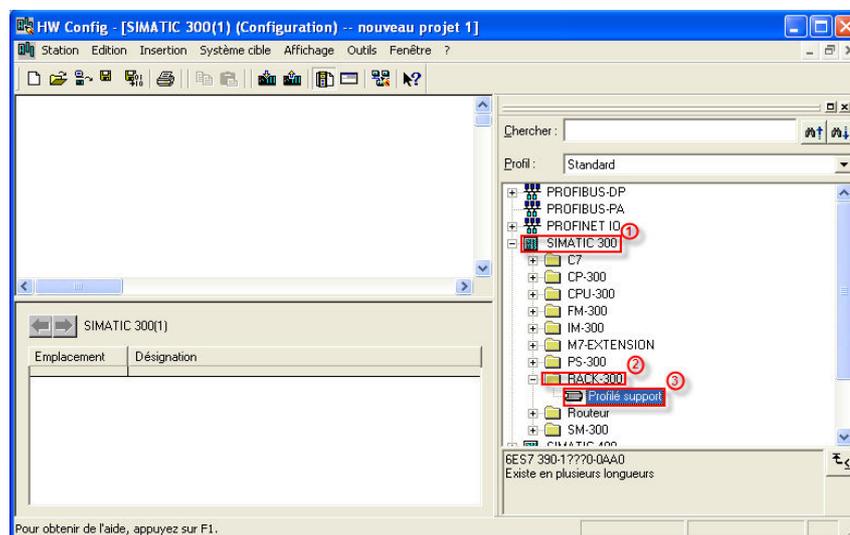


Figure II. 7. Catalogue SIMATIC 300.

Deux listes d'équipement de châssis apparaissent alors dans la fenêtre divisée en deux parties : une simple liste en haut, une vue détaillée avec les références de commande, les adresses MPI et les adresses d'E/S en bas. Comme nous montre la figure suivante :

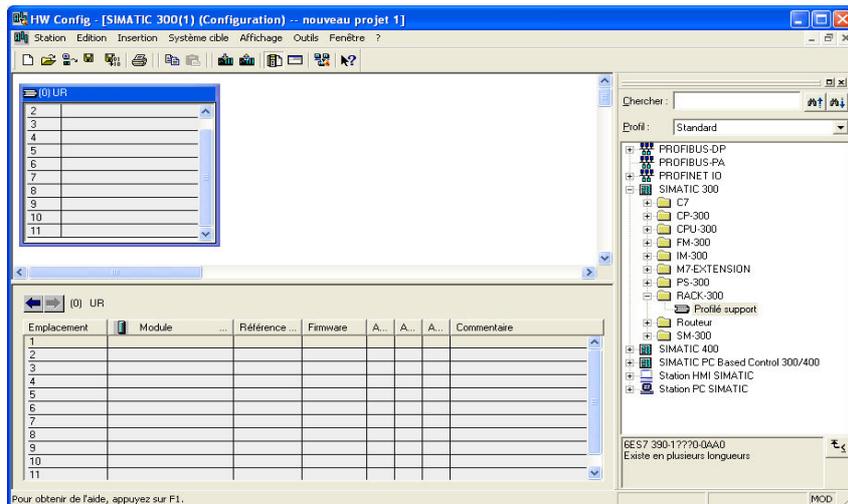


Figure II. 8. Sélection des modules (1).

Le rack possède onze emplacements :

- Le premier emplacement est réservé exclusivement au module d'alimentation, celui-ci peut être inséré par un double clic sur le module "PS-300" (PS: Power Supply) à partir du catalogue.
- Le deuxième est réservé à la CPU, celui-ci est sélectionné et inséré à partir du catalogue "CPU-300".
- Le troisième est réservé à des modules spéciaux, par exemple comme adresse logique pour un coupleur (dans une configuration multichâssis). On peut l'insérer à partir du dossier IM-300.
- À partir du quatrième emplacement, il est possible de monter au choix jusqu'à 8 modules de signaux (SM-300), processeurs de communication (CP-300) ou modules fonctionnels (FM-300). Les emplacements possibles pour un module sélectionné sont automatiquement repérés en vert.

L'entrée dans la liste à l'emplacement sélectionné préalablement se fait en double-cliquant sur le module à insérer et l'entrée à un emplacement quelconque (repéré en vert) de la liste s'effectue par glisser-déplacer.

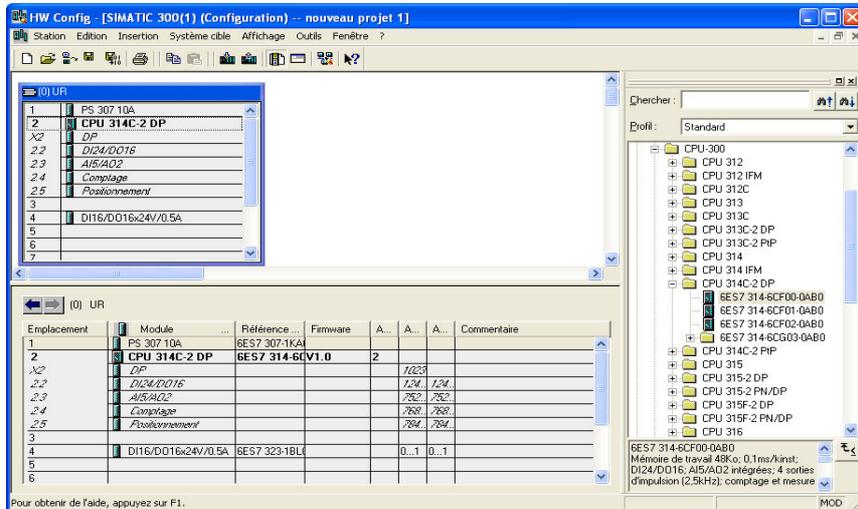


Figure II. 9. Sélection des modules (2).

Après l’insertion de tous les éléments requis pour la configuration, un double clic sur chacun d’eux permet leurs paramétrages. Cliquant sur le module CPU, la fenêtre de configuration du CPU s’ouvre :

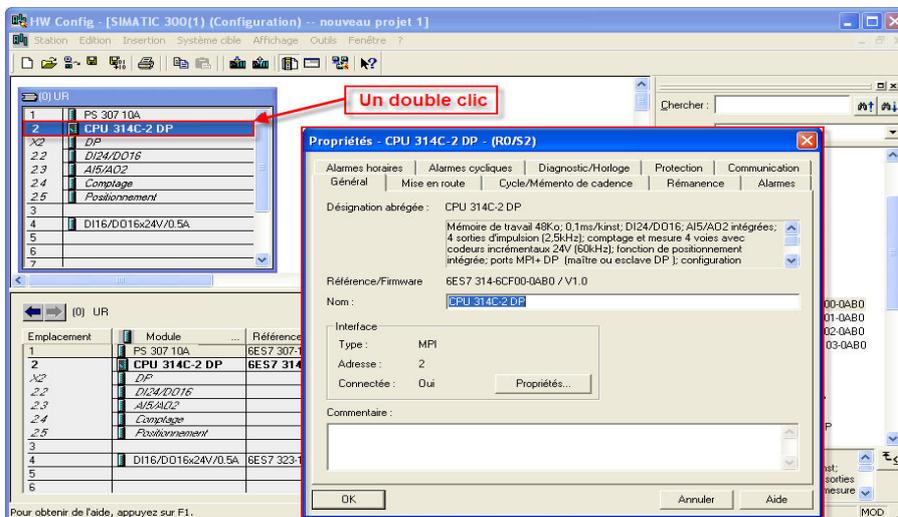


Figure II. 10. Paramétrage de la CPU (1).

La page de l’onglet « Général » contient toute les informations sur le module CPU :

- La taille de la mémoire.
- Le nombre d’entrées digitales.
- Le nombre de sorties digitales.

- Le nombre d'entrées analogiques.
- Le nombre de sorties analogiques.
- La vitesse d'exécution d'une instruction logique.
- Le numéro de référence.
- L'emplacement (châssis 0, position 2);
- La station réseau.

Une foule d'autres onglets donnent accès à des fonctions avancées.

En cliquant l'onglet « Cycle/Mémoire de cadence », la fenêtre suivante apparaît :

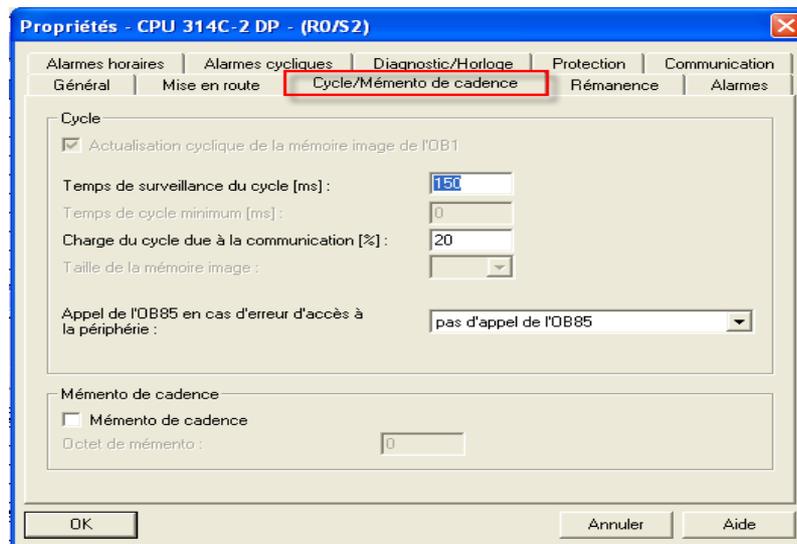


Figure II. 11. Paramétrage de la CPU (2).

- Le temps de surveillance de cycle est l'intervalle de temps maximal entre deux initialisations du « chien de garde », celui-ci et un dispositif qui surveille continuellement le processeur du module CPU. Si le processeur fonctionne correctement, il initialise le chien de garde à intervalle régulier. Si le processeur est défaillant, il n'initialisera plus le chien de garde. Dès que la dernière initialisation remonte à un délai supérieur au temps de surveillance, le chien de garde bloque le fonctionnement du processeur.

Cette sécurité assure qu'un processeur défaillant ne risque pas de causer de graves incidents.

- Les mémoires de cadence sont des mémoires qui modifient périodiquement leur valeur binaire (rapport cyclique 1:1).

Une période/fréquence donnée est affectée à chaque bit de l'octet de mémoire de cadence.

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

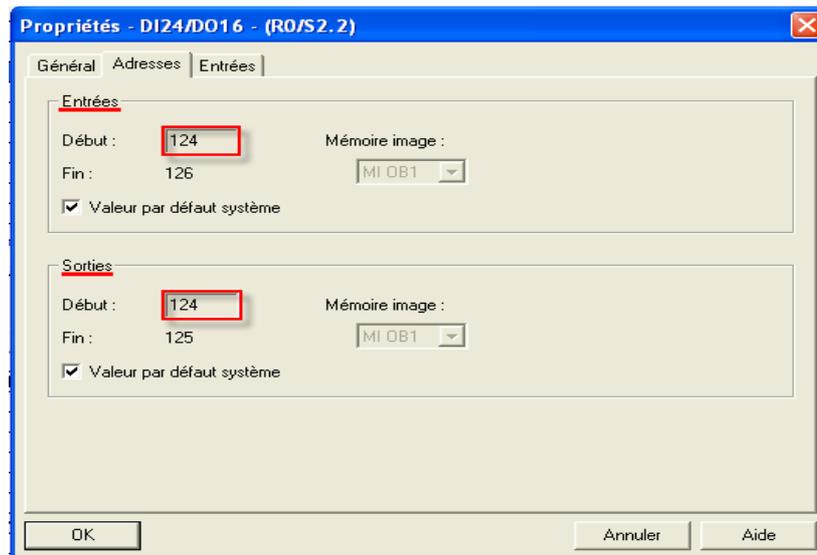


Figure II. 12. Paramétrage du module E/S TOR.

On peut aussi effectuer le paramétrage des modules d'entrées/sorties analogiques :

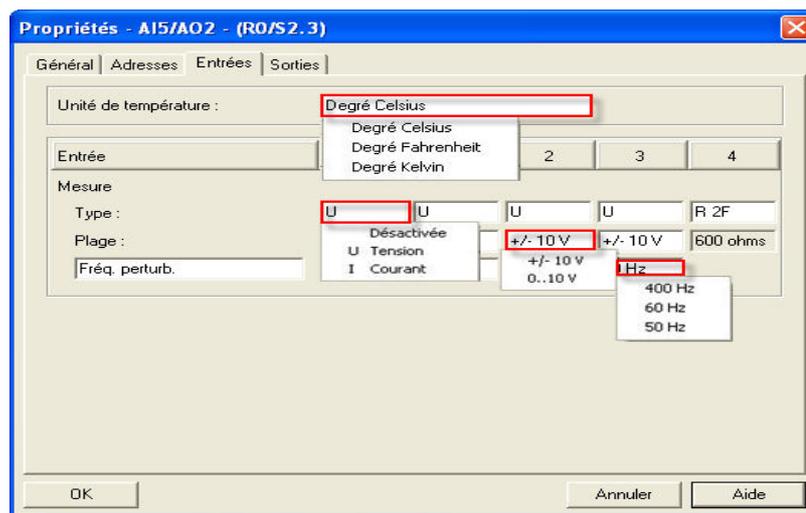


Figure II. 13. Paramétrage du module E/S analogiques.

Cette fenêtre nous permet de configurer les Entrées /Sorties analogiques par rapport au type du capteur relié à l'automate, on peut définir :

- L'unité de température (degré Celsius, fahrenheit et Kelvin), avec laquelle la valeur analogique est fournie.
- Le type de signal de mesure fournit par le capteur sous forme de tension (U) ou de courant (I). ou bien Choisissant "désactivé", lorsqu'une entrée analogique n'est pas activée.
- La plage de variation de ses signaux, par exemple pour le courant de 0 à 20 mA, et pour la tension de 0 à 10V.

- La fréquence perturbatrice (50, 60 Hz..) ou le temps d'intégration (16, 20 ms..).

II.3.2 Edition de la table des mnémoniques :

La Mnémonique est le nom donné par l'utilisateur et qui peut remplacer une variable ou un bloc de programmation. La table de mnémonique s'agit d'une table qui permet d'affecter des noms à des adresses de données globales accessibles à partir de tous les blocs.

Pour l'insertion d'une table de mnémoniques, on clique sur <<programme,mnémonique>> comme le montre la figure suivante :

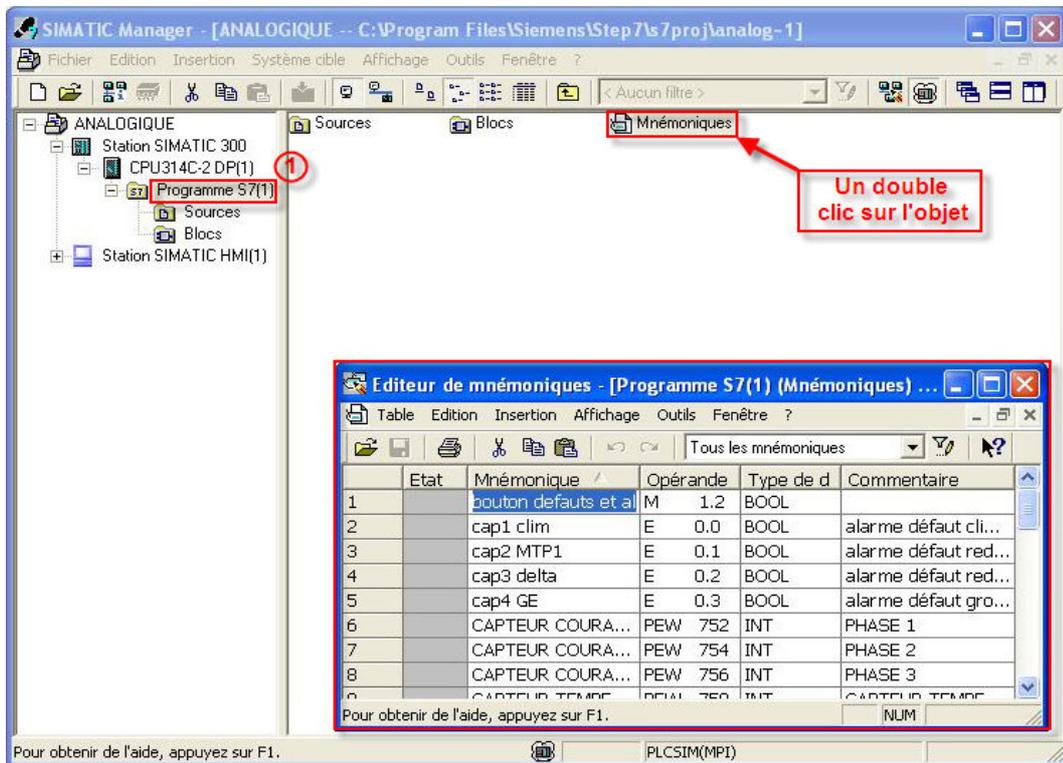


Figure II. 14. Editeur des mnémoniques.

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

Zone mémoire	Type de la variable	Valeurs possibles
Pour le bit	Booléenne	Vrai ou faux (1, 0)
Pour l'octet (8 bits)	Deux de données possibles : -BYTE : Nombre hexadécimale. -CHAR : Caractères ASCII.	-De B#16#0 à B#16#FF. -'A', 'B'.....
Pour le mot (16 bits)	Plusieurs possibilités :	

	<p>-WORD : Nombre hexadécimale.</p> <p>-INT : Nombre entier.</p> <p>- S5TIME : durée s7 en pas de 10ms (valeur par défaut).</p> <p>- DATE : Date en incréments de 1 jour</p>	<p>- De W#16#0 à W#16#FFFF.</p> <p>- De -32768 à 32767.</p> <p>- De S5T#0H_0M_0S_10MS à S5T#2H_46M_30S_0MS.</p> <p>- De D#1990-1-1 à D#2168-12-31</p>
Pour le double mot (32 bits)	<p>Plusieurs possibilités :</p> <p>- DWORD : nombre hexadécimal.</p> <p>- DINT : nombre entier.</p> <p>- REAL : nombre à virgule flottante.</p> <p>- TIME : Durée en incréments de 1 ms.</p> <p>- TIME_OF_DAY : Heure en pas de 1 ms.</p>	<p>- De DW#16#0000_0000 à DW#16#FFFF_FFFF.</p> <p>- DE L#-2147483648 à L#2147483647.</p> <p>- Limite supérieure : 3.402823e+38</p> <p>Limite inférieure : 1.175495e-38.</p> <p>- DE - T#24D_20H_31M_23S_647MS. à T#24D_20H_31M_23S_647MS.</p> <p>- De TOD#0:0:0.0 à TOD#23:59:59.999.</p>

Tableau II. 1. Les zones mémoires.

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

Nom de la zone	Description	Accès à la zone par :
Mémoire Image des Entrées (MIE)	Au début du cycle le système d'exploitation lit les entrées provenant du processus et enregistre ces valeurs dans la MIE. Le programme utilise ces valeurs pendant son traitement normal.	<p>Bit : E</p> <p>Octet : EB</p> <p>Mot : EW</p> <p>Double mot : ED</p>

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

Tableau II. 2. Accès aux différentes zones mémoire. [10]

II.3.3. Edition des programmes:

Le logiciel de programmation STEP 7 nous permet de structurer notre processus d'automatisation, c'est-à-dire de le subdiviser en différentes tâches. Les parties d'un programme utilisateur structuré correspondent à ces différentes tâches ; il s'agit des blocs du programme.

II.3.3.1. Organisation par blocs des programmes S7 :

- **Blocs d'organisation (OB) :**

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation pour gérer le traitement du programme cyclique, ainsi que pour la mise en route de l'automate programmable et détecter les erreurs.

Les blocs d'organisation définissent l'ordre (événements de déclenchement) dans lequel les différentes parties du programme sont traitées.

On appelle alarmes les événements qui déclenchent l'appel d'un OB donné. Le type d'alarme définit la classe de priorité de celle-ci. Il existe 13 types d'alarmes montré dans le tableau suivant :

Type d'alarme	Bloc d'organisation	Description
Cycle libre	OB1	il s'exécute d'une façon continue. Son traitement constitue le traitement normal pour les automates programmables. Le système d'exploitation appelle l'OB1 cycliquement et déclenche ainsi le traitement cyclique du programme utilisateur.
Alarmeshoraires	OB10 à OB17	Elles peuvent être déclenchées une seule fois à un moment donné ou périodiquement avec indication du commencement et de la fréquence de répétition.
Alarmes temporisées	OB20 à OB23	Elles permettent l'exécution retardée de certaines parties du programme utilisateur. Ce retard doit être défini précédemment.
Alarmes cycliques	OB30 à OB38	Elles sont déclenchées à des intervalles de temps précis. La période de déclenchement est toujours un multiple entier de la période de base de 1 ms.
Alarmes de processus	OB40 à OB47	Elles réagissent à des signaux provenant des modules (SM, CP ou FM).
Alarme DPV1	OB55 à OB57	Les esclaves DPV1 (modules décentralisés) peuvent déclencher des alarmes de diagnostic, de processus et de débrogage.
Alarme multiprocesseur	OB60	Le mode "multiprocesseur" correspond au fonctionnement simultané de plusieurs CPU dans un châssis central.
Alarmes de synchronisme d'horloge	OB61 à OB64	Elle permet la synchronisation du programme utilisateur contenu dans une CPU avec d'autres programmes répartis dans un réseau.

Erreur de redondance	OB70 et OB72	Elle est appelée lorsqu'une perte de redondance se produit sur le réseau PROFIBUS DP, perte de redondance des CPU, erreur desynchronisation ou erreur dans un module SYNC.
Erreurs asynchrones	OB80 à OB87	Elles sont appelées lorsqu'apparaît : une erreur de temps (le dépassement du temps de cycle maximal), une défaillance dans l'appareil de base ou dans un appareil d'extension ou un débrochage/enfichage (erreur sur l'interface au réseau MPI).
Cycle en arrière-plan	OB100 à 102	On distingue entre les modes de mise en route suivants: a. redémarrage (n'existe pas pour les S7-300 et S7-400H) : OB101 b. démarrage à chaud (le seul possible pour S7-300) : OB100 c. démarrage à froid : OB102
Erreurs synchrones	OB121 et OB122	Elles sont appelées lorsqu'apparaît une erreur de programmation (des temporisations adresséesmanquent...) ou lorsqu'une opération STEP 7 accède à une entrée ou une sortie d'un module de signaux à laquelle aucun module n'était associé lors du dernier démarrage.

Tableau II. 3. Les blocs d'organisation.

- **Fonctions et blocs fonctionnels :**

Fonctions (FC) : est un bloc de code sans mémoire. Les variables temporaires d'une fonction sont sauvegardées dans la pile des données locales. Ces données sont perdues à l'achèvement de la fonction. Les fonctions peuvent faire appel à des blocs de données globaux pour la sauvegarde des données.[10]

Blocs fonctionnels (FB) : sont des blocs de code associés à des blocs de données d'instance, dans lesquels sont sauvegardés les paramètres effectifs et les données statiques des blocs fonctionnels.

Puisqu'un FB possède une mémoire (bloc de données d'instance), il est possible d'accéder à tout moment à ses paramètres et ce, en un endroit quelconque du programme utilisateur.

Blocs fonctionnels système (SFB) et fonctions système (SFC) :

- Un bloc fonctionnel système (SFB) est un bloc fonctionnel intégré à la CPU S7. Comme les SFB font partie du système d'exploitation, ils ne sont pas chargés en tant que partie du programme. Comme les FB, les SFB sont des blocs avec mémoire. Nous devons donc également créer pour les SFB des blocs de données d'instance que nous chargeons dans la CPU en tant que partie du programme. [10]

- Une fonction système (SFC) est une fonction préprogrammée et intégrée dans la CPU S7. nous pouvons appeler les SFC à partir de notre programme. Comme ces fonctions font partie du système d'exploitation, elles ne sont pas chargées en tant que partie du programme. Comme les FC, les SFC constituent des blocs sans mémoire. [10]

II.4 .Différents langages disponibles dans le STEP7 :

Les langages CONT LIST LOG, permettent de programmer des blocs d'organisations 'OB', des blocs fonctionnels 'FB' et des fonctions 'FC'.

Dans SIMATIC Manager, un clic sur le dossier programmes, ensuite sur Blocs, double clic sur l'icône du l'OB1 ouvre l'application 'CONT LIST LOG', Choisissons par exemple le langage CONT, on peut distinguer trois Zones principales :

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

- Zone 2 : c'est la zone de déclarations où on peut définir les variables locales, les variables d'entrées et sorties ou arguments pour les fonctions et les blocs fonctionnelles.

- Zone 3 : est la zone d'édition, elle est structurée en réseaux, chaque réseau finit par une affectation ou un saut.

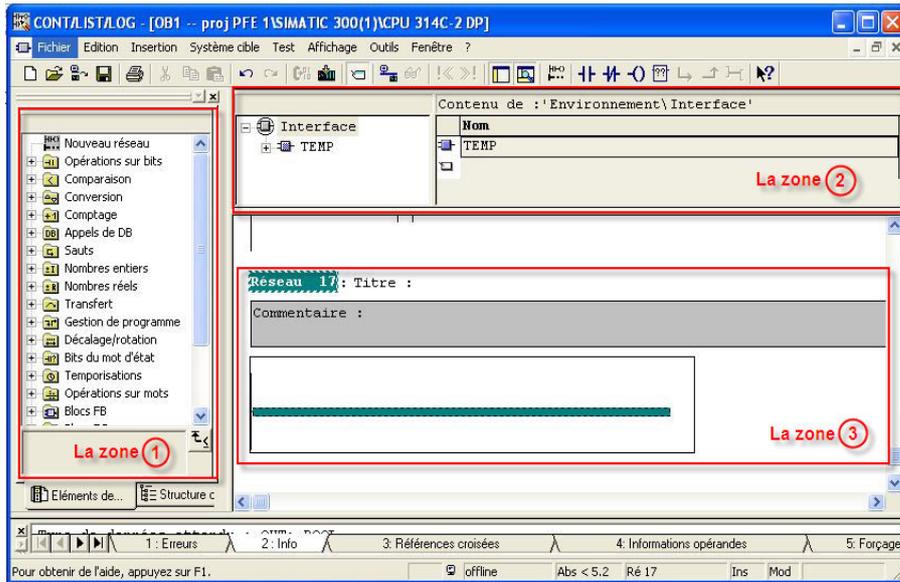


Figure II. 15. Editeur de programme CONT, LIST, LOG.

II.4.1.CONT : Le schéma a contacts est un langage de programmation graphique. La syntaxe des instructions réels fait penser aux schémas de circuits électriques. Le langage CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines. [5]

Comme nous montre la figure suivante :

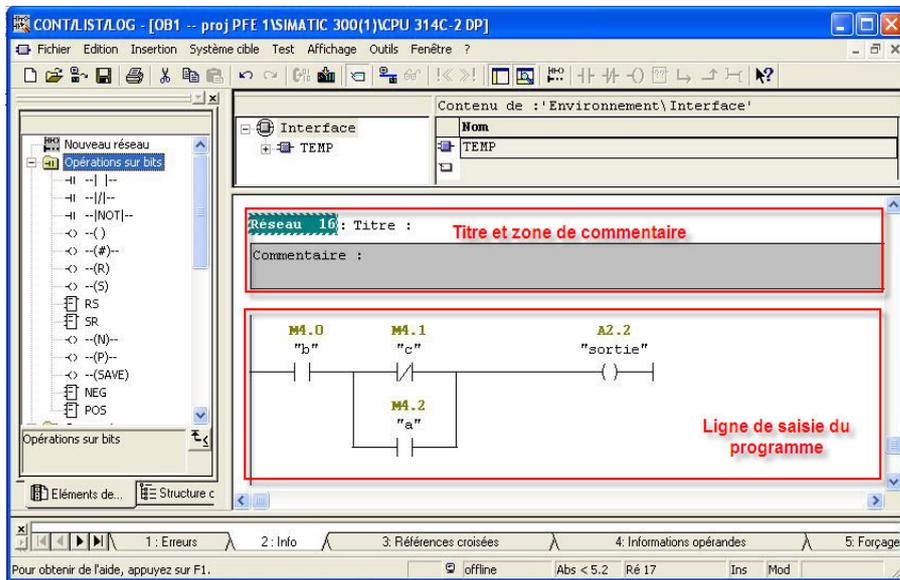


Figure II. 16. Exemple d'un programme en CONT.

II.4.2.LIST :

La liste d'instructions est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme :

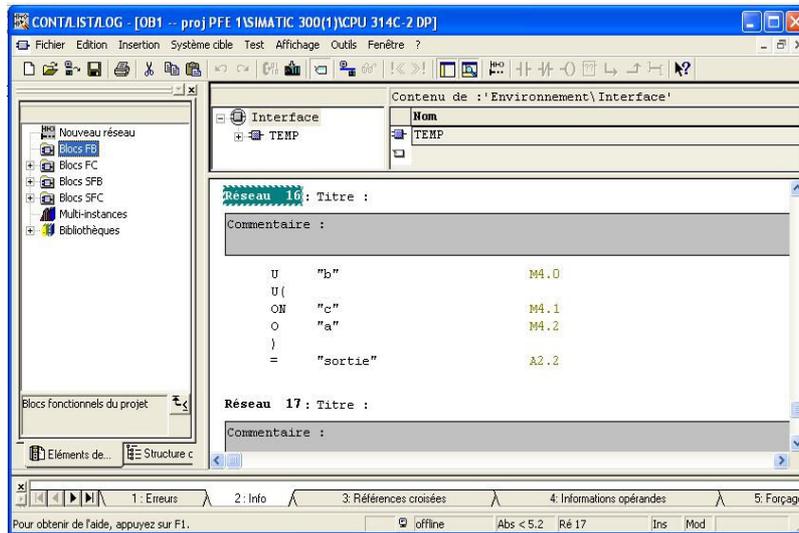


Figure II. 17. Exemple d'un programme en LIST.

II.4.3.LOG :

Le logigramme est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques, comme le montre la figure suivante :

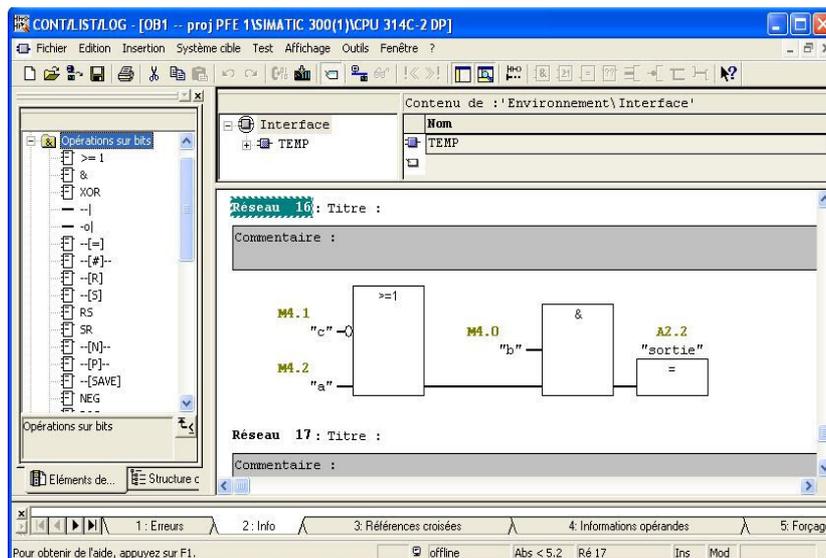


Figure II. 18. Exemple d'un programme.

II.5.Simulation des modules :

S7-PLCSIM est une application qui permet d'exécuter et de tester le programme de l'utilisateur élaboré dans un automate programmable et simulé dans l'ordinateur ou à travers une console de programmation. La simulation étant complètement réalisée par le logiciel STEP7, il n'est pas nécessaire d'établir une liaison avec un matériel S7 quelconque (CPU ou modules de signaux).

L'application S7-PLCSIM dispose d'une interface simple qui nous permet de visualiser, modifier et surveiller les différents paramètres utilisés dans le programme, comme activer ou désactiver des entrées. En exécutant le programme dans la CPU, on a la possibilité de mettre en œuvre les différentes applications du logiciel STEP7, par exemple, la table des variables afin d'y visualiser et forcer d'autres variables de prendre d'autres valeurs. On a aussi la possibilité de remédier à d'éventuelles erreurs.

II.5.1.Ouverture de l'application PLCSIM

Pour l'utilisation du simulateur S7-PLCSIM, on suit les étapes suivantes pour sa mise en route.

- 1- Ouvrir le gestionnaire de projet SIMATIC.

- 2- Sélectionner la commande «outils» et puis «simulation de modules» ou en cliquant sur son icône qui se trouve dans la boîte d'outils du gestionnaire de projet SIMATIC. Cela lance l'application S7-PLCSIM et ouvre une fenêtre CPU ayant l'adresse MPI par défaut

II.5.2.Chargement du programme

Pour charger le programme dans la CPU, on procède de la manière suivante :

- 1- Dans le gestionnaire de projets SIMATIC, on utilise la commande «Fichier Ouvrir projet», pour ouvrir le projet à charger.

- 2- On sélectionne le classeur « Blocs » dans la structure hiérarchique du projet.

- 3- Pour charger le classeur des blocs dans la CPU de simulation, on choisit la commande «Système cible Charger» ou cliquer sur le bouton de chargement.

II.5.3.Configuration du PLCSIM

Dans l'application S7-PLCSIM, on crée des nouvelles fenêtres pour visualiser les informations provenant de l'automate programmable de simulation et pour créer les diverses fenêtres, on procède comme suite :

1- Créer une fenêtre permettant de modifier l'état des entrées intervenant dans le programme.

- Choisir la commande « insertion entrée » où à partir de la barre d'outils.

2- Créer une fenêtre permettant de modifier l'état des sorties intervenant dans le programme.

- Choisir la commande « insertion sortie » ou à partir de la barre d'outils.

Les fenêtres utilisées dans le programme sont représentées dans la figure suivante :

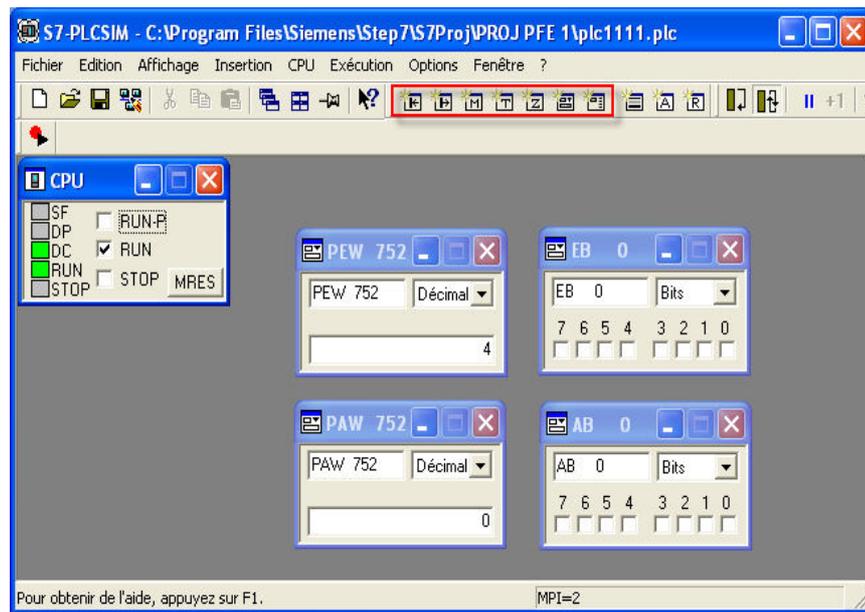


Figure II. 19.Simulation des modules.

II.5.4. Exécution du programme

Une fois l'exemple du programme chargé dans la CPU, on peut exécuter le programme. Il faut s'assurer préalablement que le cycle continu est sélectionné comme mode d'exécution. Pour sélectionner le mode d'exécution continu du programme, on choisit la commande «Exécution Mode Cycle continu», ou on clique sur le bouton correspondant dans la barre d'outils, et pour mettre la CPU en mode RUN et démarrer l'exécution du programme, on clique sur la case à cocher RUN (Marche) ou RUN P dans la fenêtre «CPU».

Dans le simulateur S7-PLCSIM, on a représenté l'ensemble des variables d'entrées, et de sorties sous forme de fenêtres. Pour visualiser le fonctionnement de l'automate, on suit les étapes de fonctionnement de la machine avec des cliques sur les entrées pour visualisation des sorties.

II.5.5. Etat de fonctionnement de la CPU

- **Etat de marche (RUN-P)**

La CPU exécute le programme et nous permet de le modifier ainsi que ses paramètres. Afin de pouvoir utiliser les applications de STEP7 pour forcer un paramètre quelconque du programme durant son exécution, on doit mettre la CPU à l'état RUN-P.

- **Etat de marche (RUN)**

La CPU exécute le programme en lisant les entrées, exécutant le programme, puis en actualisant les sorties. Lorsque la CPU se trouve à l'état de marche (RUN), on ne peut ni charger un programme, ni utiliser les applications de STEP7 pour forcer un paramètre quelconque (comme les valeurs d'entrées).

- **Etat d'arrêt (STOP)**

La CPU n'exécute pas le programme. Contrairement à l'état d'arrêt (STOP) des CPU réelles, les sorties ne prennent pas de valeurs (de sécurité) prédéfinies, mais conservent l'état auquel elles étaient lorsque la CPU est passée à l'état d'arrêt (STOP). Nous pouvons charger des programmes dans la CPU lorsqu'elle est à l'arrêt.

II.6. Conclusion :

Nous pouvons affirmer de par notre travail, que le STEP7 est l'outil indispensable à chaque projet de programmation à base d'automate Siemens.

Les différents langages présents au niveau de cet outil nous ont permis de bien nous familiariser avec les différentes applications que peut avoir l'automate. Cela nous a inspirés pour les améliorations qui peuvent être apportées à notre projet.

CHAPITRE III :

Le logiciel de conception des
interfaces Homme/Machine
Win CC flexible

III.1. Introduction :

Un maximum de transparence est essentiel pour l'opérateur, ce dernier est confronté à des environnements où les processus sont de plus en plus complexes.

La supervision est une technique industrielle qui consiste à représenter, surveiller, et diagnostiquer l'état de fonctionnement d'un procédé automatisé afin de parvenir un fonctionnement optimal. Le but est de disposer en temps réel d'une visualisation de l'état d'évolution des paramètres du processus, ce qui permet à l'opérateur de prendre rapidement les décisions appropriées. Permettre d'atteindre des objectifs tels que la cadence de production, qualité des produits et sécurité des biens et des personnes.

L'objectif de ce chapitre est la présentation du logiciel d'ingénierie et de supervision SIMATIC WinCC flexible. Ce dernier est un système de supervision de processus modulable qui offre des fonctions performantes de surveillance d'automatismes.

III.2. Présentation de WinCC

III.2.1. Description générale :

Le WinCC flexible est un système HMI (Human Machine Interface) performant qui est utilisé sous Microsoft Windows 2000 et Windows XP. Il s'agit donc de l'interface entre l'homme (l'utilisateur) et la machine (le processus). Le contrôle proprement dit du processus est assuré par les automates programmables (API). Une communication s'établit entre WinCC et l'opérateur d'une part et entre WinCC et les automates programmables d'une autre part (figure III.1).

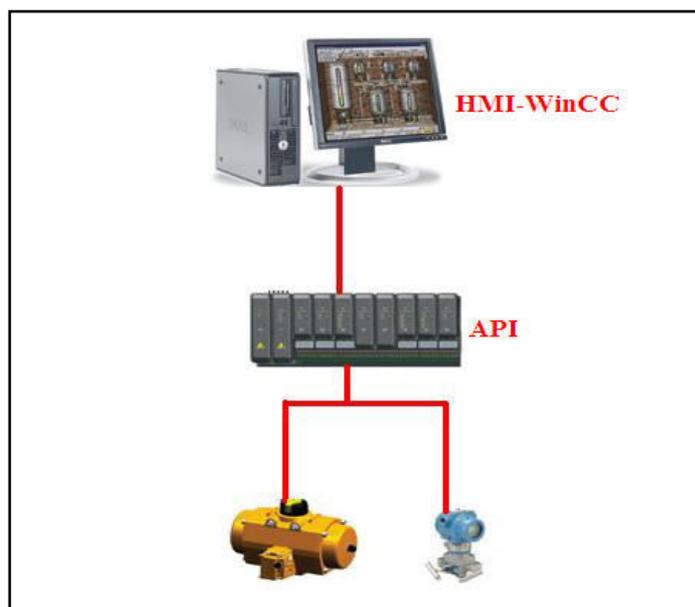


Figure III.1. Communication entre le WinCC, l'API et le processus.

Ce logiciel nous permet de visualiser le processus et de concevoir l'interface graphique destinée à l'opérateur. Il nous permet donc :

- La Surveillance : le processus étant visualisé par un graphisme à l'écran. Qui est mis à jour suivant l'évolution du processus.
- La Commande : l'opérateur peut entrer des valeurs de consigne dans l'interface.
- Le Déclenchement des alarmes : une alerte apparaît dans le cas de franchissement d'un des seuils définis.
- Imprimer et archiver sur support électronique les alarmes et les valeurs des processus. Ceci nous permet de documenter la marche du processus et d'avoir accès ultérieurement aux données de production du passé.

III .2.1.Fonctionnement de WinCC flexible :

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

- **WinCC flexible Engineering System** : est le logiciel avec lequel nous réalisons toutes les tâches de configuration requises, édition du projet de supervision.
- **WinCC flexible Runtime** : est le logiciel de visualisation de processus en temps réel. Le projet étant en mode processus, on ne pourra parler que de son exécution.

III .2.2.Eléments de l'interface utilisateur de WinCC flexible :

L'environnement de travail de WinCC flexible se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers qui ne sont visibles que lorsque cet éditeur est activé.

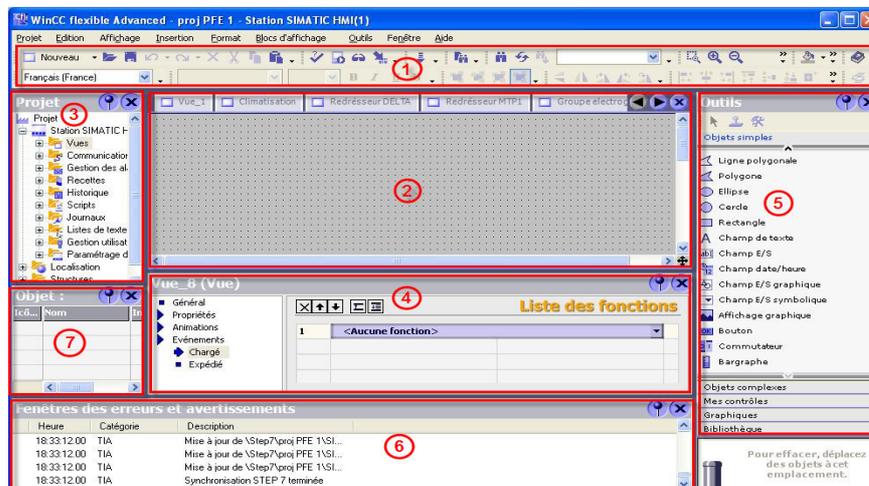


Figure III. 2.Eléments de l'interface de WinCC flexible.

1) Menus et barres d'outils :

Nous trouverons dans les menus et les barres d'outils toutes les fonctions dont nous avons besoin pour la configuration de notre pupitre opérateur. Lorsqu'un éditeur est actif, les commandes de menu ou barres d'outils y correspondant sont visibles.

2) Zone de travail :

Elle sert à éditer les objets du projet. Tous les éléments de WinCC flexible sont disposés autour de cette zone, à l'exception de la zone de travail, nous pouvons disposer et configurer, déplacer ou masquer tous les éléments comme bon nous semble.

3) Fenêtre de projet :

Tous les éléments et tous les éditeurs disponibles d'un projet sont affichés sous forme d'arborescence dans la fenêtre du projet et peuvent être ouverts à partir de celle-ci. Sous chaque éditeur se trouvent les dossiers, dans lesquels on peut faire un stockage structuré des objets.

Les vues, les recettes, les scripts, les journaux et les dictionnaires personnalisés, sont ce qu'on appelle des objets nous pouvons en outre leur y accéder directement.

4) Fenêtre des propriétés :

Elle permet de modifier les propriétés d'un objet sélectionné dans la zone de travail, le contenu de cette fenêtre dépend de l'objet sélectionné.

5) Boîte à outils et Bibliothèque:

La fenêtre d'outils nous propose un choix d'objets que nous pouvons insérer dans nos vues, par exemple des objets graphiques et éléments de commande. La fenêtre d'outils contient en outre des bibliothèques d'objets et collections de blocs d'affichage prêts à l'emploi.

La bibliothèque fait partie de la fenêtre d'outils, elle nous donne accès aux objets de vue préconfigurés. Et nous permet d'augmenter la quantité d'objets de vue disponibles.

6) Fenêtre des erreurs et avertissements :

La fenêtre des erreurs et avertissements affiche les événements système générés lors de la compilation du programme. Dans cette fenêtre les événements système sont affichées par défaut dans leur ordre d'apparition.

7) Fenêtre des objets :

La fenêtre des objets affiche tous les éléments de la zone que nous avons utilisés dans la fenêtre de projet.

III .3. Etapes de conception d'un projet sous WinCC flexible :

III .3.1. Intégration du projet WINCC flexible à STEP7 manager :

Les variables du processus représentent la liaison pour la communication entre le système d'automatisation et le système IHM. Sans l'utilisation de l'étape d'intégration du projet WinCC, nous devrions définir chaque variable à deux reprises : une fois pour le système d'automatisation et une fois pour le système IHM.

L'intégration du projet WINCC dans l'interface de configuration permet de diminuer la fréquence des erreurs et de réduire les tâches de configuration nécessaires. Durant la configuration, nous accédons directement à la table des mnémoniques de STEP 7 ainsi qu'aux paramètres de communication (figure III.3).

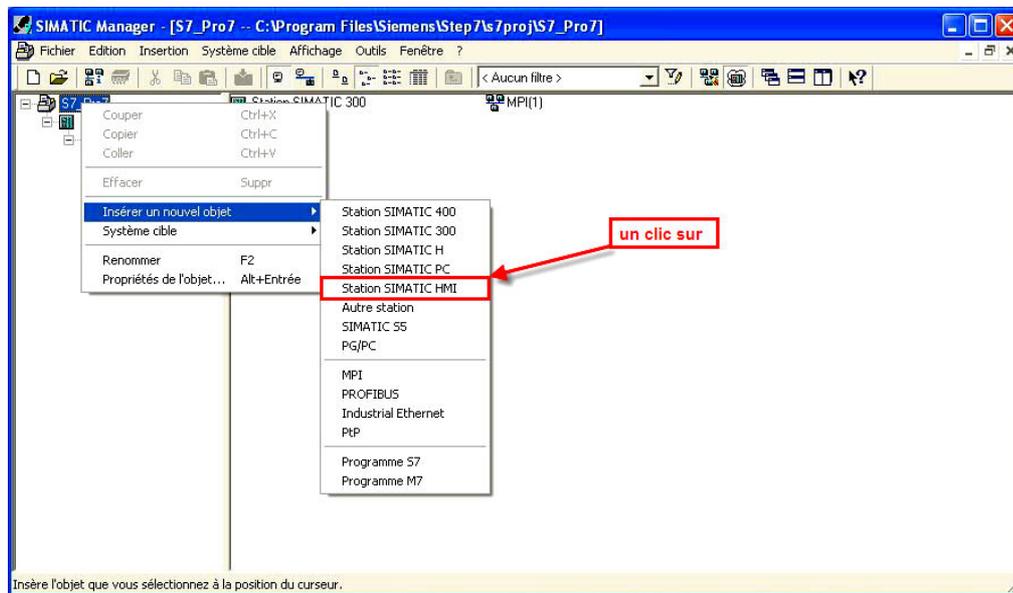


Figure III. 3. Intégration du projet WINCC flexible à STEP7 manager

III .3. 2. Création de la liaison entre le projet HMI et l'API

La communication entre les pupitres opérateur et les automates SIMATIC S7 peut être réalisée via les réseaux suivants :

- MPI (Multi point Interface) : Le pupitre opérateur est connecté à l'interface MPI de l'automate. Il est possible de connecter plusieurs pupitres opérateurs à un automate SIMATIC S7;
- PROFIBUS (processus Field Bus) ;
- Ethernet.

L'éditeur "Liaisons" affiche la connexion à l'automate configurée (Figure III.4):

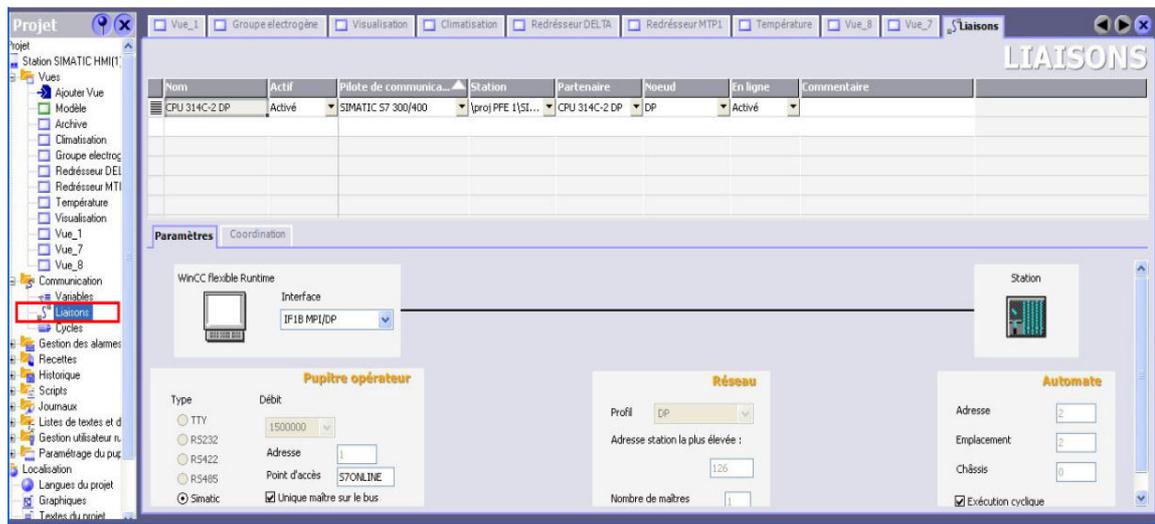


Figure III. 4.Liaison entre le pupitre et l'automate.

III .3. 3. Définition des variables :

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

Les variables externes : permettent d'échanger des données entre les composants d'un processus automatisé, et un pupitre operateur. 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 operateur que de l'automate.[11]

Les variables internes : ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre. Le pupitre operateur en question peut donc accéder en lecture et en écriture aux variables internes.

Elles sont créées par exemple pour exécuter des calculs locaux à WinCC.[11]

La déclaration des variables s'effectue dans l'éditeur "Variables" (figure III.5). A leur déclaration,une configuration de base leur est attribuée. L'éditeur "Variables" nous permet d'adapter cette configuration aux besoins de notre projet ; L'éditeur s'ouvre systématiquement par ajout de variable.

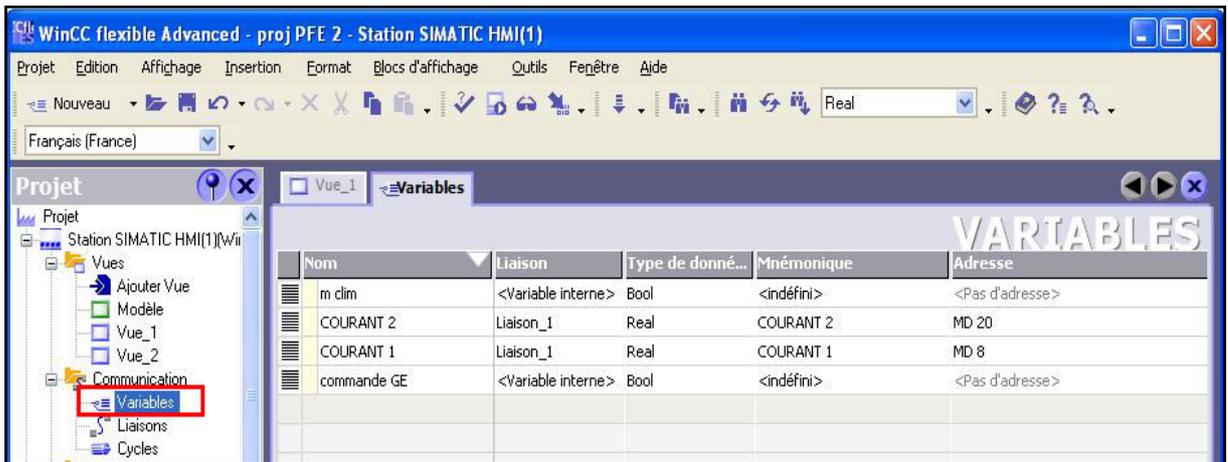


Figure III. 5.Éditeur de variable dans le Win CC flexible.

III .3. 4. Création des vues :

Dans WinCC flexible, on crée des vues pour le contrôle-commande de machines et d'installations. Pour créer des vues, on dispose d'objets prédéfinis permettant de représenter notre installation, d'afficher des procédures et de définir des valeurs de processus.

Dans la fenêtre du projet, on fait un double-clic sur l'entrée "Ajouter Vue" dans le groupe "Vues". Une nouvelle vue s'ouvre dans la zone de travail (figure III.6).

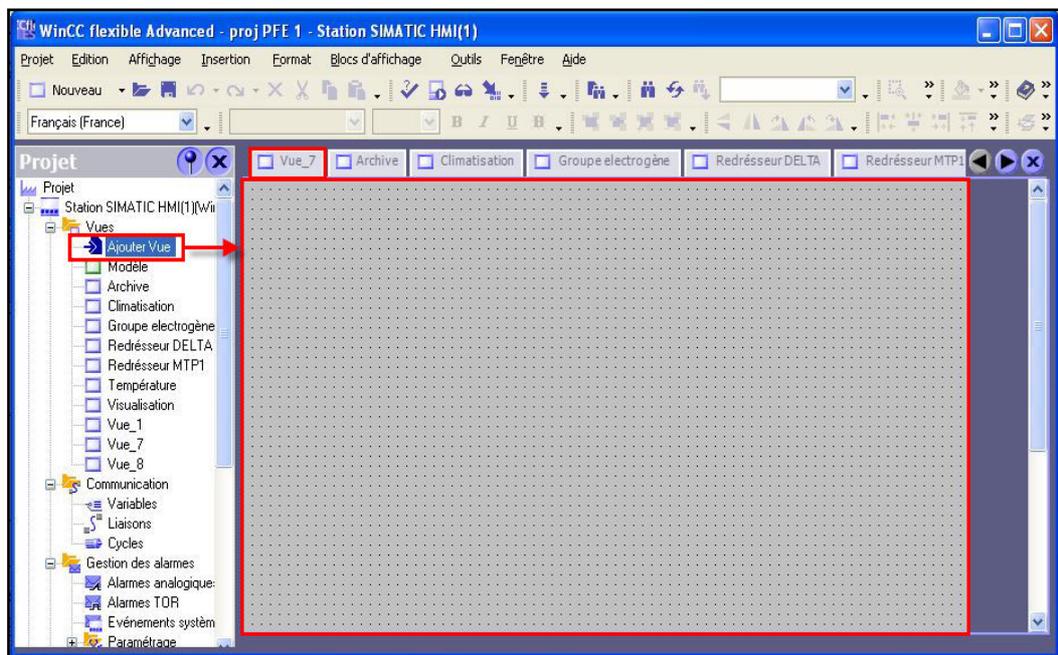


Figure III. 6. Création d'une vue.

III .3. 4.1. Les objets de l'éditeur "Vues" :

Les objets sont des éléments graphiques, qui permettent de configurer la présentation des vues duprocessus. La fenêtre des outils contient différents types d'objets. Tant que l'éditeur

"Vues" est ouvert, la fenêtre des outils propose des objets dans le groupe d'objets "Objets simples". Les objets simples sont détaillés dans (Annexe B).

III .3. 5. Gestion des alarmes :

Les alarmes montrent les événements où les anomalies qui peuvent survenir dans le processus, ces alarmes peuvent servir par exemple, à diagnostiquer les erreurs. Les alarmes utilisées sont des alarmes de type tout ou rien (TOR) ou de type analogique.

III .3. 5.1. Classes d'alarmes prédéfinies dans WinCC flexible

- "Erreurs" pour les alarmes TOR et analogiques signalant des états critiques ou dangereux du fonctionnement et du processus. Les alarmes de cette classe doivent toujours être acquittées.

- "Avertissements" pour les alarmes TOR et analogiques signalant des états normaux du fonctionnement et du processus ainsi que des déroulements du processus. Les alarmes de cette classe n'ont pas besoin d'être acquittées.

- "Système" pour les événements système qui informent sur les états de fonctionnement du pupitre opérateur et des automates. Cette classe d'alarmes ne peut pas être utilisée pour les alarmes définies par l'utilisateur.

- "Diagnostic" Les événements de diagnostic indiquant des états et des événements des automates SIMATIC S7 et SIMOTION. [12]

III .3. 5.2. Éditeur "Alarmes TOR"

L'éditeur "Alarmes TOR" permet de créer des alarmes de bit et d'en définir les propriétés.

D'abord on crée une variable interne dans l'éditeur "variable" de type "Int", celui-ci un mot dont lequel, on peut sauvegarder chaque bit des alarmes TOR.

Par exemple un mot (MW 0) peut contenir 16 alarmes de bit de M 0.0 à M 0.7, M 1.0 à M 1.7.

Puis on Effectue un double-clic dans la fenêtre de projet sur "Alarmes TOR"(Figure III.7).

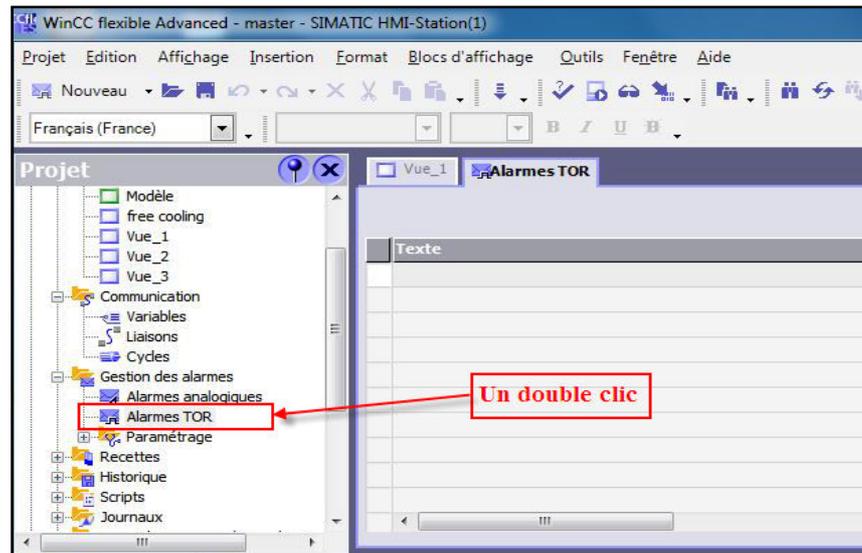


Figure III. 7. Configuration des alarmes TOR (1).

La zone de travail affiche toutes les alarmes TOR sous forme du tableau.

Effectuant un double clic sur une case vide du tableau pour créer l’alarme (étape 1) , en suite on édite les propriétés des alarmes de bit (étape 2) directement dans les cellules du tableau, choisissant le texte d’alarme, le numéro, la classe (erreurs, avertissement..) et la variable de déclenchement de cette alarme, qui est sauvegardée dans le mot (MW) qu’on a créé auparavant .



Figure III. 8. Configuration des alarmes TOR (2).

Pour afficher les alarmes TOR dans une vue en Runtime, la fenêtre des outils nous propose des objets dans le groupe "Objets complexes". On glisse l'objet 'vue des alarmes' sur la zone de travail (Figure III.9), ce qui nous permet de configurer l'affichage de la vue d'alarme TOR dans la fenêtre de propriété.

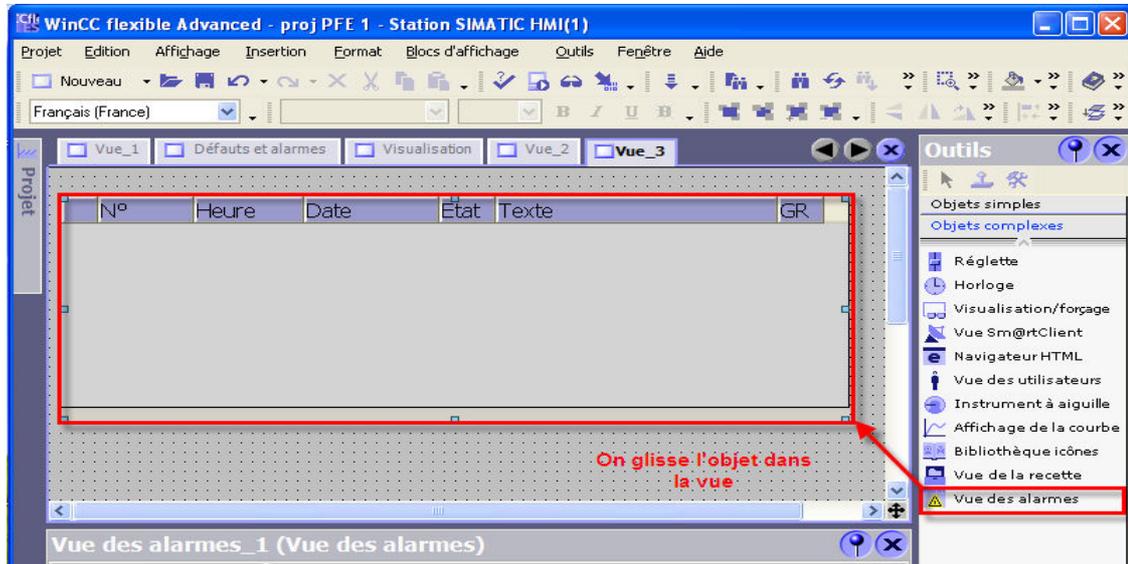


Figure III. 9. Configuration des alarmes TOR (3).

III .3. 5.3. Éditeur "Alarmes Analogiques" :

Cet éditeur permet de créer des alarmes analogiques et d'en définir les propriétés. Effectuons un double-clic (étape 1) dans la fenêtre de projet sur "Alarmes Analogique" (Figure III.9).

La zone de travail affiche toutes les alarmes analogiques sous forme de tableau. On éditant les propriétés des alarmes analogiques directement dans les cellules du tableau, on spécifie les paramètres suivant (étape 2) :

- **Texte** : Le texte du message est affiché au runtime.
- **Numéro** : numéro unique servant à l'identification de l'alarme (valeurs admissibles : de 1 à 32767)
- **Classe** : Nom de la classe d'alarmes à laquelle appartient l'alarme (p. exemple erreur, avertissement)
- **Variable de surveillance** : Cette variable fait l'objet d'une surveillance de limites.
- **La valeur limite** : c'est le seuil de déclenchement de l'alarme, elle peut être indiquée directement sous forme de constante ou indirectement sous forme de variable.

- **Déclencheur** : c'est la méthode de surveillance des valeurs limites. On peut choisir "Si front montant": l'alarme sera déclenchée en cas de dépassement de la valeur limite vers le haut, Ou "Si front descendant" l'alarme sera déclenchée en cas de dépassement de la valeur limite vers le bas.

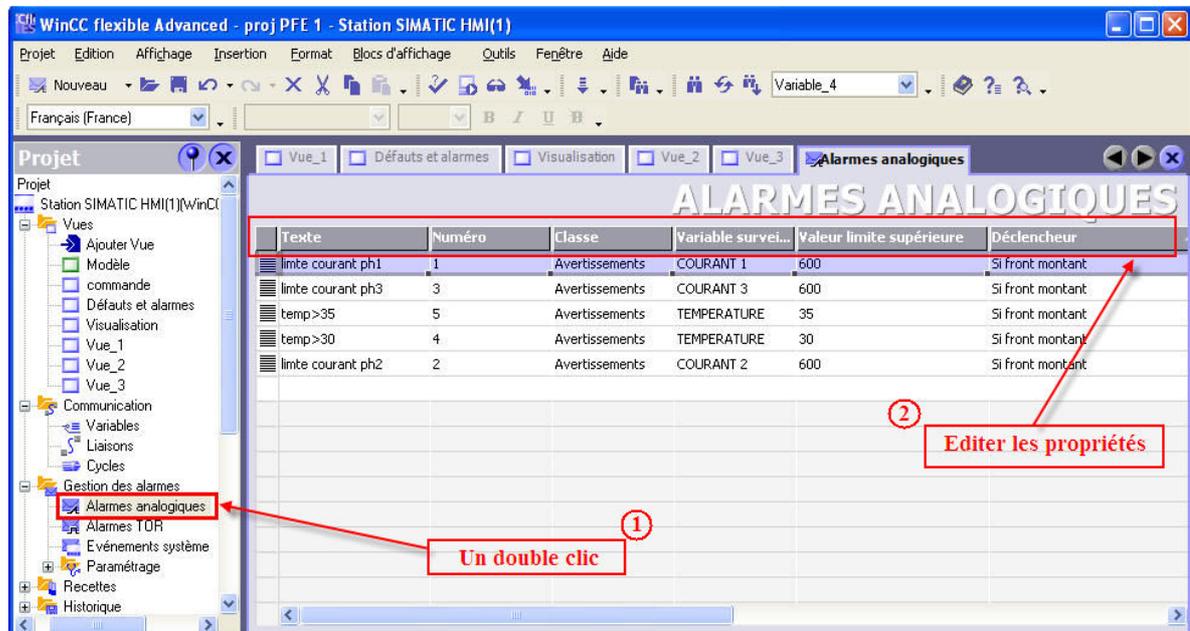


Figure III. 10. Configuration des alarmes analogiques (1).

La méthode d'affichage des alarmes analogiques dans une vue est la même que celle des alarmes TOR, nous insérant l'objet complexe 'vue des alarmes' sur la zone de et nous configurons l'affichage des alarmes analogiques dans la fenêtre propriétés.

III .3. 6 Configuration de la vue courbe d'une variable analogique :

Une courbe est la représentation graphique des valeurs d'une variable analogique dans le Runtime. Pour visualiser des courbes, nous configurons l'affichage de ces dernières dans la vue de projet et nous y définissons leurs types :[11]

- Archive : permet la visualisation des valeurs archivées d'une variable.
- Déclenchement sur impulsion temps réel : permet une visualisation des valeurs sur déclenchement horaire.
- Déclenchement sur bit temps réel : permet une visualisation des valeurs déclenchée sur événement.
- Déclenchement sur bit historique : permet une visualisation déclenchée sur événement avec acquisition de données par tampon.

Pour configurer l'affichage d'une courbe on suit les étapes suivantes :

1. Glisser l'objet "Vue de courbe" de la fenêtre d'outils(objets complexes) dans la vue.
2. Cocher dans la fenêtre des propriétés l'entrée "Propriétés" et "Courbes". La boîte de dialogue "Courbes" s'ouvre.
3. Attribuer un nom à la courbe dans la colonne "Nom".
4. Dans la colonne "Visualisation", on choisit la représentation de la courbe soit par "Lignes", "échelons", "Bargraphes" ou "points".
5. Dans la colonne "Type de ligne", on peut sélectionner entre "Massif", "Tiret" ou "Solide".
6. Dans la colonne "Type de courbe", on choisit le type de déclenchement de la courbe "Archive", "Déclenchement sur impulsion temps réel", "Déclenchement sur bit temps réel" ou "Déclenchement sur bit historique".
7. Dans la colonne "Paramètres source", on clique sur la flèche. Une boîte de dialogue s'ouvre.
8. Sélectionnez comme "Variable de courbe" la variable dont les valeurs doivent être affichées.
9. Dans la colonne "coté", on choisit le sens de l'axe de la courbe.
10. A la dernière colonne, on choisit la Couleur de la courbe.

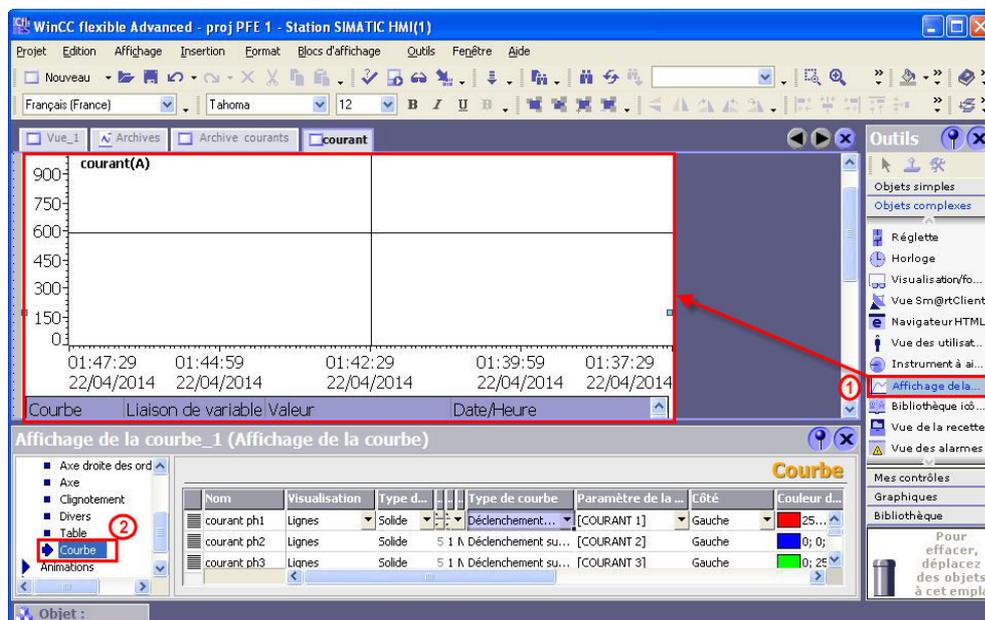


Figure III. 11. Insertion et configuration d'une vue courbe.

III .3. 7 Editeur de "Scripts" :

Le créateur de scripts est une interface de programmation avec laquelle on accède en runtime à une partie des données du projet. Dans l'utilisation de l'éditeur de scripts on s'adresse à des configureurs ayant des connaissances en Visual Basic (VB) et Visual Basic Script (VBS).

L'éditeur de Scripts nous assiste au cours de la programmation à l'aide de fonctionnalités telles que "l'IntelliSense" et "le codage couleur syntaxique". En outre, nous pouvons par exemple référencer des variables déjà déclarées par glisser-déplacer.

Dans l'éditeur de Scripts, les mots-clés sont repérés par différentes couleurs. Les instructions sont affichées en gras. Les mots inconnus sont soulignés par un trait rouge ondulé (figure III.11).

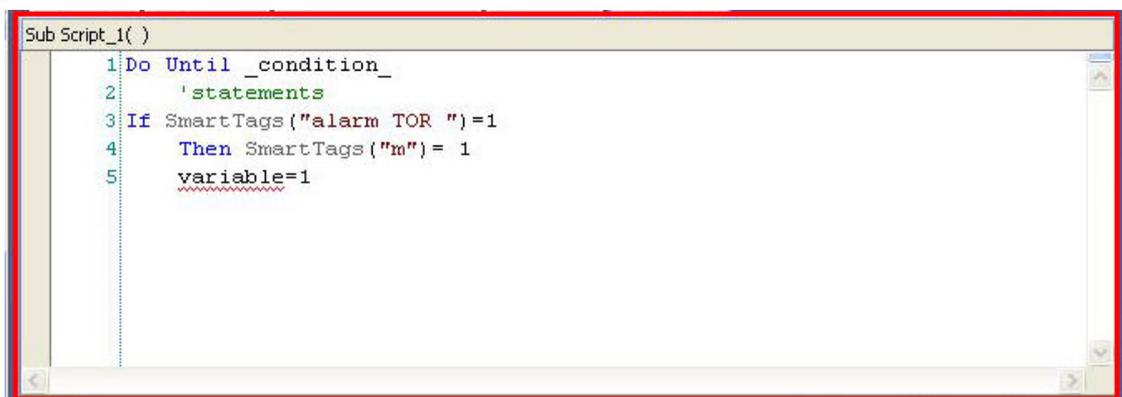


Figure III. 12. Editeur du script (1).

III .3. 7.1 : Accès aux variables :

Dans le script, nous avons accès aux variables externes et internes créées dans le projet. En runtime, nous pouvons lire ou modifier les valeurs de ces variables.

Si le nom d'une variable du projet est conforme aux conventions de noms VBS, nous pouvons utiliser directement celle-ci dans le script :

```
'VBS_Example
```

```
    If  TemperatureSalle > 100
```

```
Then    [Alarme1]
```

Si le nom de la variable dans le projet n'est pas conforme aux conventions de noms VBS, nous devons référencer la variable par la liste des variables du système "SmartTags", par un clic droit sur la zone d'édition du script (étape 1). Et on choisit la variable à utiliser (étape 2). (Figure III.13.)

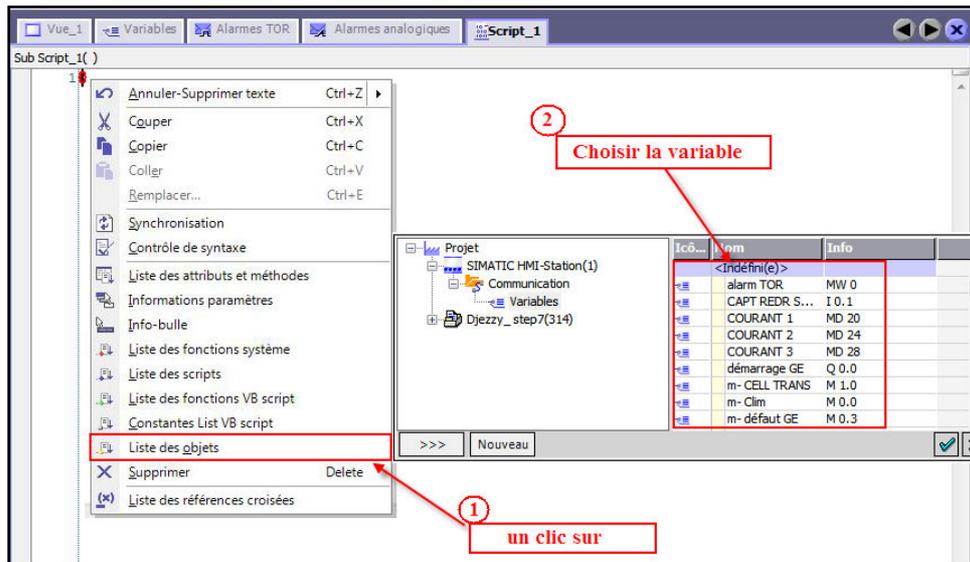


Figure III. 13. Editeur de script(2)

Aussi nous pouvons définir grâce à l'instruction "Dim" des variables locales, celles-ci ne peuvent être utilisées qu'à l'intérieur du script. Elles n'apparaissent donc pas dans l'éditeur "Variables".

Par exemple elles peuvent être utilisées comme compteur dans une instruction "For" :

'VBS_Example

Dim intCount

For intCount = 1 To 10 [demarreMoteur]Next

III .3. 7.2. Utilisation des boucles :

Dans un script, on peut insérer des boucles codes (instructions) prédéfinies dans l'éditeur de script. A partir de l'onglet " Assistant de Script", on glisse la boucle qu'on veut utiliser dans l'éditeur (Figure III.14):

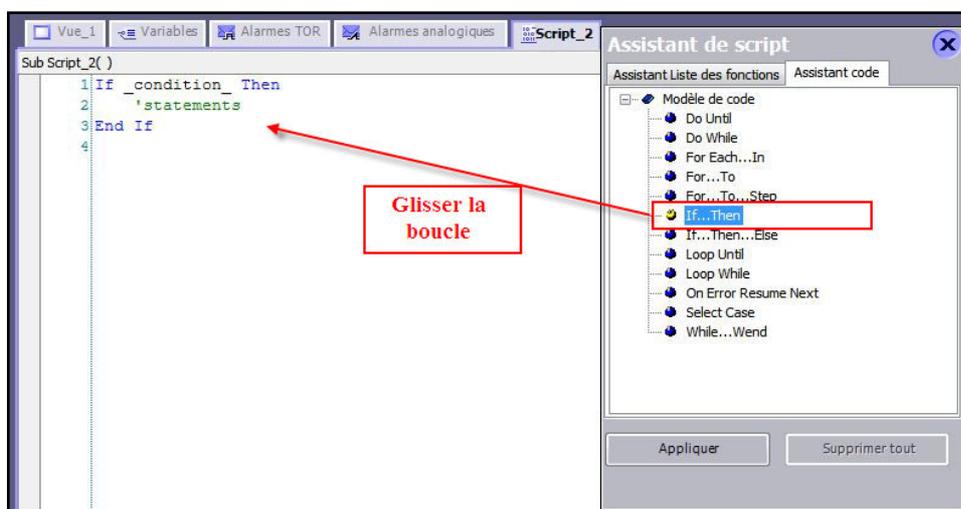


Figure III. 14. Editeur de script(3)

III .3. 8. Editeur de "Recettes" :

Les recettes regroupent des données correspondantes, comme le paramétrage d'une machine ou des données de production. Par exemples :

- Des paramétrages de machine qui sont nécessaires pour adapter la production à une autre variante de produit,
- Des composants qui donnent des produits finaux suivant les proportions de leur combinaison.

On peut créer une recette à partir de la fenêtre "Projet " >>"Ajouter recette ", choisissant comme exemple une recette "Jus d'orange" avec les proportions de mélange correspondantes pour le pur jus, le nectar et la boisson aux fruits.

Les différents éléments de la recette sont définis les paramètres :

- Le nom ainsi que le numéro de la recette l'identifie de façon univoque dans le pupitre opérateur.
- Nom d'affichage on peut donner des noms explicites ou des désignations qu'on peut associer directement à un produit.
- La version identifie la date et l'heure de la dernière modification apportée à la recette.
- Variable affectée : chaque élément de recette est associée à une variable de recette dans laquelle la valeur de l'enregistrement est enregistrée au runtime.
- La valeur par défaut : elle est utilisée comme entrée par défaut lorsqu'on crée un enregistrement de recette.
- Décimale : détermine le nombre de décimales affiché pour la valeur de l'enregistrement de recette au runtime.

Nom	Nom d'affichage	Variable	Liste de textes	Valeur par défaut	Décimales	Info-bulle
Sucre	Kilo sucre	kilo sucre	<indéfini>	0	0	
Aromes	Gramme aromes	gramme aromes	<indéfini>	0	0	
Concentré	Litre concentré	litre concentré	<indéfini>	0	0	
Eau	Litre eau	litre eau	<indéfini>	0	0	

Figure III. 15.Eléments de la recette.

Dans l'onglet "Enregistrements" s'effectue l'attribution des valeurs des différents éléments pour chaque variante de la recette, celle-ci est caractérisé par l'association des quantités requise des ingrédients la constituant. Aussi une case commentaire existe afin de préciser s'il y a une procédure spécifique de préparation à cette recette.

Nom	Nom d'affichage	Numéro	Sucre	Aromes	Concentré	Eau	Commentaire
pur jus	pur jus	1	5	50	50	50	
nectar	nectar	2	10	100	70	30	
boisson aux fruits	boisson aux fruits	3	8	40	30	70	

Figure III. 16. Les enregistrements de la recette.

Après la configuration des paramètres de la recette, on insert à partir de la fenêtre "Objets">>"Objets complexes" l'élément "Vue de la recette" dans la zone de travail. Et on édite les propriétés de l'affichage de la vue recette comme suite :

- Choisir le nom de la recette enregistré
- Enregistrer le numéro, le nom ou l'enregistrement d'une recette dans une variable.
- Choisir la permission d'éditer les valeurs d'enregistrements d'une recette en runtime ou pas.
- Choisir le type d'affichage de la vue recette (simple ou étendu).

Pour pouvoir modifier les valeurs des enregistrements de recette dans les champs d'E/S configurés hors de la vue simple de recette, la case "Synchronisation de variables" doit être activé dans les propriétés de la recette. Finalement la vue étendue de la recette s'affiche comme suite :

Nom de recette :	N°:
jus d'orange1	2
Nom d'enregistrement :	N°:
boisson aux fruits	3
Nom du constituant	Valeur
Kilo sucre	8
Gramme aromes	40
Litre concentré	30

Prêt

Figure III. 17: Vue de recette (étendue)

Les valeurs affichées ou saisies dans la vue de la recette sont enregistrées dans les "enregistrements" de recette. Ces enregistrements sont échangés avec l'automate au moyen de fonctions système.

Dans la vue de la recette, il est possible de configurer les objets de commande suivants:

Bouton	Fonction
	Le texte d'aide configuré s'affiche.
	Un nouvel enregistrement de recette est créé. Quand une valeur initiale a été configurée, elle s'affiche dans le champ de saisie.
	Les valeurs affichées de l'enregistrement de recette sont enregistrées.
	L'enregistrement de recette est enregistré sous un autre nom, indépendamment de la vue de la recette. Une boîte de dialogue s'ouvre pour la saisie du nom.
	L'enregistrement de recette affiché est supprimé.
	<ul style="list-style-type: none"> • Si la valeur d'une variable de recette possède une valeur plus récente que la vue de la recette, cette valeur est reprise dans la vue de la recette. • Si la valeur affichée dans la vue de la recette est plus récente que la valeur des variables de recette, cette valeur est reprise dans la variable de recette. <p>Pour que cette fonction puisse être utilisée, il faut que la fonction "Synchronisation variables" soit activée dans les propriétés de la recette.</p>
	Les valeurs de l'enregistrement réglé affichées dans la vue de la recette sont transférées dans l'automate.
	Les valeurs de recette de l'automate s'affichent dans la vue de la recette

Tableau III. 1. Boutons commande de la vue recette.

III .3. 9. Archivage des données dans WinCC flexible :

L'archivage des variables sert à saisir, traiter et enregistrer les données du processus d'une installation industrielle. Ces données archivées qui documentent le déroulement de la production, peuvent être utilisées pour analyser les modes de défaillance, optimiser les cycles de maintenance et assurer le respect des critères de qualité.

Dans WinCC flexible les valeurs des variables internes et externes peuvent être enregistrées dans des archives de variables. Il est possible de spécifier pour de nombreuses variables l'archive dans laquelle elles seront enregistrées.

En runtime, les valeurs de variables à archiver sont acquises, traitées et enregistrées dans une base de données ou dans un fichier.

On peut choisir parmi les lieux d'archivage suivants :[11]

- La base de données ODBC : L'archivage s'effectue au niveau du PC seulement, donc on peut l'exploiter.
- Un fichier - CSV (ASCII) : On peut lire ou exploiter les données archivées dans ce type de fichier sans WinCC flexible Runtime.
- Un Fichier – RDB : Ce lieu de stockage a une performance de lecture élevée, il ne peut être lu ou représenté qu'avec WinCC flexible.

La fenêtre propriété de l'archivage nous offre plusieurs méthodes d'archivage :

- Archive cyclique : Lorsque l'archive est remplie, les entrées les plus anciennes sont écrasées.
- Archive cyclique segmentée : Création de plusieurs archives de même taille, qui sont remplies à mesure. Lorsque toutes les archives sont entièrement remplies, l'archive la plus ancienne est écrasée.
- Afficher les alarmes système : Affichage d'une alarme système lorsqu'un niveau de remplissage donné est atteint.
- Déclencher l'événement : Dès que l'archive est pleine, l'événement "Débordement" se déclenche.

III .3.9.1. Archive variable :

On peut créer une archive variable à partir de la fenêtre "Projet">>"Historique", pour archiver par exemple une variable analogique, et afficher celle-ci dans une vue courbe.

D'abord on crée l'archive dans le tableau " archive " (étape1) qui s'affiche dans la zone de travail. Ce tableau nous permet de définir (étape 2):

- Le nom de l'archive.
- Le nombre d'enregistrement par archive.
- Le lieu et le chemin d'archivage.
- La méthode d'archivage.
- Le niveau de remplissage de l'archive de variable.

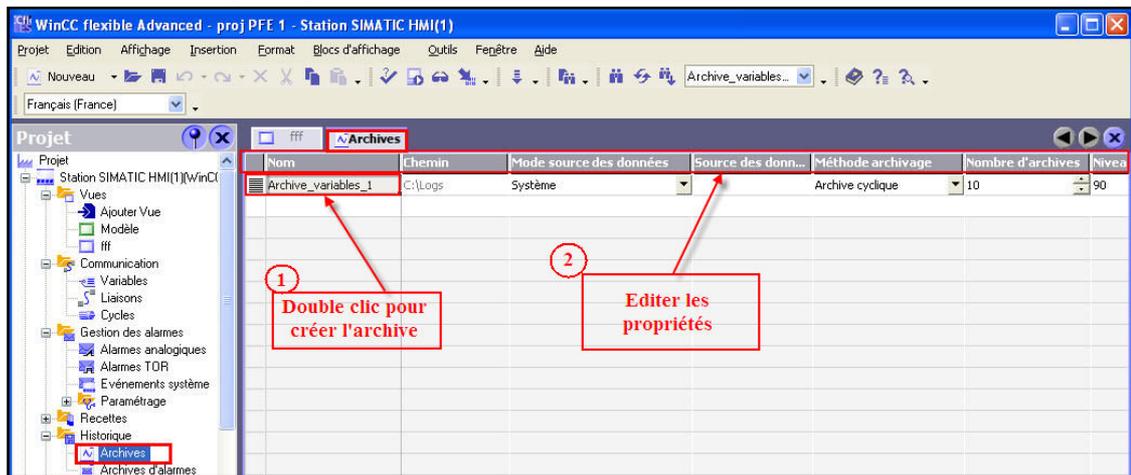


Figure III. 18. Editeur archive variable.

Pour l'affichage des alarmes archivées, on insère une "vue courbe" à partir de l'onglet "Objets complexes". Ce qui est la même pour les alarmes analogiques, mais cette fois-ci on choisit dans la fenêtre "Propriété de l'affichage" le type de courbe "Archive", et le paramètre de la source "Archive variable" qu'on a créé auparavant.

En runtime, on peut déplacer et modifier la plage de temps, afin d'obtenir à partir de l'archive les informations voulues.

III .3.9.2. Archive Alarmes :

Pour accéder à l'éditeur "Archive d'alarmes", on double-clic sur "Archive d'alarmes" dans le groupe "Historique" (étape 1). Ce dernier nous permet de configurer une archive afin de stocker les classes d'alarmes et d'en définir les propriétés (étape 2).

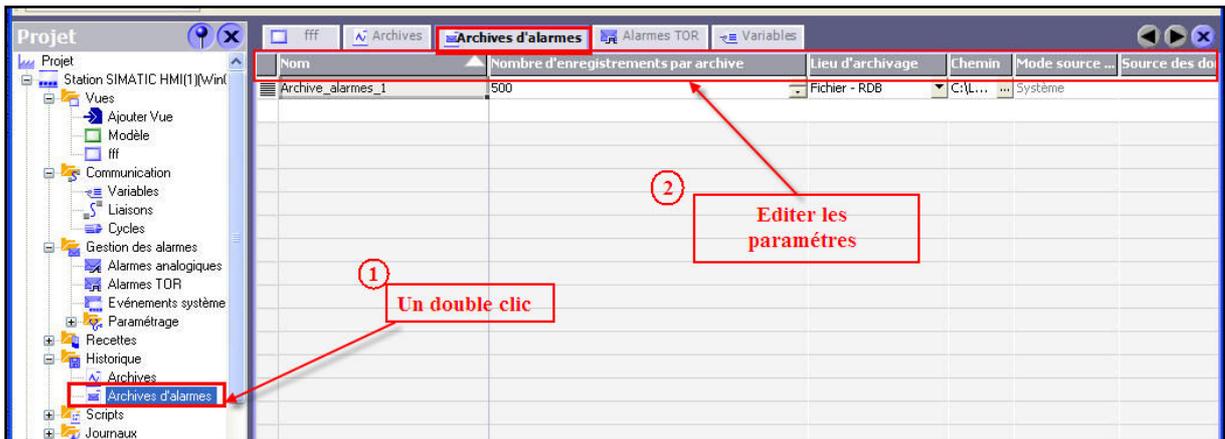


Figure III. 19. Editeur archives alarmes (1).

Pour afficher l'archive des alarmes, il faut indiquer les variables qu'on veut archiver dans le tableau des variables de WinCC.

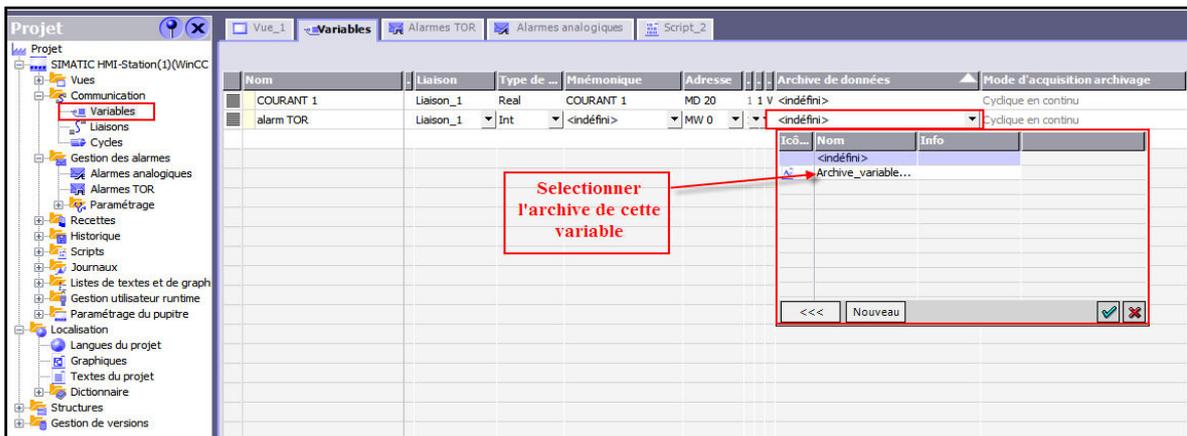


Figure III. 20. Editeur archives alarmes (2).

Pour l'affichage des alarmes en Runtime, on insert une "vue d'alarmes" comme celle des alarmes TOR dans la zone de travail, et on spécifie dans la fenêtre "propriété" >>"Générale">>"Affichage">>"archives alarmes" l'archive alarmes qu'on a créé auparavant.

III.3.10. Fonctions système en Runtime :

Pour les tâches de configuration courantes, WinCC flexible prévoit des fonctions système permettant d'exécuter un grand nombre de tâches en Runtime sans nécessairement avoir recours à la programmation.

Si on veut relier des éléments de commande à une fonctionnalité, on utilise les fonctions systèmes par exemple pour : Mettre un bit à 1 sur l'automate, modifier la valeur d'une variable, démarrer un archivage,...

Lorsqu'on configure une liste de fonctions, on sélectionne les fonctions systèmes dans une liste classée par catégorie, par exemple si on veut changer la valeur d'un bit d'une variable de notre projet dans le runtime on insert un "Bouton " dans la zone de travail .

Un double-clic sur le bouton nous permet d'afficher sa fenêtre de propriété, dans laquelle on définit son nom, sa couleurPour choisir la fonction désiré on clique sur "événements", une liste de fonctions classée par catégories'affiche.On choisit pour notre exemple " Edition de bit" >> "inverser bit", puis la variable à changer.

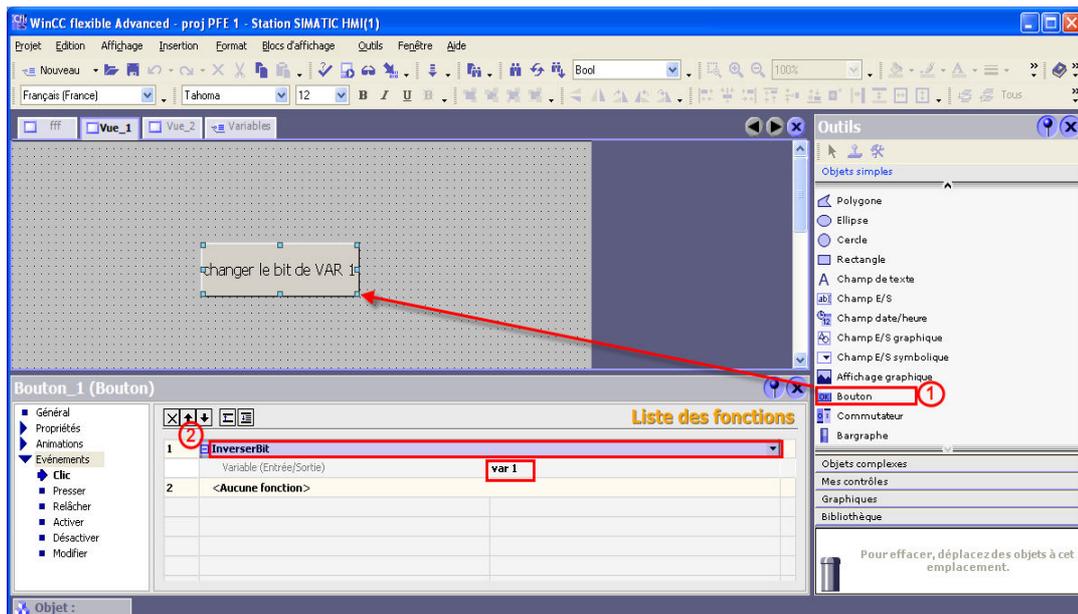


Figure III. 21. Configuration d'une fonction système.

On peut utiliser d'autres fonctions pour simplifier la manipulation dans le pupitre en runtime comme :

- Activer Vue Précédente
- Lancer Archivage
- Stopper Runtime
- Lire Enregistrement Sur API

III .4 . Simulation du projet sur WinCC flexible Runtime :

Le logiciel de visualisation de processus WinCC flexible Runtime, permet de faire fonctionner notre configuration sous Windows et de visualiser le processus. Il est également exécuté sur l'ordinateur de configuration pour tester et simuler le fichier projet compilé.

Pour simuler le projet terminé, plusieurs possibilités s'offrent à nous :

- **Simulation avec raccordement à l'automate :** Nous pouvons simuler notre projet en l'exécutant directement en runtime. Dans ce cas, les variables et les pointeurs de

zone ne peuvent fonctionner que si l'ordinateur de configuration est raccordé à l'automate concerné. Toutefois ce dernier peut être lui-même remplacé par le simulateur PLCSIM en mode intégré de STEP 7.

- On lance la simulation par clic sur l'icône située dans la barre d'outils :



- **Simulation sans raccordement à l'automate :** A l'aide du programme de simulation installé en même temps que WinCC flexible Runtime, nous pouvons simuler le projet, y compris les variables et les pointeurs de zone sans connexion à un automate. nous saisissons les paramètres des pointeurs de zone et des variables dans un tableau de simulation qui est lu par WinCC flexible Runtime pendant la simulation.

Cette simulation est lancée par un clic sur l'icône situé dans la barre d'outils :



Le tableau de forçage des variables s'affiche comme suite :

Variable	Type donnée	Valeur act.	Format	Cycle écriture	Simulation	Forcer valeur	ValMin	ValMax	Période	Lancer
Convey...	INT	26214	Déc	1,0	Incrément		-32768	32767	10,000	<input checked="" type="checkbox"/>
System_...	INT	13106	Déc	1,0	Incrément		-32768	32767	10,000	<input checked="" type="checkbox"/>
*---										<input type="checkbox"/>

Figure III. 22. Simulation sans raccordement à l'automate

La simulation peut être directement lancée en mode plein écran, cela est configuré à partir de la fenêtre du projet de WinCC flexible. Sous la rubrique "Paramétrage du pupitre" (Figure III.23) on coche la case "Mode plein écran". On peut même modifier dans cette rubrique la résolution ainsi que d'autres paramètres de l'affichage en Runtime.

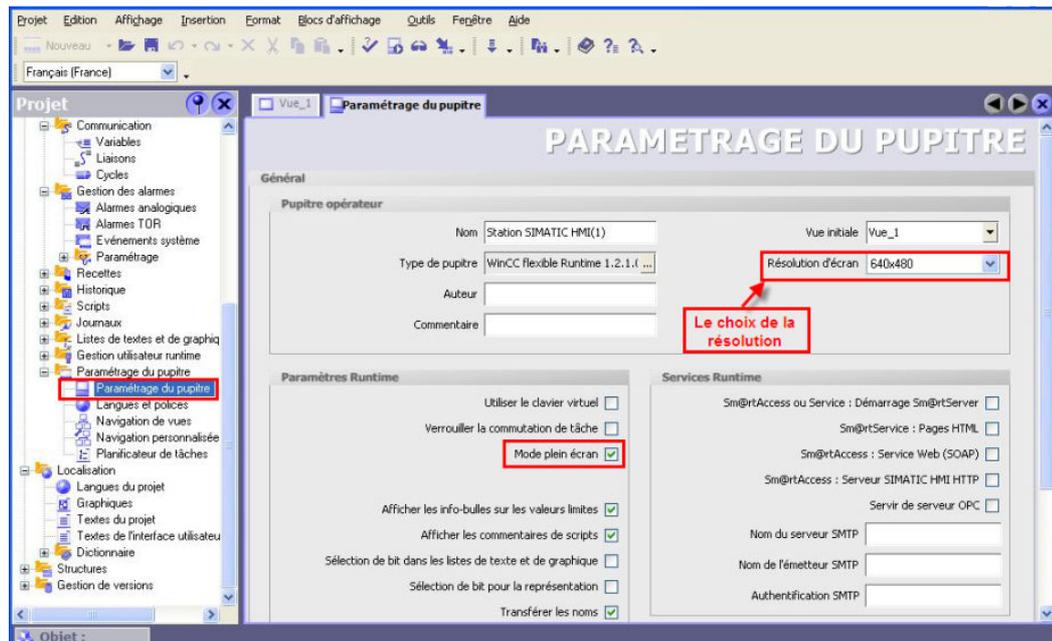


Figure III. 23. Paramétrage du pupitre.

Lorsque le Runtime est exécuté en mode Plein écran l'accès à ça fermeture doit être configuré à part. On crée un bouton prévu à cet effet pour quitter le Runtime.

III .5. Conclusion :

D'après ce qu'on a pu aborder dans ce chapitre concernant les fonctionnalités WINCC flexible, la liberté de que nous donne cet éditeur pour la configuration de fonctions, ainsi que pour la création de l'interface graphique HMI nous rapproche le plus possible de notre but, qui est la transparence entre l'opérateur et le processus.

Donc on pourra dire que WINCC flexible nous permet d'optimiser la supervision industrielle aux dimensions nécessaire à chaque cahier de charges.

Malgré tout ce qu'on a pu décrire des éditeurs du WINCC flexible, cela restera assez loin des horizons de ce que pourra être développé grâce à ce dernier.

L'étendu du développement de ces fonctionnalités tiendra de la personnalisation et de l'édition de fonctions propre à chaque processus industriel.

CHAPITRE IV :

Application

IV.1. Introduction :

Dans ce chapitre nous allons détailler notre application, dont le programme ainsi que l'automate utilisés ont été décrits dans le chapitre I.

Pour cela nous avons commencé par décrire l'installation du siège OrascomTélécom Algérie (OTA) où s'est déroulé notre stage de projet de fin d'étude. Nous nous sommes basés dans cette description aux équipements visés par notre projet sans s'attarder ni développer le reste de l'installation où l'application n'a pas de rôle direct à jouer.

Après cette brève description, nous allons voir le développement de notre application en terme propre, et comment nous avons utilisé les connaissances acquises le long de notre cursus et surtout dans ce projet de fin d'étude. Afin d'arriver à répondre au mieux aux exigences du cahier de charges de l'entreprise. Par la suite nous avons décrit quelques options, lesquelles nous avons jugé aptes à améliorer cette application de manière à la rendre plus efficace pour ce qui du suivie à court et à long terme de la station.

IV.2. Description de la station télécommunication mobile (Switch) OTA :

Le Switch OTA se compose de plusieurs équipements appelés NE (network equipments), ceux-ci permettent de gérer le réseau de télécommunication et faire la connexion entre les différents abonnés. Ce Switch est alimenté via un système d'énergie électrique sécurisé et redondant ; qui est illustré par le schéma synoptique ci-dessous [13] :

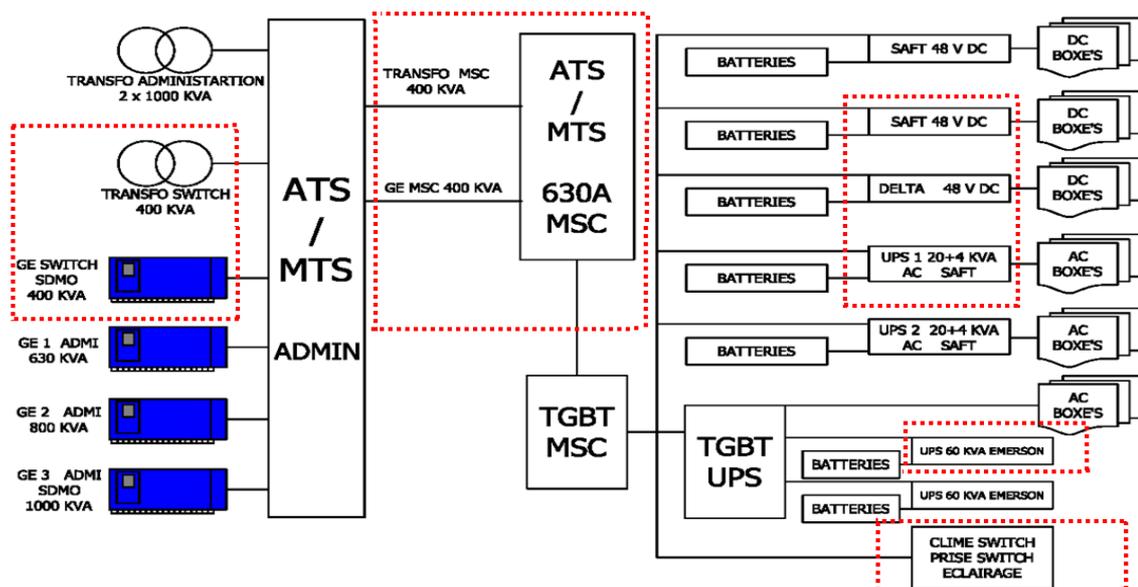


Figure IV. 1. Schéma synoptique de l'alimentation du siège OTA [13]

Les équipements que nous allons superviser sont entourés par des pointillés.

IV.2. 1. Les sources d'alimentation du Switch :

Le **SWITCH** est alimentée par une source principale qui est le réseau SONELGAZ, toutefois si celui-ci est défaillant, le groupe électrogène prend le relais pour assurer la continuité d'alimentation. La commutation est assurée par un inverseur de source connu sous le nom ATS/MTS.

IV.2. 1. 1. Le réseau principal (SONELGAZ) :

Le réseau principal est relié à celui de SONELGAZ via un poste de transformation d'énergie électrique MT/BT (Figure IV.2), il est constitué d'un transformateur de 400 KVA de puissance qui a une tension d'entrée de 10 KV et une tension de sortie de 400 V. Et des trois cellules[13] :

- Départ
- Arrivé
- Protection du transformateur



Figure IV. 2. Poste transformateur.

IV.2. 1. 2. Le Réseau Secours (Le Groupe Electrogène) :

L'alimentation de secours est assuré par un Groupe Electrogène de 400 KVA de puissance avec une tension de sortie de 400 V, constitué de :

- Un moteur de marque VOLVO TAD 1242 GE ;
- Une Génératrice LEROY SOMMER ;
- Une carte de gestion et de commande ;
- Un capotage d'isolation phonique (Insonorisé) ;
- Un kit de remplissage de gasoil automatique ;
- Un kit de préchauffage du bloc moteur ;
- Un réservoir journalier interne et une citerne externe enterrée ;

- Un chargeur de batterie intégré.



Figure IV. 3. Le groupe électrogène.

IV.2. 2. Les équipements de la salle d'énergie :

IV.2. 2.1. L'inverseur ATS/MTS :

L'ATS/MTS (Automatique Transfer Switch / Manuel Transfer Switch) c'est une armoire d'inversion de source automatique et/ou manuel entre le réseau principale et celui de secours.

La commutation automatique est activée lorsque les relais de phase et de temporisation détectent une défaillance d'une des deux sources.

Les conditions de défaillance pour le réseau principale sont :

- Min / max tension (chute de tension et surtension) ;
- Inversion de phase ;
- Manque phase ;
- Coupure du neutre ;
- Coupure du réseau.

Les conditions de défaillance du groupe électrogène sont :

- Manque phase ;
- Inversion de phase ;
- Min /Max tension ;
- Fréquence

La commutation manuelle est activée lorsque la commutation automatique est défaillante on faisant une action sur les interrupteurs et les commutateurs manuels.[13]

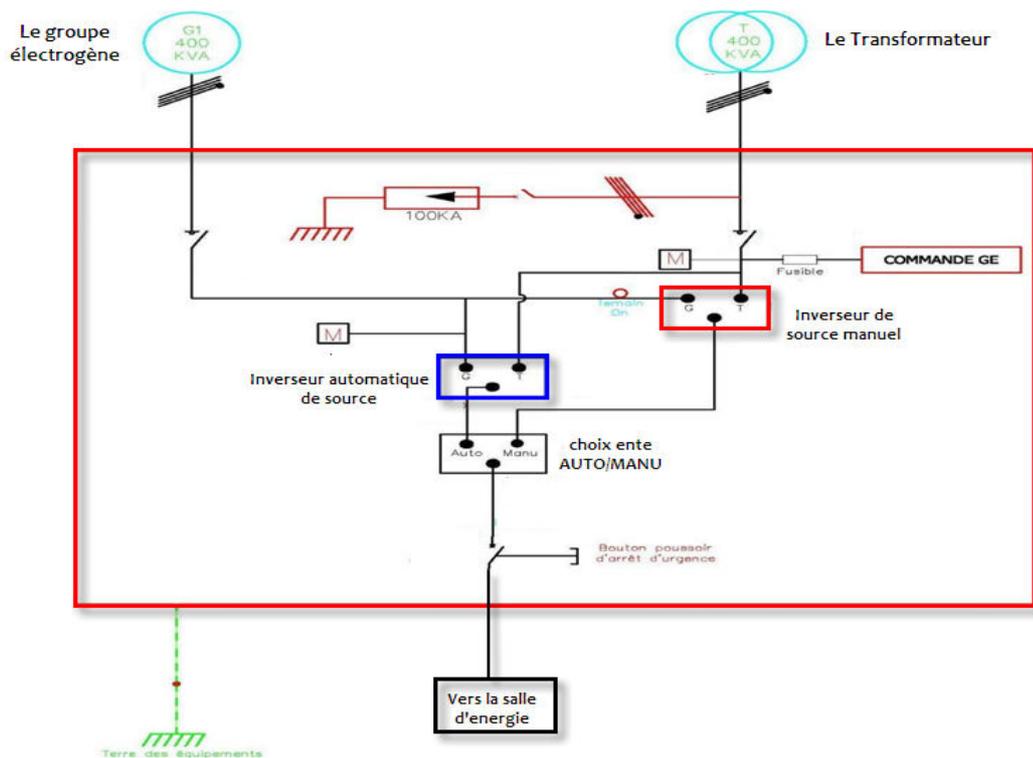


Figure IV. 4. L'inverseur de source ATS/MTS.[13]

IV.2. 2.2 . Le tableu générale basse tension (TGBT) :

Le TGBT est un tableau de distribution électrique qui assure l'alimentation de tous les équipements de la station (redresseurs, onduleurs, climatisation, etc.), il est alimenté par l'ATS/MTS.

IV.2. 2.3 . Les redresseurs :

Les redresseurs sont de type digital, ils sont muni d'un module qui s'occupe de leur gestion, toutes les informations sont affichées à travers un écran intégré à l'armoire des redresseurs. Il y a deux redresseurs SAFT1 et DELTA.

Les redresseurs utilisés sont alimentés par une source alternative 400 V, une fois redressé on obtient une tension continue de -48 V, cette énergie est utilisée pour l'alimentation des équipements NE.

En cas de coupure du réseau alternative, des batteries de secours sont prévues afin d'assurer l'alimentation permanente des équipements. Celles-ci ne sont là que pour alimenter pendant le temps de la commutation de l'ATS/MTS (Figure IV.5).

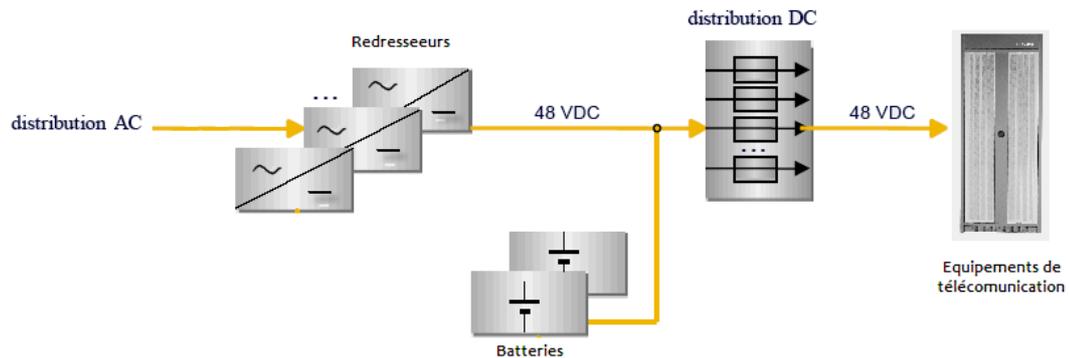


Figure IV. 5. Schéma installation des redresseurs.[13]

IV.2. 2.4 .Les alimentation sans interruption UPS (onduleur) :

Les alimentations alternatives sans interruption sont appelées couramment dans l'industrie "ONDULEUR" ou UPS (Uninterruptible Power System) son rôle est d'assurer une alimentation stable en tension et en fréquence pour la charge, ainsi qu'une continuité de disponibilité de tension avec des batteries lors d'une coupure de la source d'alimentation.

La station OTA dispose de deux types d'UPS :

- EMERSON 60 KVA (3phases).
- SAFT 24 KVA monophasé 220V.

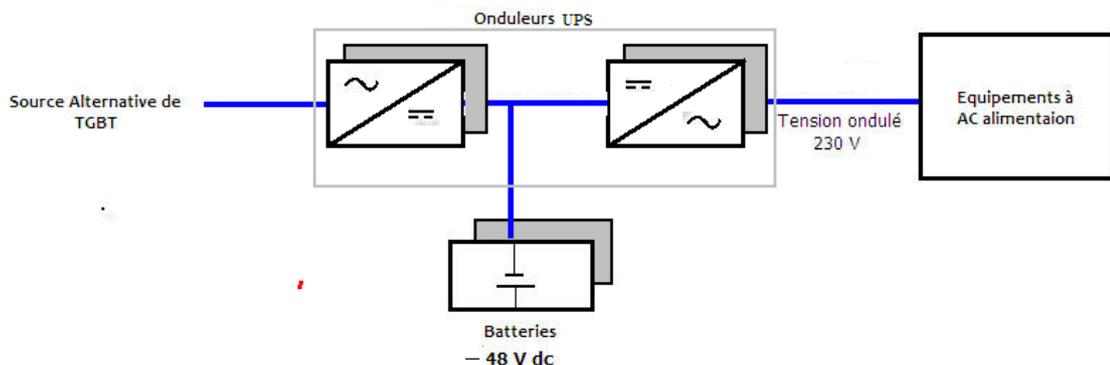


Figure IV. 6. Schéma Installation des onduleurs.[13]

IV.2. 2.5 . La climatisation :

Pour Contrôler la température de la salle d'énergie, deux types de climatisation sont utilisées:

- De précision : elle est assurée par de grands climatiseurs à grande puissance commandés entre eux par un système maître/esclave. Elle est destinée à la protection des équipements de la salle contre la surchauffe.

- De confort : destiné essentiellement au personnel mais aussi dans certain cas comme aide à celle de précision. Il est destiné à la régulation de température.

IV.2.3 . Les équipements de télécommunication :

Les équipements de télécommunication NE de OTA se trouvent au niveau de la salle Switch, ils sont alimentés par des sources alternatives, via les onduleurs SAFT et Emerson, et des sources continues, via les redresseurs SAFT et DELTA. Leur rôle est de gérer :

- La transmission téléphonique.
- Les serveurs des plateformes : service, sécurité, abonnés...etc.
- La commutation des appels.

IV.2.4 .La supervision actuelle de la station OTA :

Lorsqu'un défaut apparaît sur un équipement, ce dernier génère une alarme, sa notification s'affiche sur une station locale (ordinateur) au niveau du SWITCH. La station génère un mail qui est transmis directement à l'ingénieur. Selon le cas cela prend un temps assez long.

Les alarmes sont affichées aussi sur la station locale sous forme de texte, qui contient : la date et l'heure d'alarme, un numéro qui spécifie l'équipement et le texte d'alarme. Par exemple pour une alarme défaut groupe électrogène le texte s'affiche comme suite :

```
!! *A0628/093 /14-03-23/14 H 44/N=6824/TYP=ENP/CAT=ID/EVENT=DAL
/NCEN=SSP04 /OBJET=CDC/AGED=CDC
/TEXAL=DEFAULT GE

!! *A0628/093 /14-03-23/14 H 44/N=6824/TYP=ENP/CAT=ID/EVENT=FAL
/NCEN=SSP04 /OBJET=CDC/AGED=CDC
/TEXAL=DEFAULT GE
```

Figure IV. 7. Texte d'alarme de la supervision actuelle.[13]

Il y a quelques équipements supervisés localement, c'est à dire au niveau de l'équipement lui-même, dans un écran intégré à leur armoire, ce qui présente une difficulté d'intervention en cas d'urgence.

Pour améliorer cette supervision l'équipe des ingénieurs a proposé une nouvelle méthode de gestion des alarmes et des commandes, à base d'automate programmable, afin de pouvoir superviser les équipements de toute la station complète à travers une seule

interface graphique au niveau de la salle de commande. Cette interface contient toutes les informations (alarmes, défaut, seuil, visualisation..), ainsi que les commandes nécessaires pour gérer la station.

IV.3. Développement du projet de gestion et supervision de l'alimentation du siège OTA :

Cette étape consiste à créer et développer un programme de supervision et de commande de la station OTA. Celui-ci sera implémenté dans un automate pour gérer notre installation.

La figure ci-dessous décrit l'organigramme du développement de notre projet.

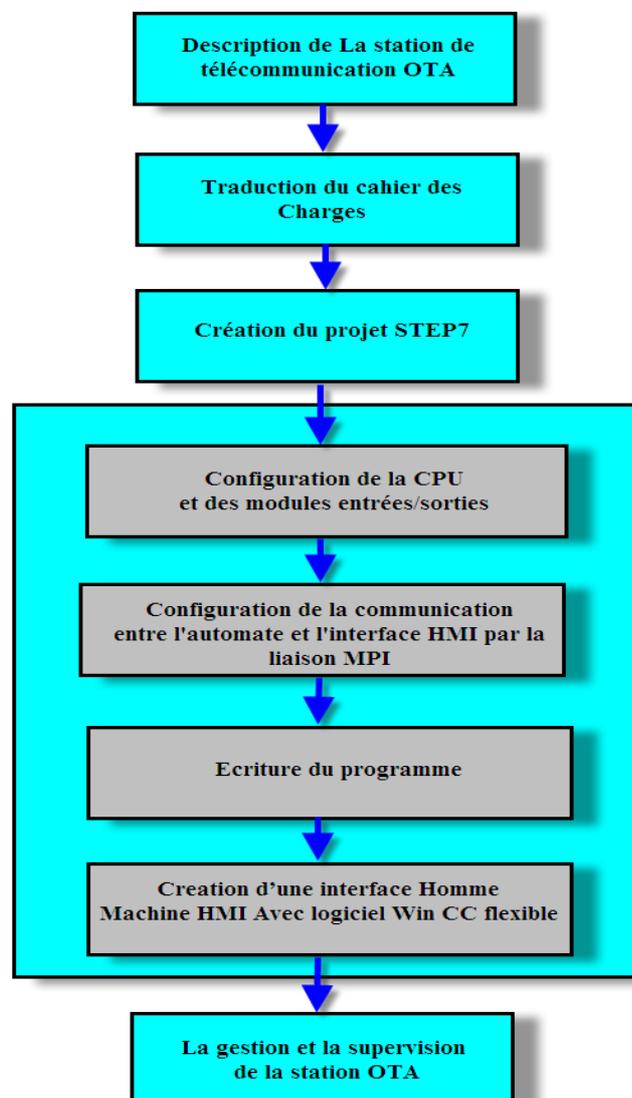


Figure IV. 8.L'organigramme du développement du projet.

IV.3. 1. Définition du cahier des charges :

Le cahier de charge a été défini par les ingénieurs d'OTA. Le travail demandé consiste à développer une Interface Homme Machine afin de commander, configurer l'installation et afficher les défauts et les alarmes. Tout en utilisant un automate programmable pour cette gestion.

On peut diviser ce cahier de charge en deux parties. La première est la supervision qui regroupe la gestion des alarmes, des informations et la visualisation. Et la deuxième qui est la commande.

IV.3. 1.1 .La partie supervision : le cahier de charges définit plusieurs paramètres importants qui doivent être suivis en permanence. L'opérateur doit être alerté dans le cas de changement critique de leurs états.

a) Gestion des alarmes et d'informations :

- Les alarmes des équipements
 - Défaut générale de groupe électrogène (GE) ;
 - Niveau gasoil bas ;
 - Défaut générale climatisation ;
 - Défaut générale redresseur SAFT ;
 - Défaut générale redresseur DELTA ;
 - Défaut générale onduleur SAFT ;
 - Défaut générale onduleur EMERSON ;
 - Défaut du disjoncteur poste de transformation ;
 - Défaut cellule fusible de protection poste de transformation.

➤ Etat d'alimentation du réseau :

L'opérateur ou toute personne habilitée doit savoir selon quel réseau est alimentée la salle d'énergie :

- Alimentation sur réseau SONELGAZ ;
- Alimentation sur groupe électrogène.

➤ **Alarmes et avertissements de dépassement de seuil :**

- Alarme température basse de la salle d'énergie(seuil inférieure), déclenche si la température est inférieure à 20 °C.
- Avertissement température haute de la salle d'énergie (seuil supérieure 1),déclenche si la température est supérieure à 30 °C.
- Alarme température très haute de la salle d'énergie (seuil supérieure 2), déclenche si la température est supérieure à 35 °C.
- Trois Avertissements : (les courants d'alimentation salle d'énergie)
 - le courant de phase 1 dépasse le seuil de 590 A ;
 - le courant de phase 2 dépasse le seuil de 590 A ;
 - le courant de phase 3 dépasse le seuil de 590 A.
- Trois alarmes : (les courants d'alimentation salle d'énergie)
 - le courant de phase 1 dépasse le seuil de 630 A ;
 - le courant de phase 2 dépasse le seuil de 630 A ;
 - le courant de phase 3 dépasse le seuil de 630 A.

a) **La visualisation :**

- Visualisation en temps réel des courants d'alimentation de la salle d'énergie de chaque phase.
- Visualisation en temps réel de la température de la salle d'énergie .

IV.3. 1. 2. La commande :

- La commande Test de démarrage du groupe électrogène, est une commande à condition.

On ne peut pas exécuter la commande si un des deux défauts suivants est présent :

- Défaut générale du groupe électrogène ;
- Niveau gasoil bas.

IV.3. 2. Définition des entrées/sorties :

D'après le cahier de charges proposé, on peut définir le type et le nombre des entrées/sorties nécessaires à la récupération du maximum d'informations pour répondre aux exigences du cahier des charges. Donc on a dimensionné :

- **Onze (11)** entrées digitales (TOR) pour les alarmes des équipements et les informations d'état du réseau.
- **Quatre (04)** entrées analogiques pour la température et les **trois (03)** phases du courant.
- **Une (01)** sortie digitale pour la commande démarrage du GE.

IV.3. 3. Le choix d'automate :

Cette étape vise à aboutir à un meilleur choix d'automate pour réaliser notre application.

Notre choix c'est fait selon les critères suivants :

- Nombre d'entrées et de sortie doit être suffisant pour notre projet (11 entrées TOR, 4 entrées analogiques, 1 sortie TOR).
- La vitesse de traitement donc une mémoire assez importante ainsi que les fonctions du processeur.
- L'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication (Profibus , MPI), donc un automate à 2 DP c'est à dire on peut le configurer comme maître ou esclave pour un réseau Profibus (ce critère est exigé par les ingénieurs d'OTA) ,
- La disponibilité du matériel (API) sur le marché, l'existence de la documentation et le savoir-faire du personnel de l'entreprise OTA sur le matériel.

Notre choix c'est porté sur l'automate **SIMATIC S7- 314C-2 DP** (Chapitre I). Ces caractéristiques répondent au mieux aux conditions du cahier des charges, de plus il présente un cout relativement faible (très bon rapport qualité prix).

IV.3. 4. Création du projet STEP 7 :

Cette étape consiste à la création du projet STEP 7, la configuration des matériels, ainsi que l'écriture du programme.

IV.3. 4.1 Configuration du matériel dans STEP7 :

Le projet a été créé selon la procédure vue dans le chapitre II. Après l'insertion de la station S7-300 nous configurons le matériel utilisés :

- Choissant la **CPU « 314 C 2-DP »**.
- Module d'alimentation : **PS 405 10 A**.
- Module communication : **MPI**.
- Module entrées\sorties : **intégrés au CPU 314 C 2-DP**.

Remarque : Pour le paramétrage des modules d'entrées/sorties analogiques, on a choisi le signal de mesure fournit par le capteur, qui est sous forme de tension (U), celle-ci varie entre 0 et 10 V pour les 4 entrées analogiques. L'unité de température définie est le degré Celsius.

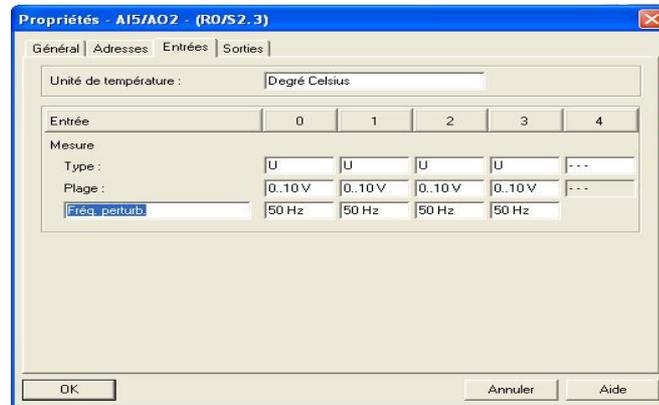


Figure IV. 9. Paramétrage E/S Analogique.

IV.3. 4. 2 Ecriture du programme :

Avant de pouvoir commencer l'écriture du programme en sens propre, nous devons définir au préalable tous les mnémoniques (Annexe C.1).

Dans la partie programmation on s'est basé sur le langage de programmation graphique CONT. Ce dernier permet de suivre le trajet du courant entre les barres d'alimentation. De plus notre programme ne nécessite pas un langage de programmation trop développé.

IV.3. 4. 2.1. Traitement du programme pour les entrées TOR :

Cette partie de programmation a pour rôle de gérer les alarmes des équipements et les informations qui concernent l'état du réseau. Cela se fait par la récupération de l'état du contact sec (0 ou 1), et l'affectation de l'entrée lue dans une mémoire. Par exemple, le traitement de l'alarme "défaut général onduleur SAFT 1" s'effectue comme suit.

Réseau 21: ONDULEUR SAFT 1

récupération d'alarme onduleur SAFT 1

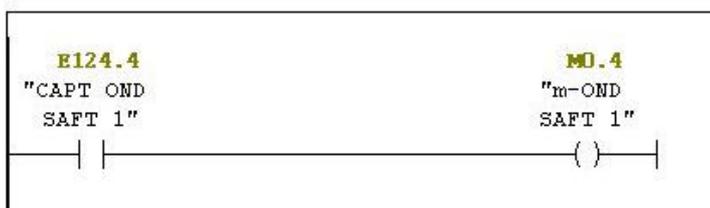


Figure IV. 10. Programme pour les entrées TOR.

Toutes les entrées digitales utilisées pour les alarmes et les informations sont traitées et mémorisées de cette même manière (Annexe C2).

IV.3. 4. 2.2. Traitement du programme pour les entrées Analogique :

Les entrées analogiques sont utilisées pour la récupération des mesures des capteurs de courants et de température, pour avoir la valeur réelle des mesures on a utilisé la fonction (SCALE FC105)de la mise à l'échelle,puis on mémorise la valeur réelle dans une mémoire double mot "MD" (figure IV.11).

Afin d'expliquer le principe de la mise à l'échelle, on a pris comme exemplele courant (phase 1) :

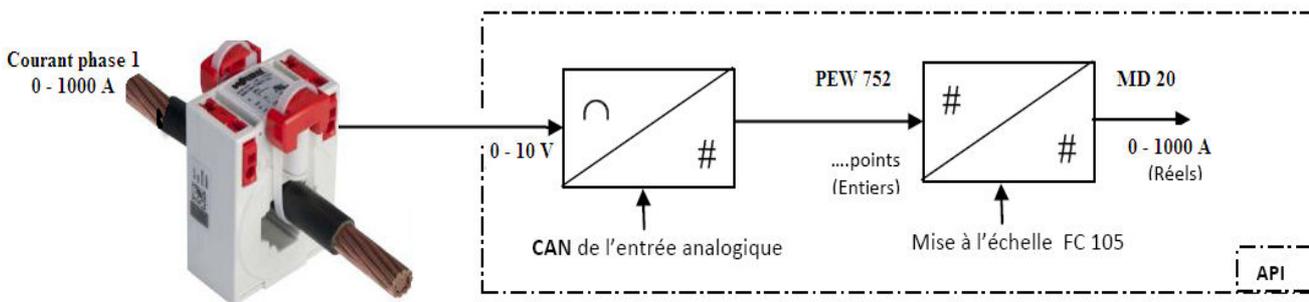


Figure IV. 11. Traitementdes entrées analogiques dans l'automate.

Le programme STEP 7 y correspondant est :

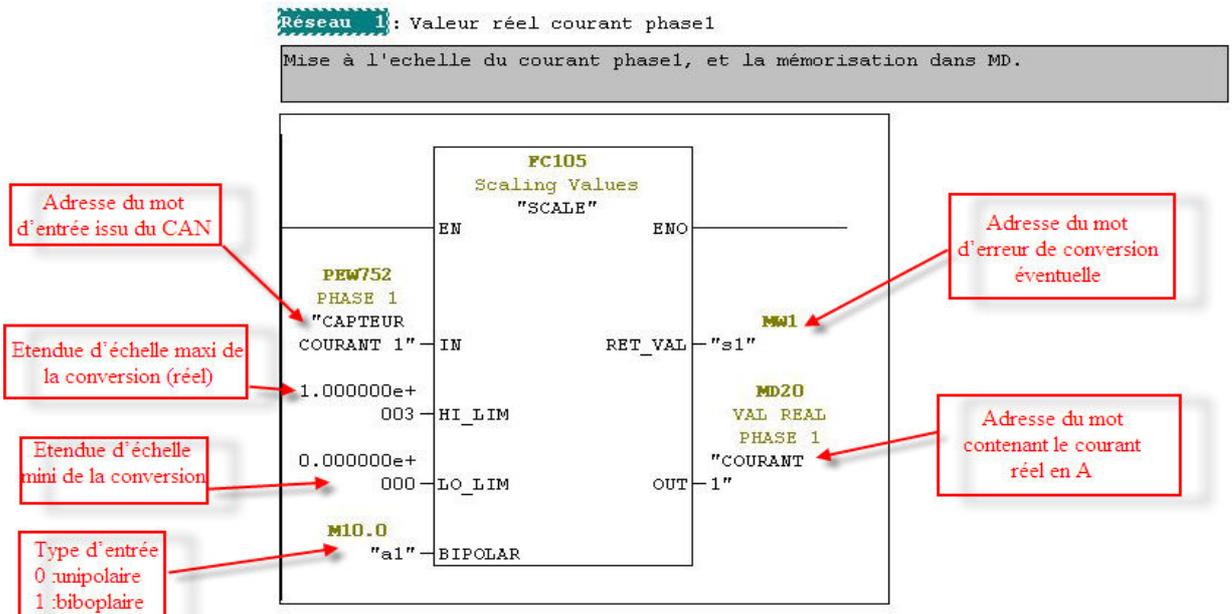


Figure IV. 12. La fonction SCALE.

Avec cette fonction (SCALE) on a traité de la même façon les entrées analogiques (Annexe C2).

IV.3. 4. 2. 3. Programmation d'avertissements pour le dépassement

deseuil :

Cette partie est nécessaire pour gérer les alarmes reliées à des variables analogiques, comme dans notre cas les courants et la température. On prend comme exemple, les avertissements de dépassement des seuils de température (Salle d'énergie).

- L'alarme basse température, se déclenche si la température est inférieure à 20°C ;
- L'avertissement haute température, se déclenche si la température est supérieure à 30°C ;
- L'alarme très haute température, doit se déclencher si la température dépasse 35°C.

Après récupération de la valeur réelle de la température par la fonction mise à l'échelle, on insère un opérateur de comparaison des nombres réels. Si la comparaison est vraie, donc le résultat logique "1" sera mémorisé dans une mémoire, suivant laquelle l'alarme se déclenche (figure IV.13).

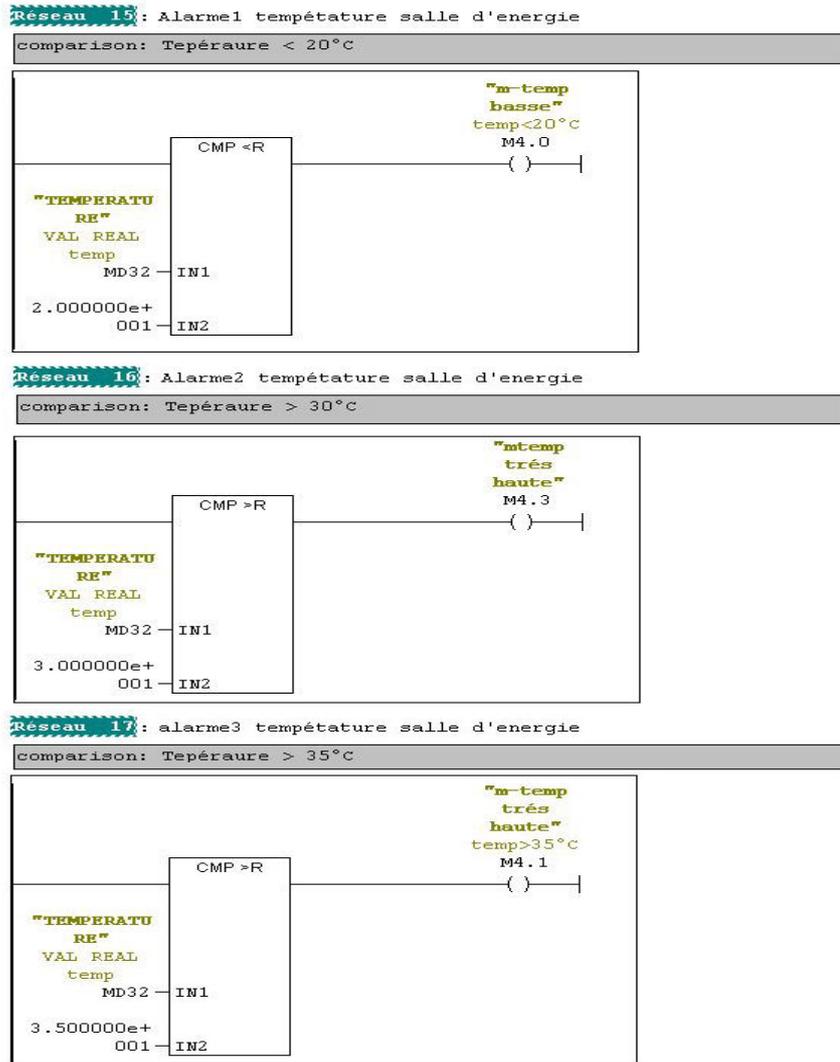


Figure IV. 13. Programme pour alarmes du dépassement de seuil.

Les alarmes des seuils de courants sont traitées de la même manière (Annexe C2).

IV.3. 4. 2. 4. Traitement du programme pour la sortie TOR :

Les sorties TOR sont utilisées pour commander des actions de type tout-ou-rien. Nous avons donc utilisé une sortie pour commander le démarrage du groupe électrogène, à partir d'un bouton de l'interface HMI. Cette commande est établie en forçant une mémoire "M 100.0" en runtime à la valeur 1 (de WinCC), suivant les conditions imposées dans le cahier des charges (figure IV.14).

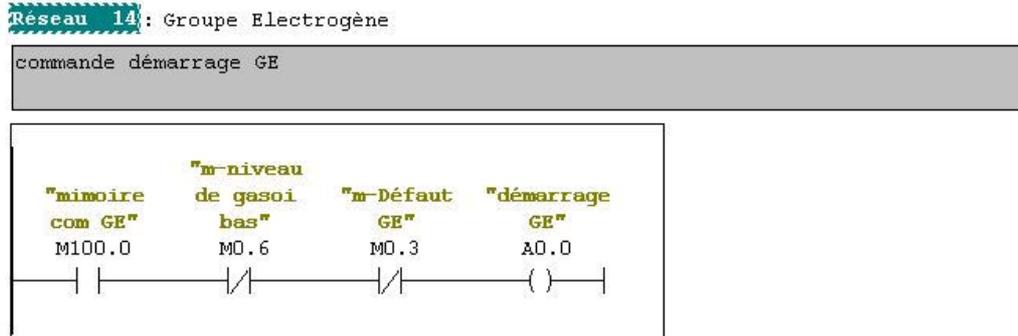


Figure IV. 14. Programme du test démarrage GE.

IV.3. 5. Configuration de la communication entre le pupitre et l'automate S7 314C-2DP :

Après l'intégration du projet WINCC flexible au STEP7 (chapitre III), la communication entre le pupitre opérateur et l'automate S7 314C-2DP est assurée par le réseau MPI configuré (figure IV.15).

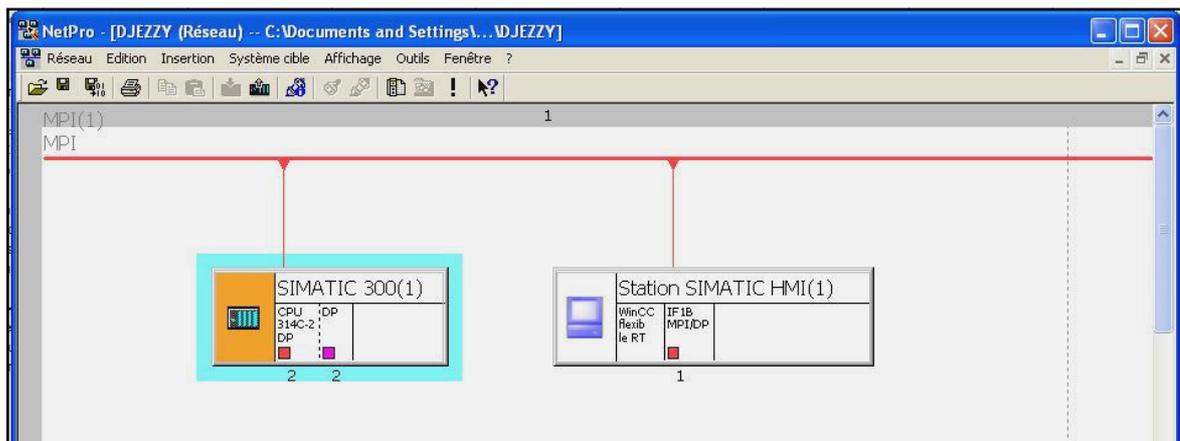


Figure IV. 15. La liaison entre l'automate et le pupitre.

Il faut aussi configurer le paramétrage PG/PC de l'adaptateur qui lie l'automate au pupitre. Nous accédons à la configuration dans l'onglet "Outils" : "Paramétrage de l'interface PG/PC ". Le jeu de paramètres choisis est "PC Adapter (MPI)" et l'entrée de l'application est "WinCC (MPI) – PC Adapter" ce qui nous permet de superviser la station en mode runtime dans le WinCC (figure IV. 16).

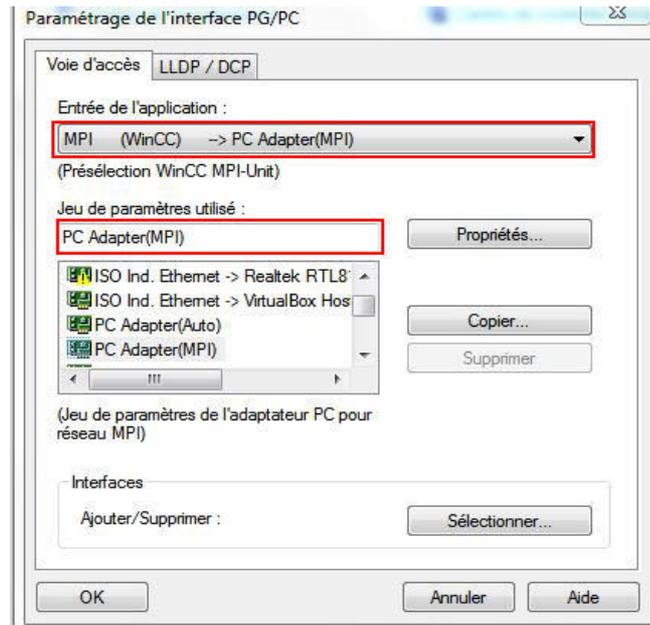


Figure IV. 16. Paramétrage de l'interface PG /PC.

IV.3. 6. Création de l'interface homme machine HMI du projet par WinCC flexible:

D'abord nous avons défini le type du pupitre sur lequel les informations seront transmises, pour notre application nous avons utilisé un pupitre **PC** (poste opérateur), lié à l'automate par une liaison MPI.

IV.3. 6.1 Elaboration de la supervision

La supervision de la station de télécommunication OTA consiste à commander et à contrôler plusieurs paramètres. Commander comme le test démarrage et l'arrêt du groupe électrogène. Le contrôle soit la visualisation en temps réel de la température et des courants d'alimentation de la salle d'énergie, ainsi que l'affichage instantané des alarmes et d'avertissements en cas d'anomalie au niveau des équipements de la salle d'énergie.

IV.3. 6.2 Description des vues :

Notre projet de supervision est structuré de manière à pouvoir visualiser l'ensemble de la station et atteindre les informations et commandes de chaque équipement dans sa vue propre ainsi que d'autres vues spécifiques, donc en tout on a quatorze vues:

- **Vue "principale"** : Cette première figure est considérée comme la page d'accueil de notre projet puisqu'elle englobe l'ensemble de l'installation OTA, elle permet la navigation entre les différentes vues développées, qui sont propres à chaque

équipement. Ceux-ci sont représentés en forme de boutons. Ces boutons permettent d'accéder à la vue et servent aussi d'indicateurs (clignote en rouge) dans le cas où au moins un défaut est présent au niveau de l'équipement en question.

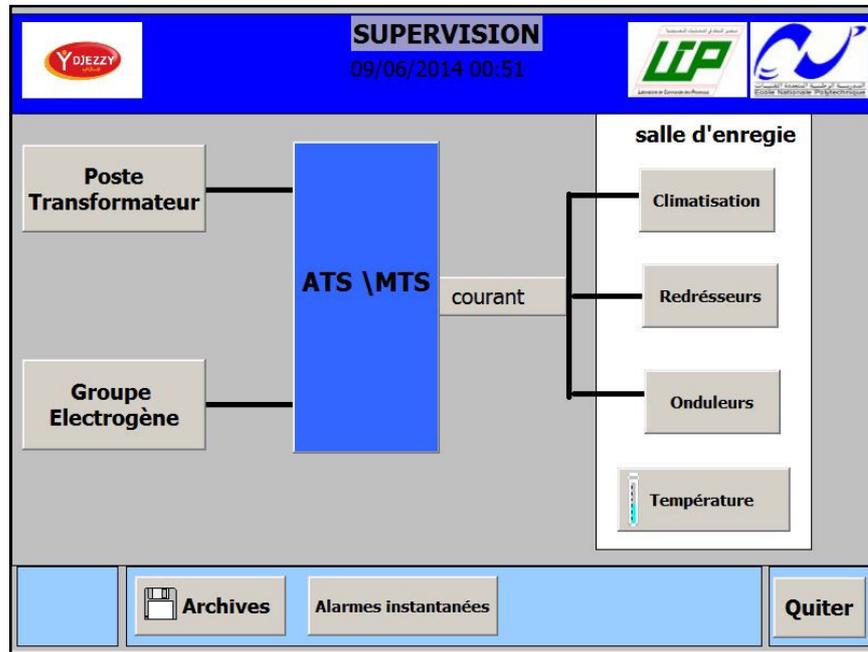


Figure IV. 17. Vue principale de l'interface graphique.

- **Vue "poste transformateur"** : on accède à cette vue à partir du bouton nommé "POSTE TRANSFORMATEUR", elle affiche les défauts du transformateur (disjoncteur et fusible). Si le défaut est présent son voyant clignote en rouge.

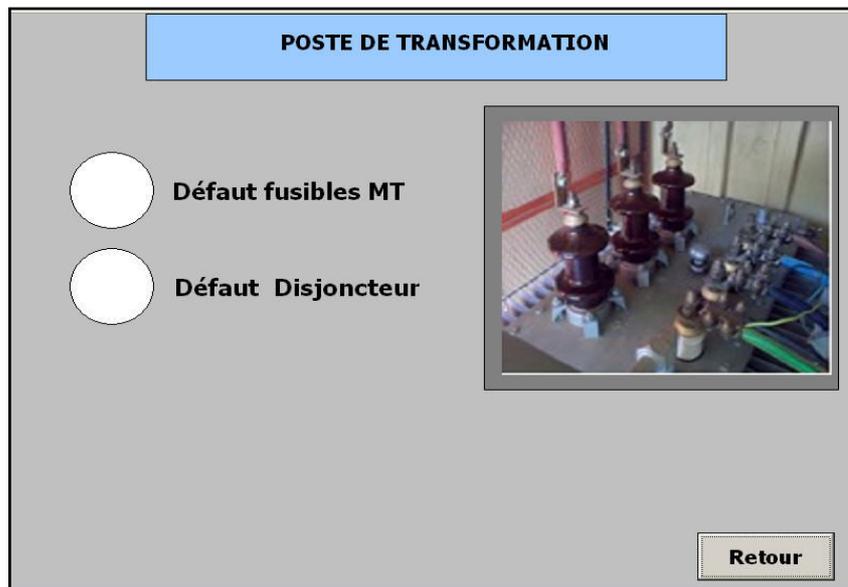


Figure IV. 18. Vue poste transformateur

- **Vue "Groupe électrogène"** : représenté par le bouton " Groupe électrogène ", elle affiche le défaut général du groupe electrogène et le niveau du gasoil s'il est bas. Comme pour la vue précédente des voyant rouge clignotant signale les evenements surveillés.



Figure IV. 19. Vue groupe electrogène

- **Vue "ATS/MTS"**: cette vue représente l'inverseur d'alimentation, la source d'alimentation y est précisée sous forme de schéma électrique. Indiqué par un interrupteur fermé ou ouvert selon l'état d'alimentation.

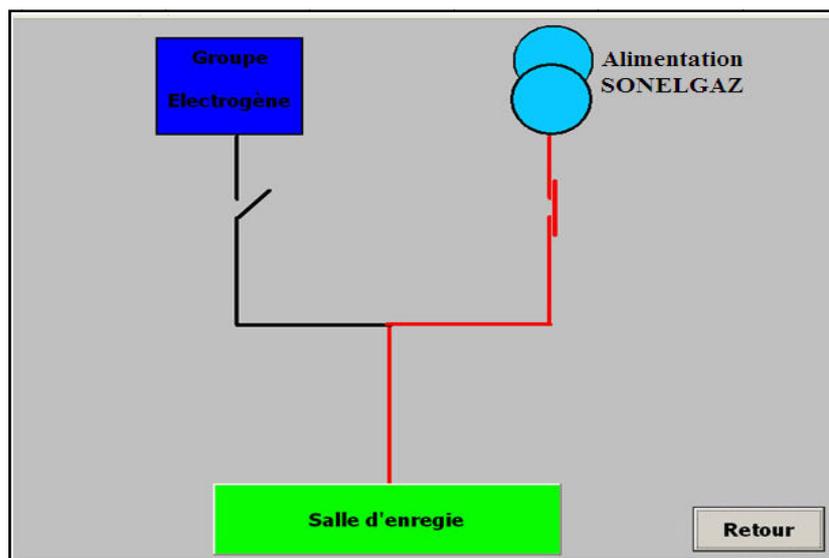


Figure IV. 20. Vue commutateur "ATS/MTS"

- **Vue "Redresseurs"** :cette vue affiche les alarmes défauts des deux redresseurs SAFT1 et DELTA.



Figure IV. 21. Vue redresseurs

- **Vue "Onduleurs"** :lorsqu'un défaut général provient sur l'un des onduleurs SAFT et EMERSON, un voyant rouge clignote.



Figure IV. 22. Vue onduleurs

- **Vue "Climatisation"** :celle-ci pour rôle d'afficher le défaut générale qui peut se produire au niveau du climatiseur de la salle d'énergie



Figure IV. 23. Vue climatisation

- **Vue "Courants"** : Cette vue sert à visualiser les courbes du courant des trois phases en temps réel, celle-ci représentées par des couleurs différentes, et donné la valeur instantanée deces courants en Ampere.

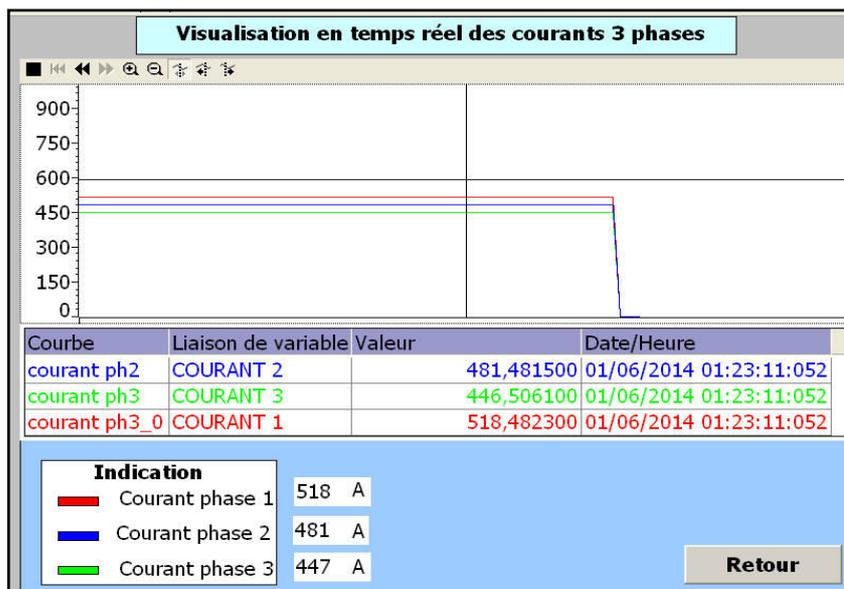


Figure IV. 24. Vue courbes courant

- **Vue "Température"** : A partir de cette vue on peut visualialiser la température de la salle d'énergie en temps réel sur une bargraphe. Elle affiche aussi les alarmes de dépassement des seuils de température (< 20°C , >30°C et >35°C).

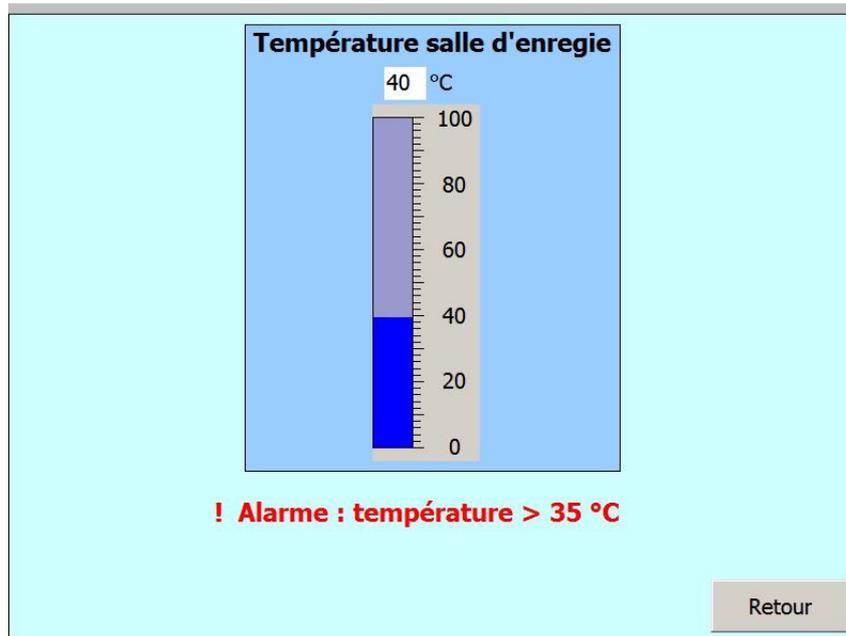


Figure IV. 25. Vue température

- **Vue " Alarmes instantanées"** : Tous les alarmes et avertissements instantanées de l'installation sont affichées dans cette vue .

Affichage instantané des alarmes					
N°	Heure	Date	Etat	Texte	GR
6	01:31:35	01/06/2014	A	Défaut Onduleurs EMERSON	0
3	01:31:30	01/06/2014	A	Défaut Redresseur DELTA	0
2	01:31:26	01/06/2014	A	défaut redresseur MTP1	0
1	01:31:26	01/06/2014	A	Défaut climatisation salle d'energieTA	0

Retour

Figure IV. 26. Vue alarmes instantanées

- **Vue "Archive" :**On peut accéder à partir de cette vue à tous les informations archivées de l'installation. Des boutons représentent les vue archives courbes courants, courbetempérature et l'archive des alarmes qui se sont déjà produites.

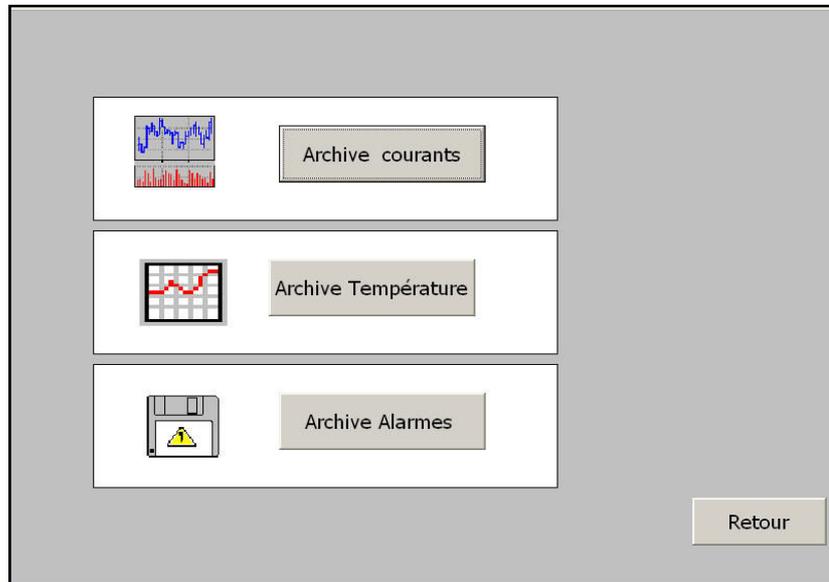


Figure IV. 27. Vue archives

- **Vue " Archive courants " :**Celle-ci affiche les courbes courants (trois phases) archivées pendant un temps choisit, ce dernier est définis lors de la configuration de l'archive dans WinCC flexible.

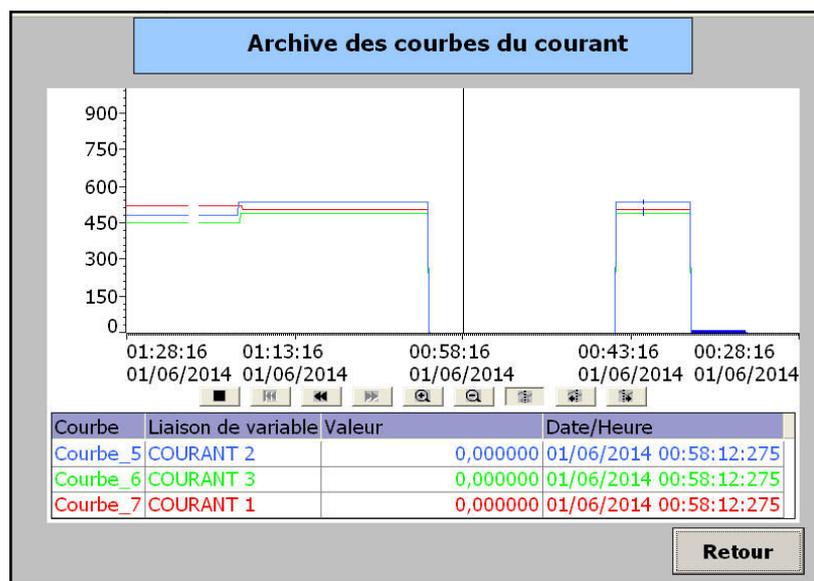


Figure IV. 28. Vue archive courants

- **Vue "Archive température" :** cette vue affiche la courbe de la variation de la température dans la salle d'énergie pendant un temps choisit.

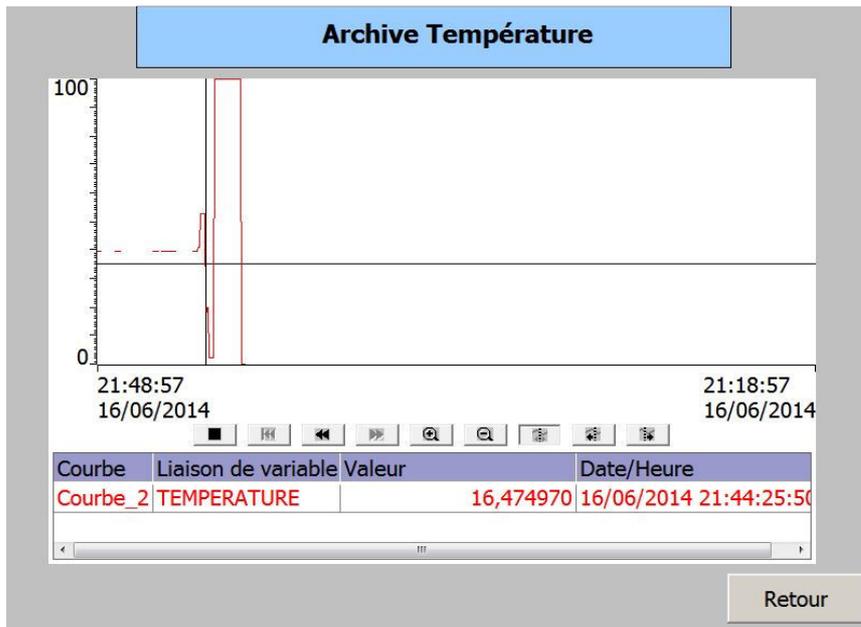


Figure IV. 29. Vue archive température

- **Vue " Archive alarmes" :** toutes les alarmes dans la station sont enregistré dans un tableau de la vue archive alarmes .celle-ci affiche la date et l’heure de l’apparition et de la disparition du défaut.

N°	Heure	Date	Etat	Texte	GR
E...9	23:16:32	16/06/2014	A	temp>35	0
A...8	23:16:32	16/06/2014	A	temp>30	0
E...9	21:48:47	16/06/2014	A	temp>35	0
A...8	21:48:47	16/06/2014	A	temp>30	0
E...9	21:47:59	16/06/2014	A	temp>35	0
A...8	21:47:59	16/06/2014	A	temp>30	0
E...9	21:46:24	16/06/2014	A	temp>35	0
A...8	21:46:24	16/06/2014	A	temp>30	0
A...8	21:45:41	16/06/2014	A	temp>30	0
E...9	21:45:41	16/06/2014	A	temp>35	0
E...9	21:44:41	16/06/2014	A	temp>35	0
E...9	21:44:38	16/06/2014	AD	temp>35	0
A...8	21:44:27	16/06/2014	A	temp>30	0
E...9	21:44:27	16/06/2014	A	temp>35	0
A...7	21:44:27	16/06/2014	AD	temp<20	0
A...7	21:44:19	16/06/2014	A	temp<20	0

Figure IV. 30. Vue archive alarmes.

IV.3. 6.3 Amélioration optionnel :

Après la création des vues de notre projet HMI, et la gestion des différents objets ce qui répond aux exigences du cahier de charges, nous en avons développé d'autres options qui serviront à améliorer la supervision, simplifier la manipulation, permettre un meilleur suivi de la station et aussi restreindre l'accessibilité aux commandes. Parmi les options :

- L'archivage des données : est utilisé pour mieux diagnostiquer les défaillances récurrente, optimiser les cycles de maintenance
- L'impression des archives : cette option permet l'archivage sous format papiers.
- Crée des sessions propres aux utilisateurs prioritaires, protégé par des mots de passe afin de restreindre l'accès à la commande à un groupe définie d'opérateur.



Figure IV. 31. Protection de la commande par mot de passe

- Personnalisation de l'interface par l'insertion d'images d'équipements dans les vues.
- Ajout de paramètres tel que la date et l'heure afin de permettre un meilleur suivi de l'archive.

IV.3. 7. Instrumentation :

On appelle instrumentation, l'ensemble des appareils de mesures destinés à capturer les valeurs des courants et de la température.

Ces appareils émettent des signaux normalisés que les récepteurs peuvent utiliser aux fins d'information, d'alerte ou de commande automatique. Les signaux électroniques utilisés varient entre 0 et 10 V.

Pour les informations logiques(TOR) sont récupérés directement à partir des contacts secs ou des relais.

IV.3. 7.1 .Récupération et branchement des entrées TOR:

On a utilisées les entrées TOR pour la récupération des alarmes et l'avertissement des anomalies qui se produisent dans la station, tel que les défauts générales redresseurs, onduleurs, climatisation ... ceux-ci proviennent essentiellement des contacts secs.

Le principe d'activation d'une alarme est basé sur celui du contact sec qui a deux états ; **NC** (NormallyClosed) normalement fermé, et **NO** (NormallyOpened) normalement ouvert.

L'alarme est activé a l'état NO du contact, donc lorsqu'aucune anomalie n'est présente, l'état du contact est NC (Figure .IV.32)



Figure IV. 32. État du contacte sec pour les alarmes.

Chaque contact sec est relié d'un côté à un équipement et de l'autre côté à un coffret d'alarme, d'où on peut récupérer tous les contacts et pouvoir les branchés à l'automate.

Par exemple l'alarme du défaut général du groupe électrogène est branchée entre le relais du groupe électrogène se trouvant sous son disjoncteur principal et les bornés 27 et 28 du coffret d'alarmes (**figure ci-dessous**).

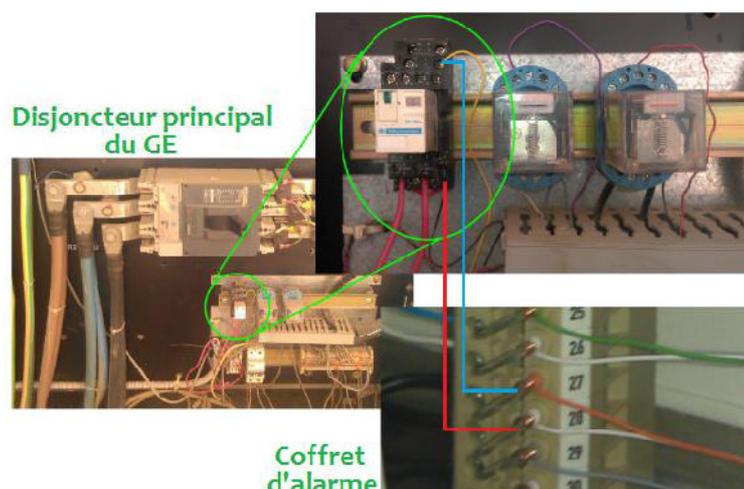


Figure IV. 33.Branchement des entrées TOR.

IV.3. 7.2 . Les capteurs analogiques :

On a choisi les capteurs analogiques pour récupérer la température de la salle d'énergie ainsi que le courant des trois phases, Pour le bon choix d'un capteur, il faut définir dans l'ordre :

- Le type de l'événement à détecter.
- La nature de cet événement.
- La grandeur de l'événement.
- L'environnement de l'événement.

IV.3. 7.2 .1. Capteur de courant :

Ce type de capteurs est déjà présent au niveau de la station OTA pour la mesure des courants alternatifs (0 à 1000 A) qui alimentent la salle d'énergie. Ce sont des transformateurs du courant.[14] :

Le transformateur est constitué par deux enroulements bobinés sur un circuit magnétique commun. Lorsqu'un courant i_1 passe dans l'un des bobinages B_1 , il crée par le circuit magnétique commun, un courant i_2 dans le bobinage B_2 . Le nombre de tours des enroulements et les courants i_1 et i_2 sont liés par la relation :

$$N_1 \times i_1 = N_2 \times i_2$$

Où N_1 et N_2 sont les nombres de tours de chaque enroulement.

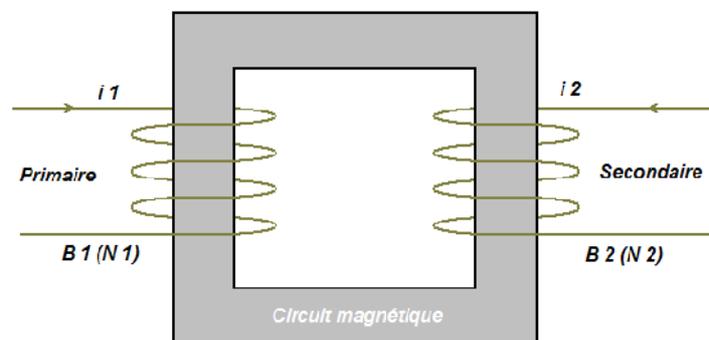


Figure IV. 34. Transformateur de courant.

Le problème qui existe lors de la récupération d'information, est due à la différence de l'ordre de grandeur de la plage de variation du capteur et de celle des modules d'entrées de l'automate choisit, la plage du capteur varie entre 0 à 5A, celle-ci est très grande par rapport aux plages des modules d'entrées analogiques qui varient entre 0 à 20 mA.

La solution proposée est de convertir la sortie courant en tension continue pour adapter les mesures au module de l'automate, cela dit la tension doit varier entre 0 et 10 V.

Afin de réaliser cette solution, la sortie courant est branché en entré dans un amplificateur opérationnel ce dernier est dimensionné de façon à ce que sa sortie ait une tension continue proportionnellement variable par rapport au courant, évoluant entre 0 et 10 V. Donc un rapport de transformation $V_S = \frac{R_2}{R_1} I$ où $\frac{R_2}{R_1} = \frac{V_S}{I} = \frac{V_{Smax}}{I_{max}} = \frac{10}{5} = 2$ on prendra $R_2 = 2k\Omega$ et $R_1 = R_3 = 1k\Omega$ (figure).

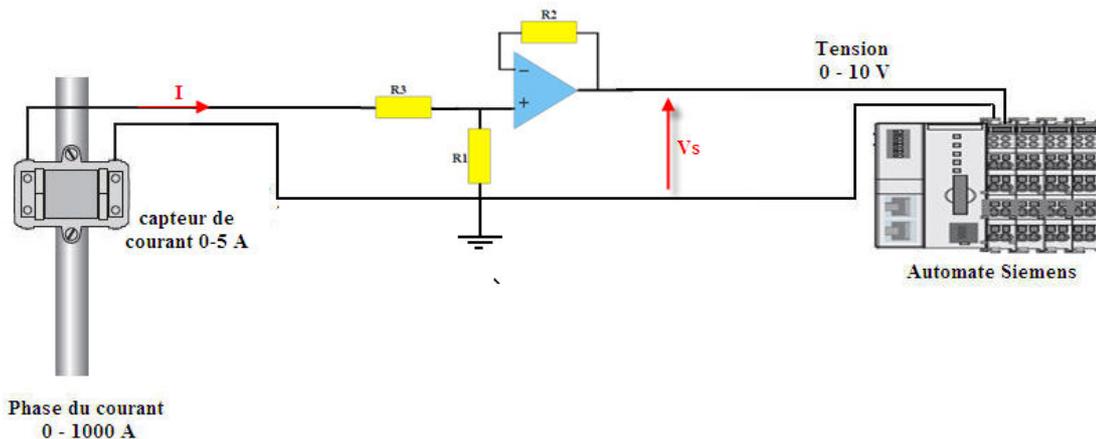


Figure IV. 35. Branchement du capteur courant à l'automate.

IV.3. 7.2 .2 Capteur de Température :

Il existe plusieurs types de capteurs de température (Annexe D), les plus utilisés dans le domaine de l'industrie sont :

- Thermomètre à résistance métallique
- Thermomètre à thermistance
- Thermomètre à thermocouple

Les critères de choix du type :

- La sensibilité de la mesure.
- La plage de variation 15 à 60°C ou le capteur doit être le plus efficace.
- Le temps de réponse réduit (temps réel).

Le Thermomètre à thermistance est celui qui répond le mieux à ces critères, ses caractéristiques sont les mieux adaptées à l'environnement de notre projet. [15] :

- Sensibilité jusqu'à l'ordre de 0,01°C.
- Plage de variation de température entre -110°C et 250°C.
- Temps de réponse relativement court par rapport aux autres thermomètres.

- Adaptable pour la mesure des températures avec ou sans contact.
- Imprécision inférieure à $0,1^{\circ}\text{C}$.

Finalement, Le modèle du capteur proposé est une sonde de température d'ambiance TB/TS à thermistance (figure IV. 36), qui mesure la température ambiante dans les locaux secs. Les caractéristiques techniques sont détaillées dans (Annexe D).



Figure IV. 36. Sonde de température TB/TS à thermistance. [16]

IV.3.7.3. Branchement de la sortie TOR :

Nous avons proposé pour la commande de démarrage du groupe électrogène, que la sortie de l'automate soit reliée à un relais qui actionne le démarrage du groupe. Ce relais sera branché en parallèle avec système de démarrage du GE. La tension de sortie excitant ce relais est de 24 V.

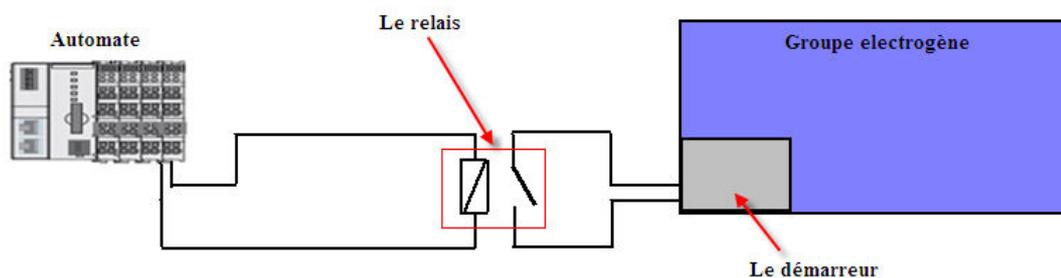


Figure IV. 37. Branchement de la sortie commande GE.

IV.4. Réalisation de l'application :

Pour la réalisation de notre application, nous avons fait les tests de notre programme sur un automate Siemens autre que celui que nous avons proposé.

L'automate S7 314 (6ES7 314-1AE04-0AB0/V1.2) avec lequel nous avons effectué les tests, possède le nombre nécessaire d'entrées TOR, mais malgré le nombre insuffisant d'entrées analogiques, et par cause d'indisponibilité d'automate que nous avons effectué le câblage et les essais de notre programme sur ce dernier (Figure IV. 38).



Figure IV. 38. Automate S7-314

Dans le tableau des borniers, nous avons testé le branchement des entrées TOR.



Figure IV39. Branchement des entrées TOR.

Pour le test des entrées analogiques, nous avons utilisé une carte électronique, dans laquelle on a varié la tension des entrées de 0 à 10 V avec des potentiomètres.



Figure IV. 39. Test des entrées analogiques.

IV.5. Conclusion :

D'après ce chapitre nous avons pu voir comment un automate peut faciliter la supervision d'une station ainsi que son suivie et cela depuis un poste de commande. Et comment booster son efficacité grâce aux bons choix des instruments, dela programmation visés (propre à l'installation) et de la bonne compréhension des attentes des clients(cahier de charges).

C'est à partir de cette combinaison que nous avons pu élaborer et concrétiser notre application. Qui permet la supervision de cette station ainsi que sont suivie. L'ajout de paramèretel que l'archivage des alarmes et des défauts, nous apermis de faciliter les opération de maintenance.

La réalisation et simulation en conditions réelles de l'application de notre projet , nous a permis d'acquérir plus de connaissance,lesquelles n'aurait pas été évidente si ce projet été purement théorique.Et c'est par celle-ci que nous avons pu corriger quelques lacunes qui au début nous semblait de meilleurs choix.

Conclusion générale :

Dans ce travail, l'objectif était de répondre à un besoin industriel, qui est la supervision de l'alimentation de la station OTA.

Dans un premier temps, nous avons présenté l'automate programmable industriel en décrivant son architecture et en citant les différents langages qui servent à le programmer. Nous avons aussi introduit un outil de programmation, à savoir, le STEP7 qui permet la programmation et la simulation de procédés automatisés avec différents langages normalisés.

Puis, nous avons présenté le logiciel de conception des interfaces Homme-Machine appelé «SIMATIC WinCC flexible». Ce dernier est un système de supervision de processus modulables qui offre des fonctions performantes de surveillance des automatismes.

L'automate programmable est l'un des appareils de commande des systèmes de production et d'automatisme les plus utilisés dans l'industrie, de ce fait, les notions présentées pourraient servir comme support de cours pour le module informatique industrielle.

Et enfin, on a détaillé le projet, l'environnement dans lequel il doit être appliqué ainsi que le cahier des charges auquel il devrait répondre. Après avoir réalisé des tests réels visant à valider le projet, nous avons, en dernière partie, proposé des solutions qui permettent de respecter le cahier des charges qui nous a été imposé.

Dans des travaux futurs on pourra envisager :

- Elargir la supervision aux reste des équipements de la station.
- Gestion des systèmes de protection et de sécurité.
- Améliorer la supervision par l'ajout de fonction de notification par envoie de sms et mail.

Bibliographie

- [1] Langages de programmation pour systèmes automatisés : norme CEI 61131-3.
- [2] A.gonzaga« les automates programmables industriels », Cours.
- [3] G.MICHEL, « Les A.P.I Architecture et application des automates programmables industriels », Edition DUNOD, 1987.
- [4] J. HUGH « AutomatingManufacturingSystemswithPLC's » (1993-2005).
- [5] P.JARGOT, « Langages de programmation pour API. Norme IEC 1131-3 », Techniques de l'ingénieur, Vol. S 8 030.
- [6] Langages de programmation pour systèmes automatisés : norme CEI 61131-3.
- [7] Online,www.siemens.com.
- [8] SIMATICStep 7 « programmation avec STEP7 »manuel Siemens.
- [9] SIMATIC « Caractéristiques techniques Manuel » référencée « 6ES7398-8FA10-8CA0 ».
- [10] SAHAR Belkacem Amine et SOUCHANE AmineCommande et supervision de la station FESTO PCS COMPACT à l'aide de STEP7 et WinCCENP,Alger 2009.
- [10] SIMATIC. « Programmer avec STEP 7 » référencée « 6ES7810-4CA10-8CW0 ».
- [11] SIMATIC HMI « WinCC flexible 2008 Compact/Standard/Advanced » référencé 6AV6691-1AB01-3AC0.
- [12] SIMATIC HMI « WinCC flexible 2008 Runtime » référence:6AV6691-1BA01-3AC0.
- [13] Documentation station OTA ,2014.
- [14] WAGO CONTACT « Solutions pour la mesure de courant et d'énergie » 002 · 0888-0599/0100-3501, 2003.
- [15] GASTON J.Beaudoin « Instrumentation et automation » n : 2-89443- 003- 5, 1994.
- [16] « Capteur de température ambiante Thermistance », Fiche technique TA200603 Issue 1/E ,2004.

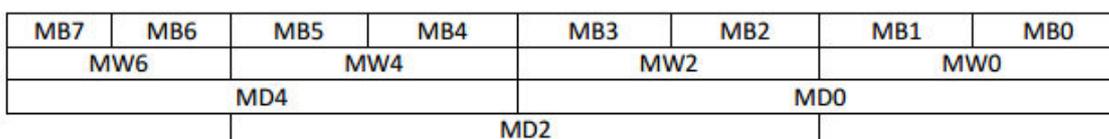
ANNEXE A

STEP 7

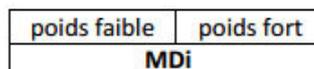
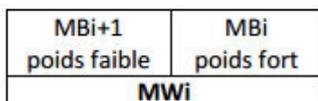
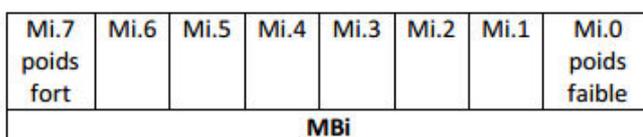
A.1. Adressage des variables Siemens:

- Bits internes M0.0 à M255.7 dépendants des mots suivants
- Octets internes : ensemble de 8 bits MBi
- Mots internes : ensemble de 16 bits MWi
- Mots doubles : ensemble de 32 bits MDi

Principe de l'adressage des mots internes : (attention aux chevauchements des mots)



Poids fort et poids faible :



A.2. Description des principaux outils du logiciel STEP 7

Langage de programmation CONT (Schéma à contact) :

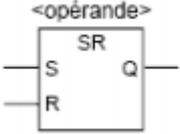
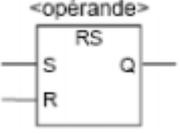
Ce langage de programmation fait partie du logiciel base STEP7, il s'inspire des schémas de circuits électriques, les éléments d'un schéma de circuit sont rassemblés dans des réseaux, un ou plusieurs réseaux forment la section des instructions complète d'un bloc de code. Dans le langage CONT, on crée le programme en utilisant un éditeur incrémental.

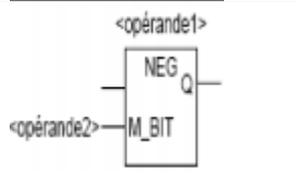
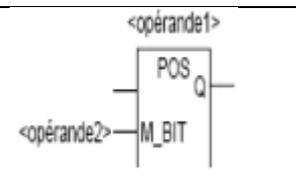
1. Opérations combinatoires sur bits :

Les opérations combinatoires sur bits utilisent deux chiffres : 1 et 0 (chiffres binaire), pour les contacts et les bobines, 1 signifie activé et 0 signifie désactivé.

Ces opérations évaluent les états de signal 1 et 0 et les combinent selon la logique booléenne, le résultat est égal soit 1 ou 0, c'est le résultat logique (RLG). Les opérations combinatoires sur bits existantes sont données dans le tableau suivant :

Noms	Représentation	Fonctionnement
Contact à fermeture	<opérande> --- ---	Le contact est fermé si la valeur du bit interrogé sauvegardée en <opérande> égale 1 alors RLG = 1, en revanche, si l'état de <opérande> est 0, le contact est ouvert et RLG= 0.
Contact à ouverture	<opérande> --- / ---	Le contact est fermé si la valeur du bit interrogé sauvegardée en <opérande> égale 0. Dans ce cas le RLG =1, en revanche, si l'état de <opérande> est 1, le contact est ouvert et RLG = 0.
Sauvegarder RLG dans RB	---(SAVE)	Cette opération sauvegarde le RLG dans le bit RB du mot d'état, dans ce cas le bit de première interrogation /PI n'est pas affecté, ainsi une opération ET dans le réseau suivant prendra en considération l'état du bit RB.
Bobine de sortie	<opérande> ---()	Cette opération fonctionne comme une bobine dans un schéma à relais. Si le RLG = 1, le bit en <opérande> est mise à 1 et si RLG = 0, le bit en <opérande> est mis à 0.
Connecteur	<opérande> ---(#)---	C'est un élément d'affectation intermédiaire qui mémorise le bit RLG dans l'<opérande> précisé.
Inverseur de RLG	--- NOT ---	Cette opération inverse le bit de résultat logique (RLG).
Mettre à 1	<opérande> ---(S)	Cette opération permet d'effectuer un SET sur l'opérande si le RLG des opérations précédentes a la valeur 1, dans ce cas, l'<opérande> précisé de l'élément est mis à 1, si RLG = 0 l'opérande reste inchangé
Mettre à 0	<opérande> ---(R)	Cette opération permet d'effectuer un RESET sur l'opérande si le RLG des opérations précédentes a la valeur 1, dans ce cas, l'<opérande> précisé de l'élément

		est mis à 0, si RLG = 0 l'opérande reste inchangé.
Bascule mise à 1, mise à 0		<p>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, la bascule est mise à 0. Si le RLG est égal à 1 aux deux entrées, c'est l'ordre qui compte : la bascule SR exécute d'abord la mise à 1, puis la mise à 0 de l'<opérande>indiqué,les opérations S et R s'exécutent uniquement si le RLG égale 1.</p>
Bascule mise à 0, mise à 1		<p>Cette opération exécute la mise à 0 si l'état de signal est 1 à l'entrée R et 0 à l'entrée S. Si l'état de signal est 0 à l'entrée R et 1 à l'entrée S, la bascule est mise à 1. Si le RLG est égal à 1 aux deux entrées, c'est l'ordre qui compte : la bascule RS exécute d'abord la mise à 0, puis la mise à 1 de l'<opérande> indiqué, les opérations S et R s'exécutent uniquement si le RLG égale 1.</p>
Détecter front descendant	<p><opérande> ---(N)---</p>	<p>Cette opération détecte le passage de 1 à 0 de l'état de signal de l'opérande et montre cette transition avec un RLG égal à 1 après cette opération.</p>
Détecter front montant	<p><opérande> ---(P)---</p>	<p>L'opération détecte le passage de 0 à 1 de l'état de signal de l'opérande et montre cette transition avec un RLG égal à 1 après opération.</p>

Détecter front descendant de signal		Cette opération compare l'état de <opérande1> à celui provenant de l'interrogation d'état de signal précédent figurant dans <opérande2>.
Détecter front montant de signal		Cette opération compare l'état de signal de <opérande1> à celui provenant de l'interrogation d'état de signal précédent figurant dans <opérande2>.

A.3. Opérations de comparaison :

Les opérations de comparaison comparent les entrées IN1 et IN2, si la comparaison est vraie, son résultat logique (RLG) est 1 ; sinon, il est égal à 0, dans le langage CONT on possède les opérations de comparaison suivantes :

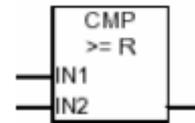
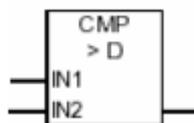
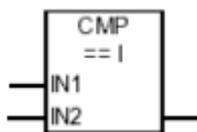
1. CMP I : Comparer entiers de 16 bits.
2. CMP D : Comparer entiers de 32 bits.
3. CMP R : Comparer nombres réels

Les types de comparaison sont :

1. == IN1 égal à IN2
2. <> IN1 différent de IN2
3. > IN1 supérieur à IN2
4. < IN1 inférieur à IN2
5. >= IN1 supérieur ou égal à IN2
6. <= IN1 inférieur ou égal à IN2

Leurs représentations se fait par des blocs, ou à l'intérieur on mentionne l'opération et le type de comparaison par les symboles qu'on a ci-dessus.

Exemples :

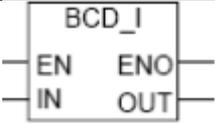
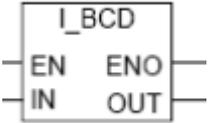
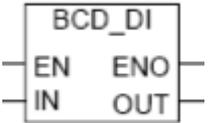
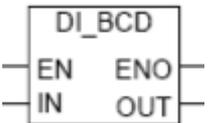
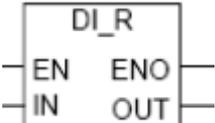
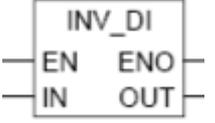
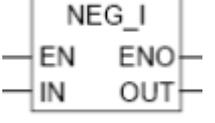


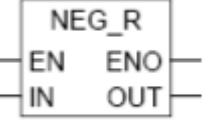
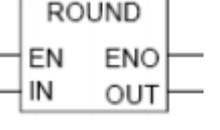
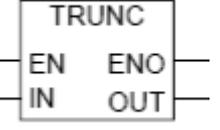
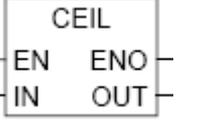
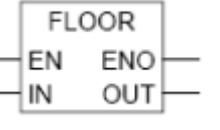
Egalité entre 2 entiers comparaison entre 2 entiers comparaison entre 2 réel : 16 bits 32 bits : IN1 supérieur à IN2 IN1 supérieur ou égal à IN2

A.4. Opérations de conversion :

Les opérations de conversion lisent le contenu du paramètre d'entrée IN, le convertissent ou en changent le signe. Le résultat est rangé dans le paramètre de sortie OUT.

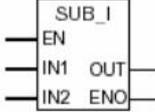
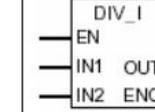
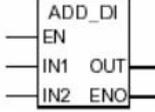
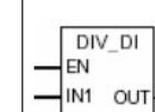
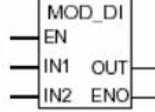
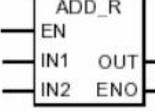
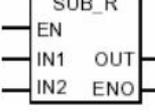
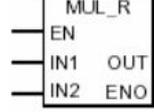
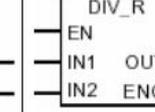
Les opérations de conversion sont données dans le tableau suivant :

Représentation	Fonctionnement
	Cette opération permet de convertir un nombre en BCD en un nombre entier de 16 bits
	Cette opération permet de convertir nombre entier sur 16 bits en un nombre en BCD.
	Cette opération permet de convertir nombre entier sur 16 bits en un entier sur 32 bits
	Cette opération permet de convertir un nombre en BCD en un nombre entier de 32 bits
	Cette opération permet de convertir nombre entier sur 32 bits en un nombre en BCD.
	Cette opération permet de convertir nombre entier sur 32 bits en un nombre réel.
	Cette opération permet de donner le complément à 1 d'un entier sur 16 bits
	Cette opération permet de donner le complément à 1 d'un entier sur 32 bits
	Cette opération permet de donner le complément à 2 d'un entier sur 16 bits

	<p>Cette opération permet de donner le complément à 2 d'un entier sur 32 bits</p>
	<p>Cette opération permet d'inverser le signe d'un nombre réel</p>
	<p>Cette opération permet d'arrondir un réel en un entier de 32 bits.</p>
	<p>Cette opération permet de prendre la valeur entière d'un réel et la mettre sous forme d'un entier de 32 bits</p>
	<p>Cette opération permet de convertir un nombre réel en un nombre entier supérieur le plus proche</p>
	<p>Cette opération permet de convertir un nombre réel en un nombre entier inférieur le plus proche</p>

A.5. Opérations arithmétiques :

Les opérations arithmétiques sont effectuées entre les entrées IN1 et IN2 qui sont des réels ou des entiers sur 16 bits ou sur 32 bits. Si l'état de signal est 1 à l'entrée de validation EN, le résultat est rangé dans la sortie OUT etsi ce résultat est hors de la plage autorisée pour un nombre entier de 16 bits, les bits DEB et DM du mot d'état sont mis à 1 et la sortie ENO est à mise à 0. Ainsi, les opérations suivant les opérations arithmétiques et qui y sont connectées par ENO (cascade) ne sont pas exécutées.

opérations	Addition	Soustraction	Multiplication	Division	Reste de division
Entiers sur 16 bits					—
Entiers sur 32 bits					
Réels					—

ANNEXEB

WinCC

Les « objets simples » de la bibliothèque de WinCC :

Icône	Objet	Remarques
	"Ligne"	On peut choisir des terminaisons de lignes droites, arrondies ou en forme de flèche.
	"Ligne polygonale"	Une ligne polygonale est composée de sections reliées entre elles et peut contenir un nombre quelconque de points d'angle. Les points d'angle sont numérotés dans l'ordre de leur création. On peut les modifier ou les supprimer. On peut choisir des terminaisons de lignes droites, arrondies ou en forme de flèche. L'objet ligne polygonale est un objet ouvert. Même lorsque le point de départ et le point d'arrivée possèdent les mêmes coordonnées, la surface ne peut pas être remplie.
	"Polygone"	Les points d'angle d'un polygone sont numérotés dans l'ordre de leur création. On peut remplir un polygone d'une couleur ou d'un motif.
	"Ellipse"	On peut remplir l'ellipse d'une couleur ou d'un motif.
	"Cercle"	On peut remplir le cercle d'une couleur ou d'un motif.
	"Rectangle"	On peut définir un arrondi quelconque pour les angles du rectangle. On peut remplir le rectangle d'une couleur ou d'un motif.
	"Champ de texte"	Un champ de texte permet de représenter un texte d'une ou plusieurs lignes et de lui affecter une couleur et une police. On peut donner au champ une couleur ou un motif comme arrière-plan.
	"Champ E/S"	<p>Un champ d'E/S peut posséder les fonctions suivantes en runtime :</p> <ul style="list-style-type: none"> • Sortie des valeurs d'une variable • Entrée de valeurs par l'opérateur ; les valeurs saisies sont enregistrées dans une variable. • Entrée et sortie combinée ; dans ce cas, l'opérateur peut éditer et redéfinir la valeur de la variable. <p>On peut définir des valeurs limites pour les valeurs de variable représentées dans le champ d'E/S. Si l'entrée de l'opérateur ne doit pas être visible en runtime, configurons "Entrée masquée".</p>

	<p>"Champ date/heure"</p>	<p>Un champ date/heure peut posséder les fonctions suivantes en runtime :</p> <ul style="list-style-type: none"> • Sortie de la date et de l'heure • Entrée et sortie combinée ; dans ce cas, l'opérateur peut éditer et redéfinir la date et l'heure. <p>On peut sortir l'heure système et une variable correspondant en tant que source pour la date et l'heure.</p> <p>On peut choisir d'affichera date sous une forme extensive(mardi 16Avril 2014) ou abrégée (16.04.2014).</p>
---	---------------------------	--

Icône	Objet	Remarques
	<p>"Champ E/S graphique"</p>	<p>Un champ d'E/S graphique peut posséder les fonctions suivantes en runtime :</p> <ul style="list-style-type: none"> • Sortie des entrées d'une liste de graphiques • Entrée et sortie combinée ; dans ce cas, l'opérateur peut sélectionner un graphique dans la liste des graphiques et modifier le contenu du champ d'E/S graphique. <p>Exemple d'utilisation comme champ de sortie :</p> <p>pour afficher en runtime l'état d'une vanne, on sort dans le champ d'E/S graphique la représentation d'une vanne ouverte ou fermée.</p>
	<p>"Champ E/S symbolique"</p>	<p>un champ d'E/S symbolique peut posséder les fonctions suivantes en runtime</p> <ul style="list-style-type: none"> • Sortie des enregistrements d'une liste de texte • Entrée et sortie combinée ; dans ce cas, l'opérateur peut sélectionner un texte dans la liste de textes et modifier le contenu du champ d'E/S symbolique. <p>Exemple d'utilisation comme champ combiné d'entrée et de sortie:</p> <p>Pour piloter un moteur en runtime, l'utilisateur sélectionne dans la liste de textes un des textes "Moteur ARRÊT" ou "Moteur MARCHÉ". Selon la sélection, le moteur est démarré ou arrêté, le champ d'E/S symbolique affiche l'état du moteur (Moteur ARRÊT/Moteur MARCHÉ)</p>
	<p>"Vue de graphique"</p>	<p>La vue de graphique permet d'afficher dans une vue des graphiques qui ont été créés dans des programmes graphiques externes. On peut afficher des graphiques aux formats suivants : "*.emf", "*.wmf", "*.dib", "*.bmp", "*.jpg", "*.jpeg", "*.gif"et "*.tif".</p> <p>La vue de graphique permet d'insérer sous forme d'objets OLE (Object Linking and Embedding) des graphiques qui ont été créés dans des programmes graphiques externes. Dans la fenêtre des propriétés de cette vue, il est possible d'appeler les objets OLE et de les éditer directement dans le</p>

		programme avec lequel ils ont été créés.
	"Bouton"	Les boutons permettent de piloter un procès. on configure l'exécution de fonctions ou de scripts sur le bouton.
	"Commutateur"	Le commutateur sert en runtime à l'entrée et à l'affichage de deux états, p.ex. MARCHE/ARRÊT ou Enfoncé/Relâché. Le commutateur peut porter un texte ou un graphique qui représente son état en runtime
	"Bargraphe"	Un bargraphe permet de représenter en runtime une valeur provenant de l'automate sous forme de barre munie d'une échelle. Un barographe permet d'afficher p.ex. les variations des niveaux de remplissage.

Icône	Objet	Remarques
	"Réglette"	Une réglette permet l'opérateur de lire et de saisir les valeurs numériques de la manière suivante. <ul style="list-style-type: none"> • La position du curseur affiche une valeur actuelle provenant de l'automate. • Pour saisir les valeurs, l'opérateur déplace le curseur sur la valeur voulue. La réglette est configurable uniquement verticalement.
	"Horloge"	L'horloge permet d'afficher l'heure système sur le pupitre en runtime en analogique ou en numérique.
	"Visualisation/forçage"	La fonction Visualisation/forçage permet à l'opérateur d'accéder directement en lecture et en écriture aux zones d'adressage de l'automate SIMATIC S7 ou SIMATIC S5 connecté.
	"Vue Sm@rtClient"	L'affichage Sm@rtClient permet à l'opérateur d'utiliser le contrôle-commande sur un autre pupitre de l'installation.
	"Navigateur HTML"	Le navigateur HTML permet d'afficher des pages HTML pour l'opérateur.
	"Vue des utilisateurs"	Dans WINCC flexible, nous pouvons protéger le pilotage des objets de bibliothèque par un mot de passe. La vue des utilisateurs ("vue personnalisée") permet à un administrateur de gérer les utilisateurs du pupitre au runtime. Un utilisateur sans droits d'administrateur peut modifier son mot de passe en runtime via la vue des utilisateurs.
	"Instrument à aiguille"	L'instrument à aiguille permet d'afficher en runtime des valeurs numériques via des aiguilles.

		<p>L'apparence de l'instrument à aiguille est configurable. On peut p.ex. modifier la couleur d'arrière-plan du graphique ou la représentation de l'échelle.</p>
	<p>"Vue de la courbe"</p>	<p>Avec la vue de courbe, On peut représenter en même temps plusieurs courbes dont les valeurs proviennent de l'automate ou d'une archive. Les axes de la vue de courbe sont configurables (échelles, unités, etc.)</p>
	<p>"Vue simple de recette"</p>	<p>La vue simple de recette permet d'afficher, d'éditer et de gérer les enregistrements en runtime.</p>
	<p>"Vue des alarmes"</p>	<p>Dans l'affichage d'alarmes, l'opérateur peut consulter en runtime certaines alarmes ou événements provenant du tampon d'alarmes ou de l'archive d'alarmes.</p>

Tableau III. 1. Objets simples du WinCC flexible.

ANNEXE C
APPLICATION

C.1.dition des mnémoniques :

Les différentes variables de notre projet sont déclarées dans le STEP 7 comme suite :

Mnémonique	opérande	Type de variable	Commentaire
DEM GE	A 0.1	BOOL	Sortie démarrage GE
CAPT CLIM	E 124.0	BOOL	Entrée alarme défaut climatisation (salle d'énergie)
CAPT REDR SAFT1	E 124.1	BOOL	Entrée alarme défaut redresseur SAFT1 (urgent)
CAPT REDR DELTA	E 124.2	BOOL	Entrée alarme défaut redresseur DELTA (urgent)
CAPT défaut GE	E 124.3	BOOL	Entrée alarme défaut groupe électrogène
CAPT OND SAFT 1	E 124.4	BOOL	Entrée alarme défaut Onduleur SAFT1 (urgent)
CAPT OND EMERSON	E 124.5	BOOL	Entrée alarme défaut Onduleur EMERSON(urgent)
CAPT niveau gasoil bas	E 124.6	BOOL	Entrée alarme niveau gasoil BAS
CAPT GE ON	E 124.7	BOOL	Entrée information alimentation sur GE.
CAPT réseau ON	E 125.0	BOOL	Entrée information alimentation sur réseau
CAPT DISJ TRANS	E 125.1	BOOL	Entrée défaut du disjoncteur poste de transformation
CAPT CELL TRANS	E 125.2	BOOL	Entrée défaut cellule de protection poste de transformation
m- Clim	M 0.0	BOOL	Mémoire alarme climatisation
m - REDR MTP1	M 0.1	BOOL	Mémoire alarme défaut redresseur SAFT 1
m -REDR DELTA	M 0.2	BOOL	Mémoire alarme défaut redresseur DELTA
m- défaut GE	M 0.3	BOOL	Mémoire alarme défaut GE
m- OND SAFT 1	M 0.4	BOOL	Mémoire alarme défaut Onduleur SAFT 1
m-EMERSON	M 0.5	BOOL	Mémoire alarme défaut Onduleur EMERSON
m-température 2 seuil	M 0.6	BOOL	Mémoire alarme dépassement des 2 seuil température
m-OND SAFT1+ EMERSON	M 0.7	BOOL	Mémoire alarme défaut des 2 onduleurs SAFT ou EMERSON
m-niveau de gasoil bas	M 1.0	BOOL	Mémoire alarme niveau gasoil bas
m-Salle énergie ON	M 1.1	BOOL	Mémoire information salle d'énergie est alimenté

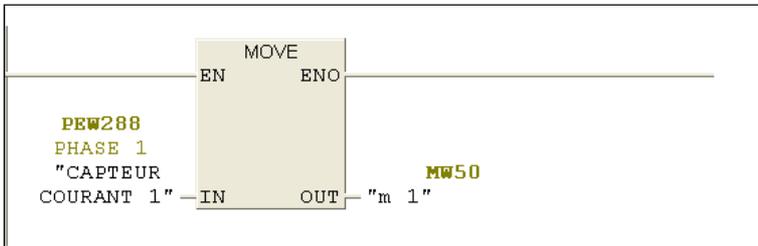
m-REDR SAFT1+DELTA	M 1.2	BOOL	Mémoire alarme défaut des 2 redresseurs SAFT1 ou DELTA
m-GE ON	M 1.3	BOOL	Mémoire information alimentation sur GE
m-Réseau ON	M 1.4	BOOL	Mémoire information alimentation sur réseau
m-gasoil bas+Défaut GE	M 1.5	BOOL	Mémoire alarme défaut de GE : Gasoil bas ou défaut générale
m-alarmes courant	M 1.6	BOOL	Mémoire alarme dépassement de seuils pour les trois phases
m- temp haute	M 1.7	BOOL	Mémoire alarme température > 35°C
m-temp basse	M 2.0	BOOL	Mémoire alarme température < 20°C
m-C1 >590A	M 2.1	BOOL	Mémoire alarme courant Ph1>590A
m-C2 >590A	M 2.2	BOOL	Mémoire alarme courant Ph2>590A
m-C3 >590A	M 2.3	BOOL	Mémoire alarme courant Ph3>590A
m-C1 >630A	M 2.4	BOOL	Mémoire alarme courant Ph1>630A
m-C2 >630A	M 2.5	BOOL	Mémoire alarme courant Ph2>630A
m-C3 >630A	M 2.6	BOOL	Mémoire alarme courant Ph3>630A
m- DISJ TRANS	M 2.7	BOOL	Mémoire défaut du disjoncteur poste de transformation
m- CELL TRANS	M 3.0	BOOL	Mémoire défaut cellule de protection poste de transformation
m-commande GE	M 100.0	BOOL	Mémoire commande démarrage de GE
COURANT 1	MD 20	REAL	Valeur courant real phase 1
COURANT 2	MD 24	REAL	Valeur courant real phase 2
COURANT 3	MD 28	REAL	Valeur courant real phase 3
TEMPERATURE	MD 32	REAL	Valeur température real salle d'énergie
CAPT ANAL C1	PEW 752	INT	Entrée analogique Courant phase 1
CAPT ANAL C 2	PEW 754	INT	Entrée analogique Courant phase 2
CAPT ANAL C 3	PEW 756	INT	Entrée analogique Courant phase 3
CAPT ANAL température	PEW 758	INT	Entrée analogique température
SCALE	FC 105	FC 105	Fonction de la mise à l'échelle analogique

Tableau :Les mnémoniques déclarées du projet.

C.2. Le programme complet de notre projet

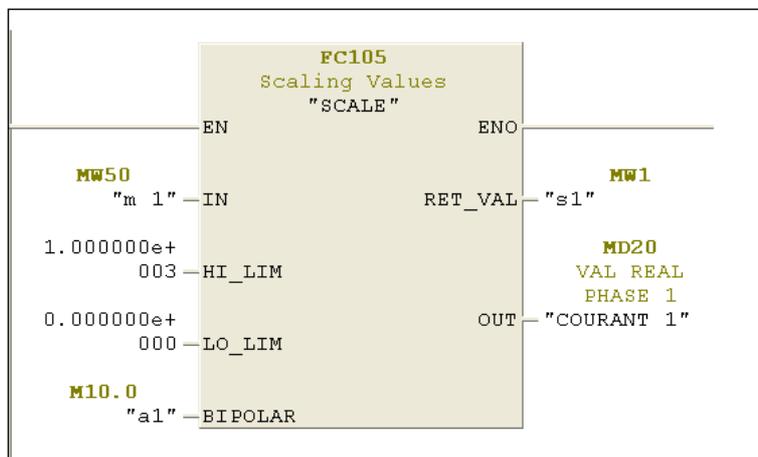
Réseau 1: conversion

l'entrée est convertie de la valeur entiere au word



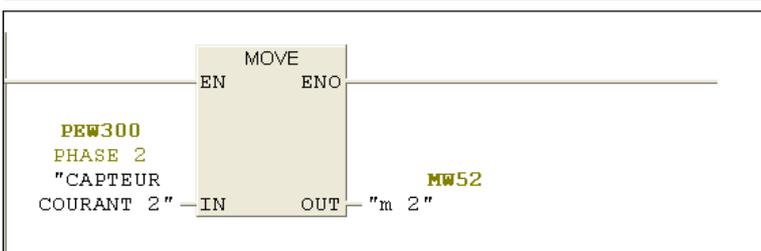
Réseau 2: Valeur réel courant phase1

Mise à l'échelle du courant de la phase 1.



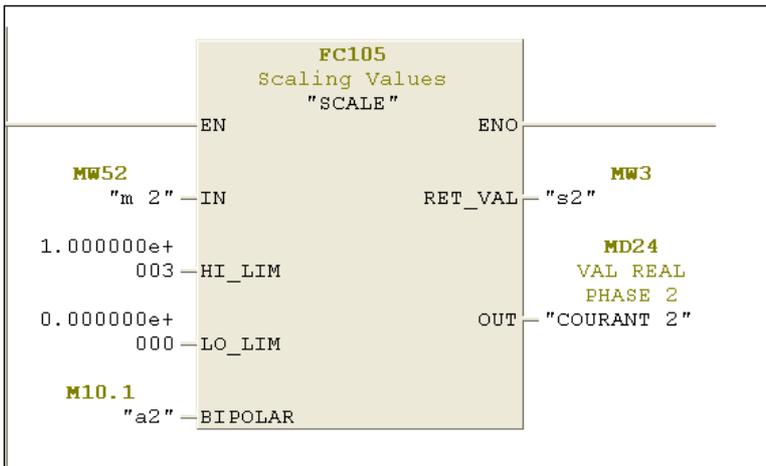
Réseau 3: conversion

l'entrée est convertie de la valeur entiere au word



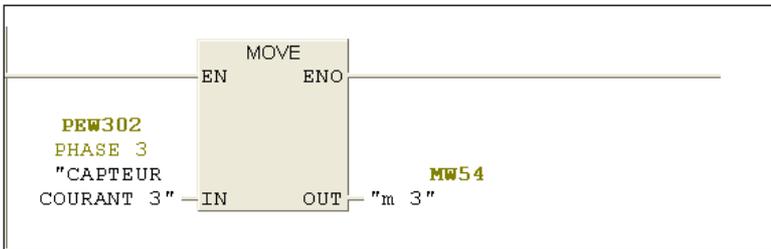
Réseau 4: Valeur réel courant phase2.

Mise à l'échelle du courant de la phase 2.



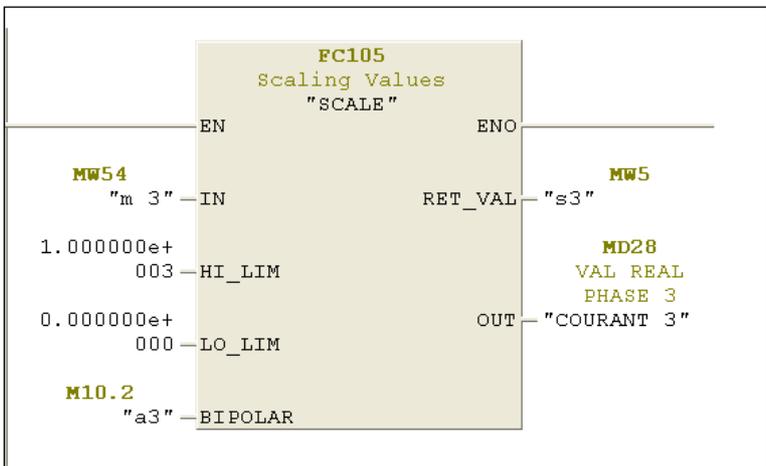
Réseau 5: conversion

l'entrée est convertie de la valeur entière au word



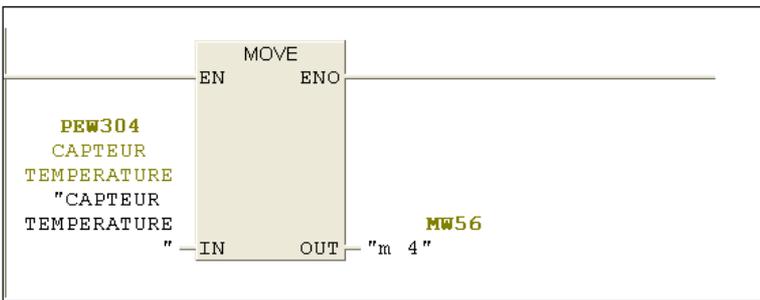
Réseau 6: Valeur réel courant phase 3.

Mise à l'échelle du courant de la phase 3.



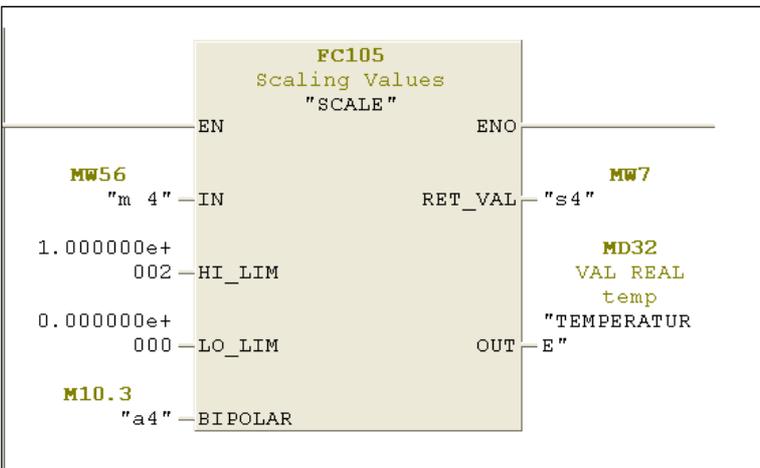
Réseau 7: conversion

l'entrée est convertie de la valeur entière au word



Réseau 8: Entrée analogique température

Mise à l'échelle de la température de la salle d'énergie.



Réseau 9: climatisation (salle d'énergie)

récupération de l'alarme du :
.Défaut générale climatisation



Réseau 10: redresseur SAFT1

récupération de l'alarme du :
.Défaut générale du redresseur SAFT



Réseau 11 : redresseur DELTA

récupération de l'alarme du :
•Défaut générale du redresseur DELTA



Réseau 12 : redresseurs

Mémoirisation de l'alarme du redresseur
concernent les deux défauts des 2 redresseurs :SAFT1 et/ou DELTA



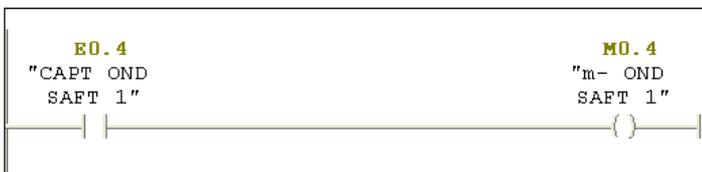
Réseau 13 : groupe électrogène

récupération de l'alarme du :
.défaut groupe électrogène



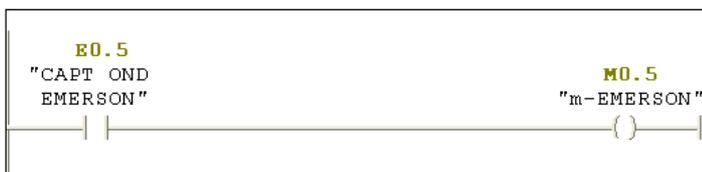
Réseau 14 : ONDULEUR SAFT 1

récupération de l'alarme du :
• Défaut générale onduleur SAFT



Réseau 15 : ONDULEUR EMERSON

récupération de l'alarme du :
•Défaut générale onduleur EMERSON



Réseau 16: onduleurs

Mémoirisation de l'alarme du redresseur
concernent les deux defauts des 2 onduleurs :SAFT1 et/ou EMERSON



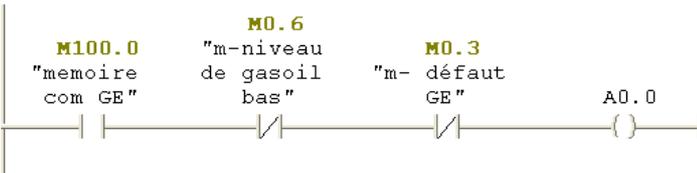
Réseau 17: Groupe Electrogène niveau gasoil

récupération de l'alarme:
.alerte niveau gasoil BAS



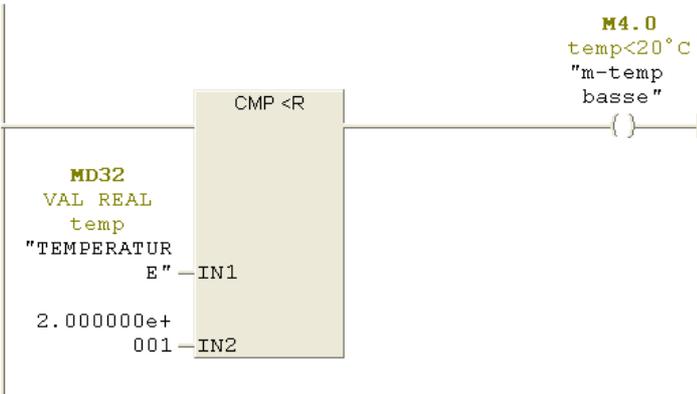
Réseau 18: Groupe Electrogène

commande de démarrage du GE



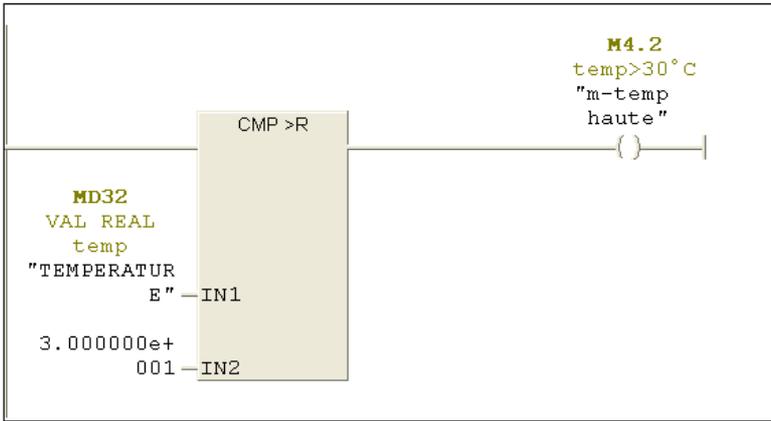
Réseau 19: température salle d'énergies

récupération de l'alarme:
.alerte température salle d'énergies inferieure à 20 °C



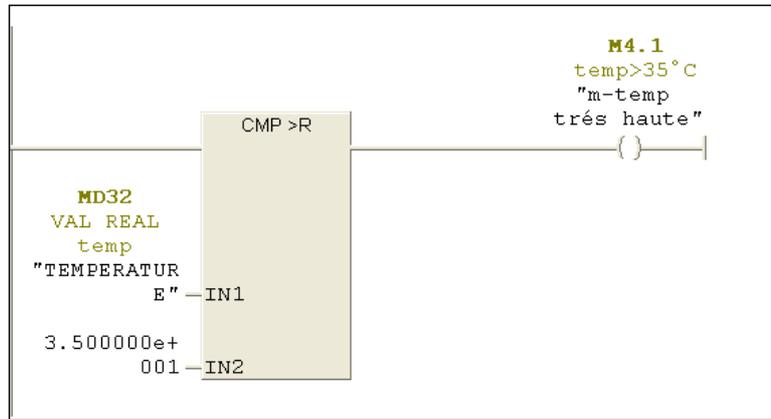
Réseau 20: température salle d'énergies

récupération de l'alarme:
.alerte température salle d'énergies supérieure 30 °C



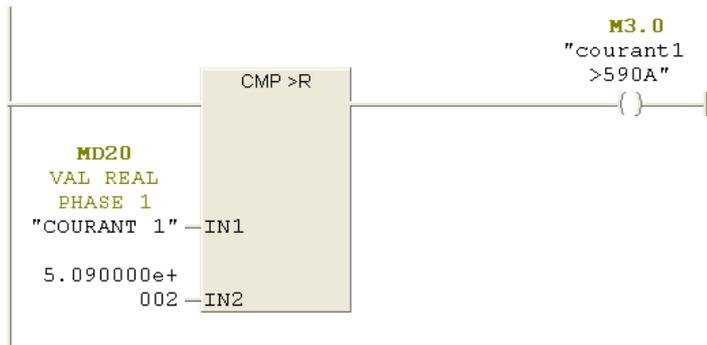
Réseau 21: température salle d'énergies

récupération de l'alarme:
.alerte température salle d'énergies supérieure 35 °C



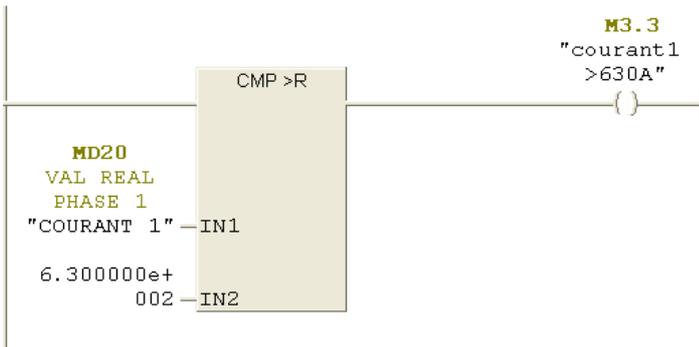
Réseau 22: Courant phase 1

récupération de l'alarme:
.alerte le courant de la phase 1 dépasse le seuil 590 A



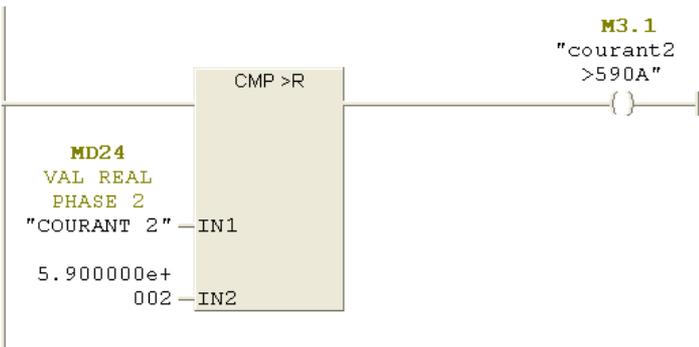
Réseau 23: Courant phase 1

récupération de l'alarme:
.alerte le courant de la phase 1 dépasse le seuil 630 A



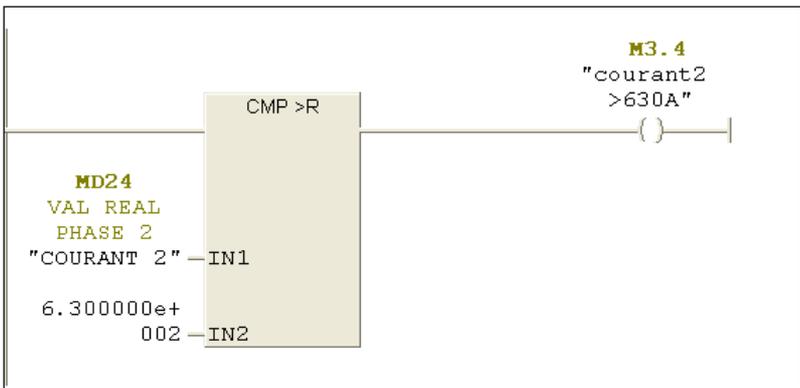
Réseau 24: Courant phase 2

récupération de l'alarme:
.alerte le courant de la phase 2 dépasse le seuil 590 A



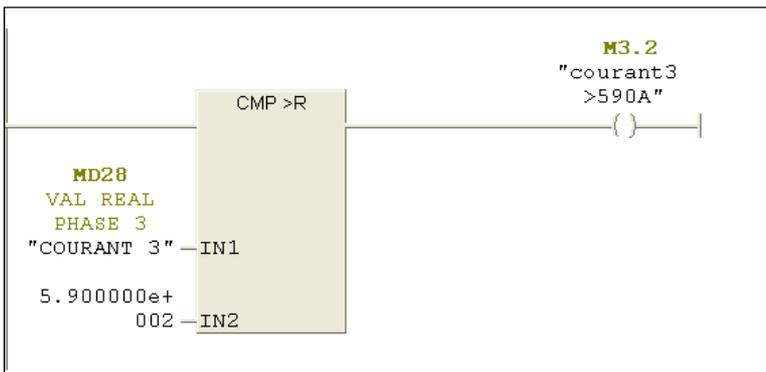
Réseau 25: Courant phase 2

récupération de l'alarme:
.alerte le courant de la phase 2 dépasse le seuil 630 A



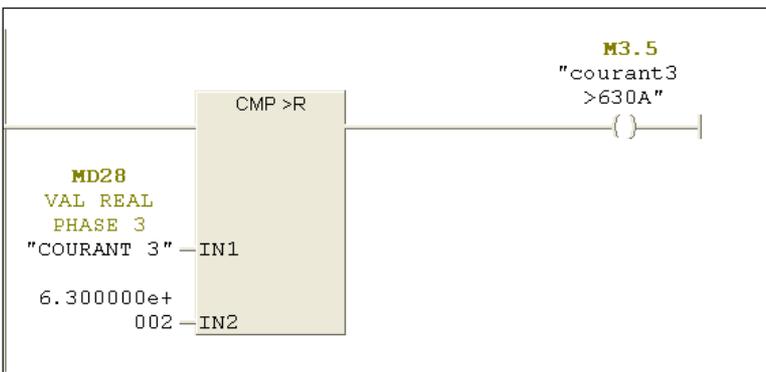
Réseau 26: Courant phase 3

récupération de l'alarme:
.alerte le courant de la phase 3 dépasse le seuil 590 A



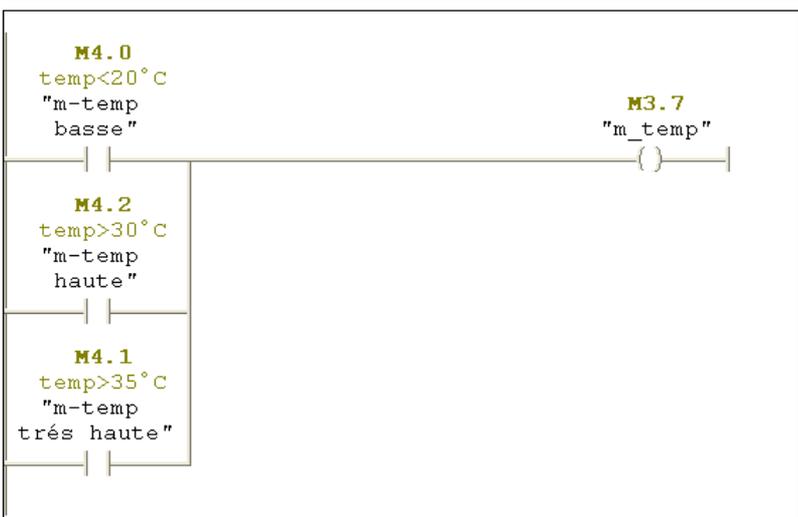
Réseau 27: Courant phase 3

récupération de l'alarme:
.alerte le courant de la phase 3 dépasse le seuil 630 A



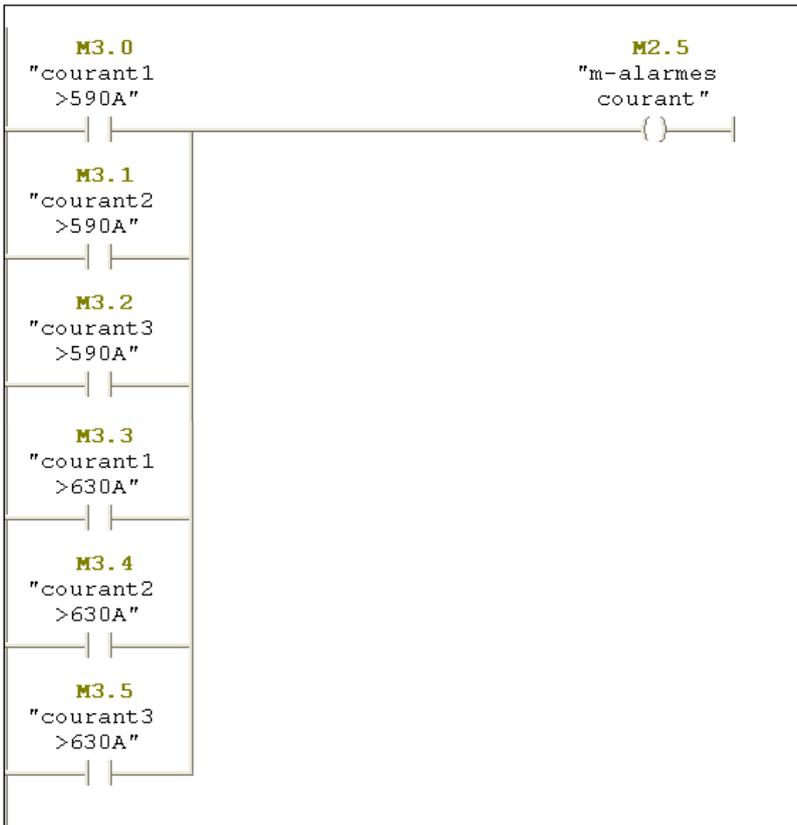
Réseau 28: température 3 seuil

Mémoirisation de l'alarme de température
concernent les Trois alerte dépassement de seuils.



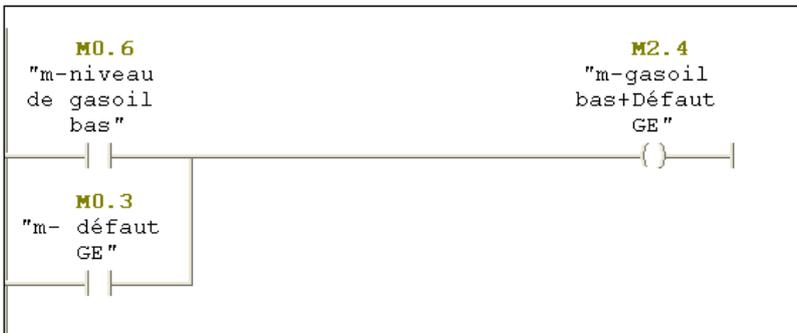
Réseau 29: dépassement de seuils courant ph1,2,3.

Mémoirisation de l'alarme de courants concernant les alerte de dépassement des deux seuils 590 et 630 a pour les trois phases



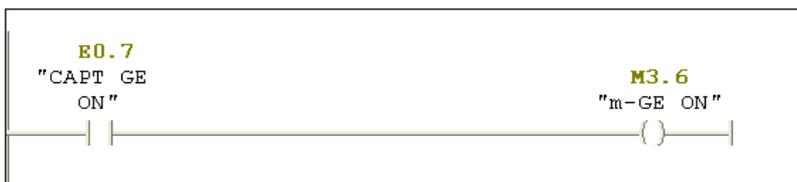
Réseau 30: Groupe électrogène.

Mémoirisation de l'alarme du groupe électrogène concernant les deux défauts du GE.



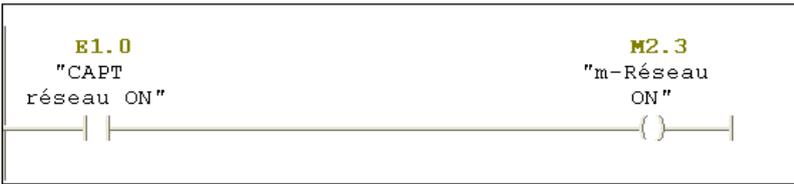
Réseau 31: Informations état du réseau

Alimentation sur groupe électrogène.



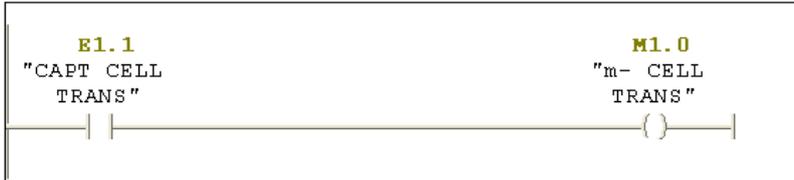
Réseau 32: Informations état du réseau

Alimentation sur réseau électrique SONELGAZ



Réseau 33: poste de transformation

récupération de l'alarme de la cellule de protection du poste de transformation



Réseau 34: poste de transformation

récupération de l'alarme du disjoncteur du poste de transformation



Réseau 35: poste de transformation

Mémoirisation de l'alarme du poste de transformation concernant les deux défauts fusible et disjoncteur.



Réseau 36: information alimentation du reseau

definie quel source alimente la salle d'energie



Annexe D
Application

D.1. Capteurs de température

D.1.1. Thermomètre à résistance métallique

D.1.1.A. Le capteur ou la sonde

Le capteur de ces thermomètres, appelés aussi RTD (résistance température détecteur) est un conducteur métallique, généralement en forme de fil, parcouru par un courant électrique. Sa résistance augmente, lorsque la température du milieu dans lequel il est plongé s'élève, selon la loi générale simplifiée :

$$R_{t_2} = R_{t_1} [1 + \alpha (t_2 - t_1)]$$

R_{t_2} et R_{t_1} : résistances à t_1 , et t_2

α : coefficient positif de température, spécifique du conducteur utilisé ; il définit la variation d'une résistance de un ohm par degré K : α étant un coefficient variable avec la température, l'équation n'est applicable que pour des variations de température relativement faibles et la valeur de α est une valeur moyenne. Ce coefficient est exprimé en ($\Omega/\Omega \cdot K$) ou simplement en K^{-1} il a donc les mêmes dimensions que la sensibilité du transducteur.

Les caractéristiques souhaitables pour le métal du fil utilisé comme

- un coefficient de température élevé (grande sensibilité) ;
- une résistivité élevée ;
- un respect satisfaisant de la loi générale de linéarité ;
- une bonne stabilité dans le temps (meilleure reproductibilité).

D.1.1. B. Avantages et caractéristiques

les caractéristiques des thermomètres à résistance métallique sont:

- bonnes reproductibilité, stabilité et interchangeabilité ;
- imprécision pouvant atteindre $\pm 0,2\%$ de l'étendue d'échelle;
- temps de réponse de 0,5 à 5 secondes ou plus ;
- étendues de mesure de -200°C à 700°C ;
- leurs éléments de mesure peuvent utiliser le courant continu ou le courant alternatif.

Ils ne nécessitent pas de compensation complexe comme c'est le cas pour la température de la soudure froide ou pour les courants vagabonds (du bruit de fond) dans les thermocouples.

D.1.1. C. Inconvénients

- Les RTD sont des appareils passifs: une alimentation électrique extérieure d'excitation est indispensable à leur fonctionnement;
 - ils sont influençables par l'humidité et les gaz corrosifs ;
 - réchauffement de leur résistance (auto-échauffement) par le passage du courant d'excitation et qui s'ajoute à réchauffement par le milieu à mesurer, nécessite une compensation délicate.

D.1.2. Thermomètre à thermistance

D.1.2. A. Le capteur

L'élément sensible de ces thermomètres est une thermistance faite d'un matériau semi-conducteur dont la résistivité diminue très rapidement quand la température augmente, c'est-à-dire dont le coefficient de température est négatif (CTN) ; l'élément sensible passe donc, à température élevée, d'isolant à conducteur. De plus, ce coefficient CTN est environ dix fois plus élevé en valeur absolue que celui des conducteurs. On notera cependant que la variation de la résistance avec la température n'est pas linéaire; la relation est logarithmique.

Les thermistances sont généralement constituées d'agglomérats compacts d'oxydes ou de sels de Mn, Ni, Co, Fe. etc.

Elles peuvent se présenter sous forme de perles, de disques, de cylindres ou de rondelles, de volume réduit et de diamètre inférieur à 1 ou 2 mm pour les perles, par exemple.

La résistance initiale ou nominale des thermistances est variable selon leur volume et leur composition. Sa valeur peut être, par exemple, de 50 k Ω à 20 °C et de 80 Ω à 500°C.

D.1.2. B. L'élément de mesure

La mesure de résistance s'effectue là aussi le plus souvent par la méthode du pont de Wheatstone. Cependant, vu leur impédance élevée. L'influence de la température sur les connexions est moins importante que dans les thermomètres à résistance métallique.

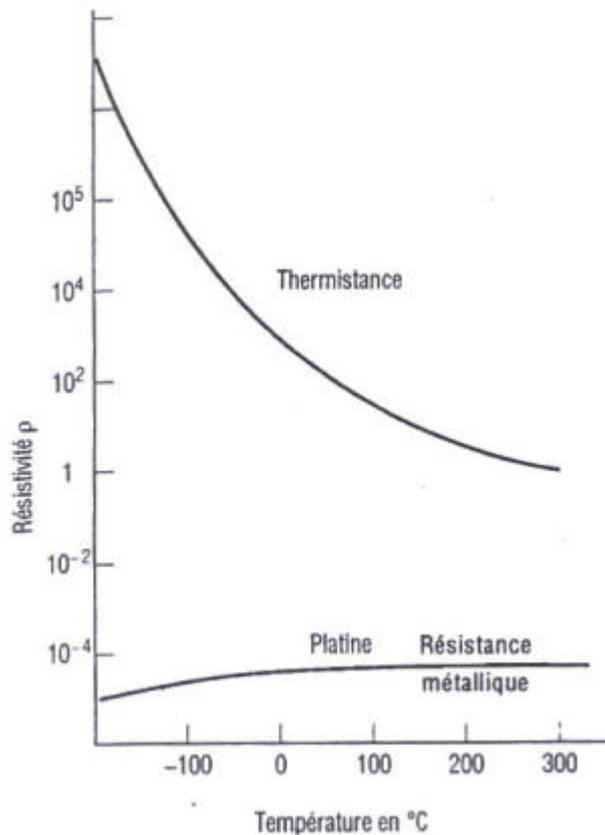


Figure C.1.
Résistivité vs température

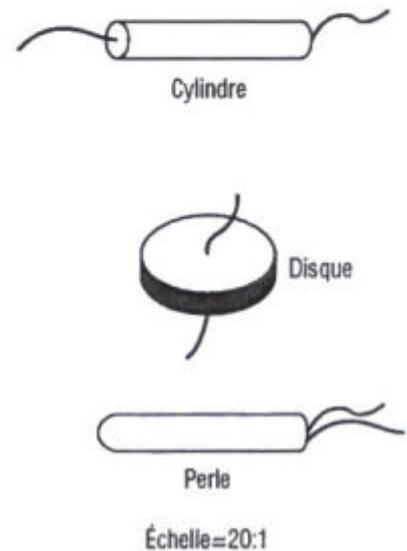


Figure C.2.
Forme des thermistances

D.1.2.C. Avantages et caractéristiques

- Les thermomètres à thermistance ont une étendue d'utilisation comprise entre $-110\text{ }^{\circ}\text{C}$ et $250\text{ }^{\circ}\text{C}$. Elle peut cependant atteindre $700\text{ }^{\circ}\text{C}$;
- leur volume très réduit rend leur temps de réponse relativement court;
- ce sont les plus sensibles de tous les thermomètres électriques; leur seuil de mobilité peut être inférieur à $0,01\text{ }^{\circ}\text{C}$;
- leur imprécision est inférieure à $\pm 0,1\text{ }^{\circ}\text{C}$;
- leur stabilité augmente avec le temps ; 90 % du vieillissement se produit dans la première semaine d'utilisation;
- ils sont adaptables pour la mesure des températures des surfaces solides avec ou sans contact.

Inconvénients

- Echelle non linéaire, étendue réduite ;
- Interchangeabilité compliquée ;
- Mauvaise stabilité au-dessus de 300 °C;
- Reproductibilité moyenne;
- Auto-échauffement élevé.

Remarque

Lorsque le capteur est constitué de semi-conducteurs de type germanium ou silicium, le thermomètre s'appelle alors thermomètre à semi-conducteur au lieu de thermistance.

D.1.2.E. Comparaison entre thermocouples et thermomètres résistifs

- Les thermocouples permettent de mesurer les températures les plus élevées ;
- les thermomètres à résistance mesurent les plus basses températures ;
- les thermomètres à thermistance sont les plus petits et les plus sensibles ;
- tous les thermomètres électriques se prêtent bien à la transmission électrique des mesures ;
- tous conviennent à la mesure de la température des solides (par exemple les parois de conduites ou de réacteurs) et à la mesure de la différence entre deux températures.

D.1.3. Thermomètre à thermocouple

D.1.3. A. Élément primaire - capteur

Les critères de choix d'un couple thermique, sont notamment:

- le point de fusion et la résistance à l'atmosphère ambiante ;
- la stabilité, la sensibilité et la reproductibilité ;
- interchangeabilité.

Une gaine protectrice est le plus souvent nécessaire pour protéger la soudure chaude; l'ensemble (SC + gaine) s'appelle canne pyrométrique. La gaine est en général métallique. L'espace entre les conducteurs et la gaine doit être rempli de perles réfractaires qui limitent

réchauffement de la tête de raccordement par convection. La tête isolante de la gaine porte les bornes de liaison des extensions. Le diamètre de la canne est compris entre 1 et 20 mm et sa longueur est de quelques dizaines de cm.

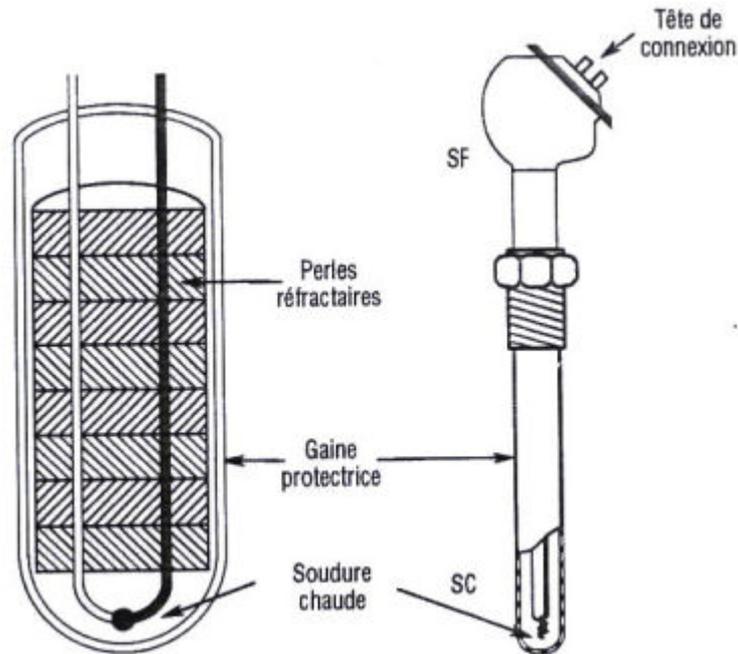


Figure C.3. Gaine protectrice du thermocouple

D.1.3. B. Avantages

- Imprécision de Tordre de $\pm 0,2 \%$ de l'étendue d'échelle depuis le voisinage du zéro absolu jusqu'à 2700°C ;
- utilisation à certaines distances sans besoin d'énergie extérieure;
- la bonne reproductibilité des mesures ;
- le fait que la longueur et le diamètre des câbles du circuit de la f.é.m. importent peu (tant que ces câbles sont homogènes);
- leur robustesse ;
- l'absence de fluide dans leur conception ;
- leur volume réduit qui permet un temps de réponse relativement court ;
- la possibilité de les souder aux surfaces solides;
- leur prix modéré ;

- l'interchangeabilité des couples ne nécessite pas un réétalonnage de l'appareil de mesure de la f.é.m.

D.1.3. C. Inconvénients

- _ La complexité de compensation ;
- l'impossibilité d'utilisation sans gaine dans les liquides conducteurs;
- la faible f.é.m. développée et l'effet du bruit de fond électrique dans une ambiance industrielle, facteur délicat à corriger.

D.2. Caractéristique technique de la sonde de température à thermistance de marque TB /TS :

D.2. 1. Description Caractéristiques

Le Capteur de Température Ambiante Thermistance TB/TS, à montage mural, est conçu pour une réponse thermique optimale, il possède un boîtier de faible épaisseur. Plusieurs options sont disponibles, comprenant un bouton au profil peu saillant pour le réglage d' un potentiomètre (ex. réglage du point de consigne), un bouton poussoir (dérogation occupation), deux LED (indication sur l' état d'occupation), et un switch de régulation 5 positions pour ventilateur. Ces options sont disponibles dans plusieurs configurations.



Figure C.4.Sonde TB /TS

D.2.2. Raccordement : le capteur est branché dans l'automate comme suite

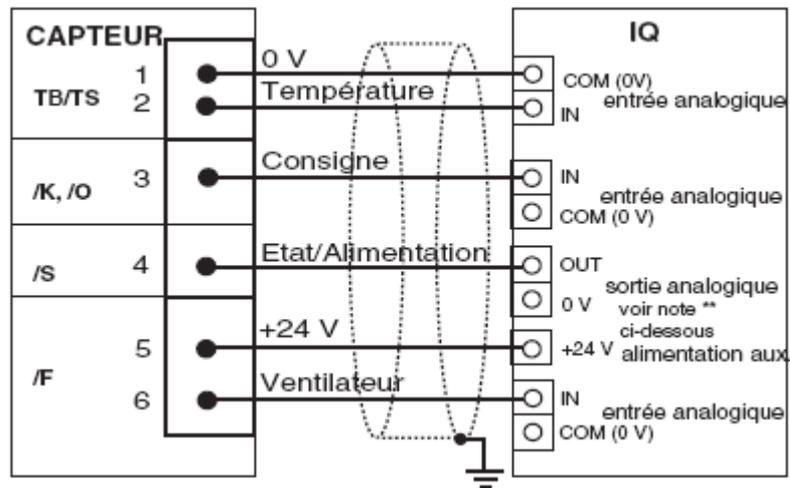


Figure C.5. Raccordement

SPÉCIFICATIONS

Thermistance : 10 k Ω à 25 °C

Plage de température : 0 à +100 °C (conseillée)

Précision de température : du capteur, ± 0.44 °C (0 à +100 °C)

Potentiomètre : 1 k Ω à 11 k Ω ± 20 %.

Dérogation

/O : Durée d'impulsion 7 secondes (entrée consigne circuit ouvert).

/E : contact libre de potentiel

LED d'état : Occupation LED verte 5,5 V à 10 V,

Non-Occupation LED jaune 4,5 V à 5,0 V

Sélecteur de Vitesse Ventilateur

pour l'IQ : 5 niveaux de tension sélection parswitch (0 V à 9,7 V)

pour l'IQL : 5 niveaux de résistance sélection parswitch 4,7k Ω à 17,9k Ω

(TB/TS/KEF uniquement).