

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole nationale polytechnique



Département d'Automatique  
Société des Eaux et d'Assainissement d'Alger

Mémoire de projet de fin d'études  
Pour l'obtention du diplôme d'Ingénieur d'état en Automatique

# Automatisation et télégestion d'une station de pompage

**Kadem Chérifa**

**Ouail Isma**

Sous la direction de :

**PR : E.M.BERKOUK**

Soutenu publiquement le 20 /06/2017

Membres du jury:

<b>Président : Mr. H. CHEKIREB,</b>	<b>Professeur</b>	<b>Ecole Nationale Polytechnique</b>
<b>Rapporteur : Mr. M. BERKOUK,</b>	<b>Professeur</b>	<b>Ecole Nationale Polytechnique</b>
<b>Examineurs : Mr. HEMICI,</b>	<b>Professeur</b>	<b>Ecole Nationale Polytechnique</b>

**Enp 2017**



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole nationale polytechnique



Département d'Automatique  
Société des Eaux et d'Assainissement d'Alger

Mémoire de projet de fin d'études  
Pour l'obtention du diplôme d'Ingénieur d'état en Automatique

# Automatisation et télégestion d'une station de pompage

**Kadem Chérifa**

**Ouail Isma**

Sous la direction de :

**PR : E.M.BERKOUK**

Soutenu publiquement le 20 /06/2017

Membres du jury:

<b>Président : Mr. H. CHEKIREB,</b>	<b>Professeur</b>	<b>Ecole Nationale Polytechnique</b>
<b>Rapporteur : Mr. M. BERKOUK,</b>	<b>Professeur</b>	<b>Ecole Nationale Polytechnique</b>
<b>Examineurs : Mr. HEMICI,</b>	<b>Professeur</b>	<b>Ecole Nationale Polytechnique</b>

**Enp 2017**

## ملخص:

يستند العمل المقدم في هذه المذكرة على استخدام المبرمج الآلي SIEMENS S7 1200 من أجل التحكم في محطة لضخ المياه. هذا العمل هو التحكم والإشراف على المحطة باستعمال برنامج TIA Portal وضمان اتصالها مع غرفة المراقبة من أجل المراقبة والتحكم عن بعد باستخدام برنامج TOPKAPI وتقنية الجيل الثالث لاتصال.

**الكلمات المفتاحية:** المبرمج الآلي، TIA Portal، SIEMENS، محطة ضخ المياه، WinCC، تحكم عن بعد،

TOPKAPI

## Abstract:

The work presented in this project is essentially based on the utilization of SIEMENS S7-1200 Programmable Logic Controllers. Our work is the control and supervision of this station using the programming software TIA Portal. Remote management and a distant supervision using TOPKAPI software.

**Keywords:** PLC, TIA Portal, SIEMENS, HMI, pumping station, WinCC, remote management, TOPKAPI.

## Résumé :

Le travail présenté dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables SIEMENS S7-1200 pour l'automatisation d'une station de pompage. Ce travail est la commande et la supervision de cette dernière par l'utilisation du logiciel de programmation TIA Portal. Ainsi que la télégestion et la supervision à distance à l'aide du logiciel TOPKAPI.

**Mots clés :** API, TIA Portal, SIEMENS, HMI, station de pompage, WinCC, télégestion, TOPKAPI.

## *Dédicaces*

*Je dédie ce modeste travail à mes chers parents, tous les mots du monde ne sauraient exprimer l'immense amour que je vous porte, ni la profonde gratitude que je vous témoigne pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon éducation et mes études.*

*A ma sœur Lamia et mon frère Mohamed pour qui j'ai toujours tenu à donner le meilleur de moi-même*

*A mon cher frère Fouad qui m'a transmis l'amour du savoir et m'a tant conseillé et soutenu.*

*A ma précieuse petite sœur Rayane qui étais toujours présente pour moi.*

*A mes adorables nièces et mon cher neveu ainsi que mes belles sœurs.*

*A mon binôme ISMA pour ces cinq années de travail, de complicité et de souvenirs.*

*A mon ami Abdelkader qui m'a tant apporté son soutien et encouragement qu'il trouve ici toute ma gratitude*

*Je dédie ce mémoire à : Amira, Amina, Fella, Hadjer, Lydia et à tous mes camarades de l'école polytechnique avec qui j'ai eu d'agréables moments.*

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

*Chérifa*

## ***Dédicaces***

*C'est avec un énorme plaisir et une immense joie que je dédie ce modeste travail à*

*Ma mère, qui a œuvré pour ma réussite, par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.*

*Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien venu de toi.*

*Mes deux frères Mohamed et Nabil et à ma grande mère que je ne remercierai jamais assez.*

*Chérifa pour la complicité et pour tous les moments qu'on a passé ensemble.*

*Redha pour le soutien infailible, les conseils et l'encouragement. Que ce travail soit le témoignage de ma reconnaissance et gratitude. Merci d'être toujours là pour moi.*

*Tous mes amis en particulier Anfal, Sabrina, Amel, ainsi que ma chère cousine Sabrina.*

*Isma*

## **Remerciement :**

*Nous rendons grâce à Dieu le Tout puissant et le Miséricordieux de nous avoir donné la force morale, physique et l'aide pour accomplir ce modeste travail.*

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

*Nous remercions également Monsieur O.STIHI et Monsieur M.BOUCHERIT pour leurs conseils et aide.*

*Nous remercions chaleureusement membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions*

*Nous présentons également nos remerciements à notre encadreur Mr M.AMMOUR tout en lui témoignant notre gratitude pour ces précieux conseils.*

*Nous tenant aussi à remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, en particulier, nos professeurs d'Automatique qui nous ont encadrées pour les connaissances qu'ils nous ont transmis, et leurs efforts qu'ils ont fourni*

*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 en particulier Aflah ELOUNEG et Redha LEMMOUI.*

# ***Table des matières***

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale.....	15
Plan de travail.....	16

## **Chapitre I : Automates programmables industriels**

1. Introduction .....	18
2. Historique .....	18
3. Définition .....	18
4. Rôles de l'API dans un système de production .....	19
5. Architecture des automates : .....	20
5.1. Aspect extérieur : .....	20
5.2. Structure interne : .....	21
6. Principales fonctions .....	22
7. Fonctionnement d'un A.P.I .....	23
8. Les langages de programmation .....	24
9. Présentation de la gamme SIMATIC de SIEMENS .....	25
9.1. SIMATIC contrôleurs modulaires .....	24
9.2. Caractéristiques de la CPU 1214C DC/DC/RELAY (6ES7214-1HG40-0XB0) .....	29
9.3. Caractéristiques des modules d'entrées TOR SM 1221 .....	32
9.4. Caractéristiques des modules de sorties TOR SM 1222 .....	32
10. Conclusion.....	33

## **Chapitre II: TIA Portal**

1. Introduction : .....	35
2. Vue du portail et vue du projet : .....	35
2.1. Vue du portail : .....	35
2.2. Vue du projet : .....	36
3. Création d'un projet : .....	37
3.1. identification du matériel : .....	38
3.2. Configuration et paramétrage du matériel : .....	39
3.3. Les variables API : .....	40
3.3.1. Adresses symbolique et absolue : .....	40



3.3.2. Adressage des E/S :	41
3.4. Table des variables API	42
3.5. Tableau d'affectation :	43
3.6. Les références croisées	44
4. Les blocs dans TIA Portal:	44
4.1. Les blocs d'organisation - OB :	44
4.2. Les fonctions – FC :	44
4.3. Les blocs fonctionnels – FB :	44
4.4. Les blocs de données – DB :	44
4.5. Création et programmation d'un bloc :	45
4.6. Appel d'un bloc depuis l'OB1 :	46
5. Les langages de programmation dans TIA Portal:	46
6. Compilation et chargement :	46
7. La simulation avec PLCSIM :	47
8. Visualisation du processus :	47
8.1. Présentation du WinCC sous TIA Portal :	47
8.2. Définition de l'interface homme-machine (HMI) :	47
8.3. Configuration du HMI dans le TIA portal:	47
8.4. Eléments de l'interface utilisateur de WinCC :	49
8.5. Définition des variables :	50
8.6. Les alarmes dans WinCC :	51
8.6.1. Classes d'alarmes :	52
8.6.2. Types des alarmes :	53
8.6.3. Configuration d'alarme :	54
8.6.4. Etats des alarmes :	56
8.7. Les recettes :	56
8.7.1. Editeur de recette :	56
8.8. Les archives :	57
8.8.1. Archives d'alarmes :	57
8.8.2. Archives de variables :	58
8.9. Le test de la visualisation du processus :	59
9. Conclusion	60
<b>Chapitre III: Communication et télégestion</b>	
1. Introduction :	62
2. Quelques définitions :	62
3. PROFINET :	63

3.1.	Définition :	64
3.2.	Historique	65
3.3.	PROFINET et le modèle OSI :	65
3.4.	Caractéristiques de la couche physique :	66
3.5.	Le PROFINET IO et le PROFINET CBA :	68
3.6.	Définition PROFINET IO :	69
3.6.1.	Constituants d'un réseau PROFINET IO :	70
3.6.2.	Topologies de PROFINET IO :	70
3.6.3.	Echange des données avec le PROFINET IO :	71
3.6.4.	Priorisation des trames PROFINET :	71
4.	MODBUS	72
4.1.	Définition :	72
4.2.	Structure des messages dans le MODBUS :	73
4.3.	Mode de fonctionnement de MODBUS :	74
5.	La télégestion	75
6.	Logiciel de supervision et de télégestion Topkapi	76
6.1.	Synoptiques graphique :	77
6.2.	Les courbes	78
6.3.	Alarmes	79
6.4.	Télégestion dans TOPKAPI	80
7.	Conclusion	81

#### **Chapitre IV: Etude descriptive de la station SP5**

1.	Introduction	72
2.	Description de la station de pompage SP5	72
2.1.	Description générale	72
2.2.	Partie électrique	73
2.2.1.	Equipement Electrique MT	73
2.2.2.	Equipement Electrique BT	74
2.3.	Instrumentation	77
2.3.1.	Actionneurs	77
2.3.2.	Capteurs	81
3.	Cahier des charges de la station	84
4.	Conclusion	86

#### **Chapitre V: Programmation et télégestion**

1. Introduction.....	88
2. Objectif du travail .....	88
3. Configuration matérielle .....	89
4. Développement du programme.....	89
4.1. Description des FB .....	89
4.2. Description des FC .....	100
4.3. Description des DB.....	101
5. Développement de la supervision .....	102
5.1. Création de la station HMI .....	102
5.2. Création des alarmes.....	102
5.3. Création des vues.....	103
6. Développement de l'application télégestion avec TOPKAPI.....	107
6.1. Configuration de protocole de communication : .....	108
6.2. Configuration de l'automate.....	108
6.3. Paramétrages des données et remplissage de la fiche tableur .....	109
7. Réalisation matérielle.....	111
7.1. Présentation du montage.....	111
7.2. Implémentation du programme dans le matériel .....	114
7.3. Le test de la télégestion .....	116
8. Conclusion .....	117
Conclusion générale.....	118
Bibliographie.....	119
Annexe A.....	120
Annexe B.....	122
Annexe C.....	125
Annexe D.....	129
Annexe E.....	134

## Liste des tableaux :

<b>I-1</b> Caractéristiques de la CPU. ....	28
<b>I-2</b> Caractéristiques des modules d'entrées TOR SM 1221. ....	31
<b>I-3</b> Caractéristiques des modules des sorties TOR SM 1222. ....	31
<b>IV-1</b> Caractéristiques des GEP de la ligne DEKAKNA. ....	79
<b>IV-2</b> Caractéristiques des GEP de la ligne HADJ YAKOUB. ....	79
<b>IV-3</b> Caractéristiques des démarreurs progressifs. ....	80
<b>V-1</b> Les entrées-sorties physiques pour chaque pompe. ....	97
<b>V-2</b> Caractéristiques de l'HMI. ....	102
<b>V-3</b> Affiliation de la vue synoptique. ....	104
<b>V-4</b> Affiliation de la vue de pompe. ....	105
<b>V-5</b> Affiliation de la vue défaut électrique. ....	106

## ***Liste des figures :***

<b>I-1</b> Automate en commande. ....	20
<b>I-2</b> Automate compact (Allen-Bradley). ....	20
<b>I-3</b> Automate modulaire (Siemens). ....	21
<b>I-4</b> structure interne d'un API. ....	21
<b>I-5</b> Cycle d'un API. ....	24
<b>I-6</b> CPU 1214C DC/DC/RELAY. ....	28
<b>II-1</b> Vue de portail. ....	35
<b>II-2</b> Vue projet. ....	36
<b>II-3</b> Vue création de projet. ....	37
<b>II-4</b> Identification d'un module. ....	37
<b>II-5</b> Identification d'une CPU. ....	38
<b>II-6</b> Onglet d'information. ....	38
<b>II-7</b> Exemple d'adresse symbolique et physique. ....	39
<b>II- 8</b> Table de variable. ....	40
<b>II-9</b> Table de variable. ....	41
<b>II-10</b> Table d'affectation. ....	41
<b>II-11</b> Vue des blocs disponibles. ....	43
<b>II-12</b> Programme d'un bloc. ....	43
<b>II-13</b> Appel d'un bloc dans OB1. ....	44
<b>II-14</b> Les étapes de compilation. ....	44
<b>II-15</b> Etat de la CPU. ....	45
<b>II-16</b> Vue PLCSIM. ....	45
<b>II-17</b> Schéma explicatif. ....	46
<b>II-18</b> Vue configuration HMI. ....	47
<b>II-19</b> Assistant HMI. ....	48
<b>II-20</b> Vue d'interface WinCC. ....	49
<b>II-21</b> Vue des alarmes HMI. ....	50
<b>II-22</b> Vue classe d'alarme. ....	51
<b>II-23</b> Vue configuration d'alarme de bit. ....	52
<b>II-24</b> Vue configuration de recette. ....	54
<b>II-25</b> Vue configuration d'archive. ....	56
<b>II-26</b> Lancement du RUNTIME. ....	57

<b>III-1</b> PROFINET et le modèle OSI. ....	62
<b>III-2</b> Les 3 stations de PROFINET. ....	63
<b>III-3</b> Priorisation PROFINET. ....	65
<b>III-4</b> Structure du réseau MODBUS. ....	66
<b>III-5</b> Structure d'un message dans le protocole MODBUS. ....	66
<b>III-6</b> Bibliothèque de TOPKAPI. ....	69
<b>III-7</b> Vue des courbes dans TOPKAPI. ....	69
<b>III-8</b> Vue des alarmes dans TOPKAPI. ....	70
<b>IV-1</b> Schéma représentatif de la station SP5. ....	73
<b>IV-2</b> GEP de refoulement ligne HADJ YAKOUB. ....	73
<b>IV-3</b> GEP de refoulement ligne DEKAKNA. ....	73
<b>IV-4</b> La bache d'aspiration. ....	74
<b>IV-5</b> Transformateur abaisseur. ....	74
<b>IV-6</b> Vue de l'armoire TGBT. ....	75
<b>IV-7</b> Armoire départ arrivé .....	76
<b>IV-8</b> Armoire couloir HADJ YAKOUB et DEKAKNA. ....	76
<b>IV-9</b> Vue extérieure et intérieure de l'armoire d'automatisme. ....	77
<b>IV-10</b> Armoire batterie de compensation. ....	77
<b>IV-11</b> Groupe électrogène.....	78
<b>IV-12</b> Anti-bélier ligne HADJ YAKOUB. ....	80
<b>IV-13</b> débitmètre électromagnétique et son convertisseur. ....	81
<b>IV-14</b> sonde de pression. ....	84
<b>V-1</b> Configuration du matériel. ....	89
<b>V-2</b> Schéma descriptif du démarrage-arrêt de la ligne DEKAKNA. ....	98
<b>V-3</b> Schéma descriptif du démarrage-arrêt de la ligne HADJ YAKOUB. ....	99
<b>V-4</b> Tableau de création des alarmes. ....	103
<b>V-5</b> Synoptique de la station SP5. ....	103
<b>V-6</b> Vue des pompes. ....	104
<b>V-7</b> Vue défauts électriques générales. ....	105
<b>V-8</b> Vue des alarmes .....	106
<b>V-9</b> Vue des alarmes tampons. ....	107

<b>V-10</b> Vue caractéristiques pompe et moteur. ....	107
<b>V-11</b> Fenêtre principale du logiciel TOPKAPI. ....	108
<b>V-12</b> Configuration du port. ....	108
<b>V-13</b> Configuration de la station automate. ....	109
<b>V-14</b> Configuration des données TOR. ....	109
<b>V-15</b> Configuration des données alarmes. ....	110
<b>V-16</b> Configuration des données télécommande. ....	111
<b>V-17</b> La fiche tableur SP5 après remplissage. ....	111
<b>V-18</b> la maquette d'armoire d'automatisme. ....	112
<b>V-19</b> Les abonnés accessibles. ....	113
<b>V-20</b> configuration des adresses IP.....	114
<b>V-21</b> Chargement du programme dans l'API. ....	114
<b>V-22</b> Chargement du programme dans l'HMI. ....	115
<b>V-23</b> Communication 3G entre la station et la supervision. ....	115
<b>V-24</b> La fiche tableur en mode exploitation. ....	116
<b>V-25</b> Fenêtre des alarmes et d'archives. ....	116

## Liste des abréviations

<b>API</b>	Automates programmables industriels
<b>BT</b>	Basse tension
<b>CONT</b>	Langage à base de contacte
<b>CPU</b>	Central Processing Unit
<b>FB</b>	Bloc de fonction
<b>FBD</b>	Function Block Diagram
<b>FC</b>	Fonction
<b>HMI</b>	Humain Machine Interface
<b>IL</b>	Instruction List
<b>IP</b>	Internet Protocol
<b>IP</b>	Indice de protection
<b>LD</b>	Ladder Diagram
<b>LOG</b>	Langage à base de logigrammes
<b>MT</b>	Moyenne tension
<b>OB</b>	Organisationnel Bloc
<b>OSI</b>	Open Systems Interconnection
<b>SCL</b>	Structured Control Language
<b>SEAAL</b>	Société des Eaux et d'Assainissement d'Alger
<b>SFC</b>	Sequential Function Chart
<b>SP5</b>	Station de pompage 5
<b>ST</b>	Structured Text
<b>TIA Portal</b>	Totally Integrated Automation Portal
<b>TOR</b>	Tout Ou Rien



## **Introduction générale :**

Dans le cadre du plan de développement de l'hydraulique et afin de satisfaire la demande croissante de la population en eau potable et de garantir un approvisionnement régulier en adéquation avec la croissance économique, les entreprises du secteur font recours à de nouvelles technologies de production par l'automatisation du process de la production et de la distribution.

L'entreprise SEAAL consciente de l'importance de ce projet, lance un programme d'automatisation et de supervision de ses stations en utilisant des automates programmables dans le but d'améliorer leurs production et garantir une meilleure desserte en eau potable.

C'est dans cette conjoncture que la direction de SEAAL à KOUBA, nous a proposé de mener une étude visant à développer une solution à base d'automates programmables SIEMENS S7-1200 afin d'automatiser une station de pompage et développer une solution pour la supervision et la commande à distance de la station de pompage SP5 qui se situe à TESSALA EL MERDJA. Notre travail consiste en premier lieu à suivre un cahier de charge spécifique décrit par l'entreprise et élaborer un programme permettant un fonctionnement automatique sécurisé ainsi qu'une supervision de la station de pompage à l'aide du logiciel TIA Portal, puis établir une communication à distance avec le logiciel TOPKAPI. Finalement réaliser un prototype de l'armoire d'automatisme de la station.

## **Plan du travail**

Le travail est présenté en cinq chapitres complémentaires :

Dans le chapitre I on a mené une étude sur les automates programmables industriels en décrivant leur architecture, fonctionnement, rôle et langages de programmation.

Après une brève description de la gamme SIMATIC de SIEMENS, on a fait une étude détaillée sur l'automate S7-1200, les caractéristiques de sa CPU et les modules d'entrées et de sorties utilisés dans notre projet.

Le deuxième chapitre est une description de l'environnement TIA Portal, un logiciel qui intègre le STEP7 et le WINCC, il est dédié à la programmation des automates SIEMENS, la simulation des programmes, la création et la configuration des interfaces HMI nécessaires à la supervision, le diagnostic et la visualisation du process.

Le chapitre III présente au début le type de communications filaire de l'automate avec son environnement industriel, tels que les réseaux industriels PROFINET et le MODBUS. En suite une présentation de la communication sans fils a été faite suivi d'une introduction au logiciel TOPKAPI.

Le chapitre IV est une étude descriptive de la station de pompage à automatiser, dans lequel on a décrit la partie électrique MT et BT, l'instrumentation ainsi que le cahier des charges à suivre dans notre projet.

Le dernier chapitre sert à présenter l'objectif du travail à réaliser, les blocs du programme édité ainsi que les vues HMI créées à l'aide du logiciel TIA Portal, le chargement du programme dans le matériel de la maquette réalisée. Finalement la configuration de l'interface de supervision à distance et le paramétrage des routeurs pour la communication 3G de l'automate avec une supervision à distance en utilisant le logiciel TOPKAPI.

**Chapitre I :**  
**Automates**  
**programmables**  
**industriels**

## 1. Introduction

Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour cela que l'API s'est substitué aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que les coûts de câblage et de maintenance devenaient trop élevés.

Dans ce chapitre, on présente architecture et fonctionnement des automates programmables, par la suite on aborde brièvement la gamme SIMATIC de Siemens, et plus précisément l'automate S7-1200, CPU et modules que nous avons choisi pour la réalisation de notre projet.

## 2. Historique

L'automate programmable, souvent appelé automate programmable industriel (API, en anglais PLC pour *Programmable logic controller*) pour rappeler son domaine privilégié d'utilisation, l'industrie, est apparu voici 40 ans et s'est rapidement répandu dans la production, la logistique, le conditionnement, la gestion technique de bâtiments, etc. Son développement a accompagné celui de l'automatisation de la production, la faisant passer du stade de la machine automatisée à celui du système automatisé de production (SAP), et il en est devenu, avec le robot, un composant majeur, le « fantassin de l'automatisation industrielle », suivant l'expression de C. Laurgeau, l'un des auteurs du premier ouvrage français sur la question. Il s'est vendu à des millions d'exemplaires, et seul le PC, né plus tard, avec un champ d'application plus large, a fait mieux.

La question de le ramener à cet outil de communication universel qu'est le PC s'est alors rapidement posé. Lorsque l'on cerne bien la définition rappelée ci-après et le domaine concerné, l'industrie et les services gérant du matériel, que l'on prend en compte l'évolution des moyens de transmission de l'information, ces systèmes à processeur ne s'emploient pas de manière optimale au même niveau. Ils sont complémentaires dans une optique de traitement numérique de plus en plus poussé, intégrant toutes les étapes du processus de production [1].

## 3. Définition

La définition donnée aux Automates programmables par la norme française NF 61131 dans sa partie 1 est la suivante : « *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 ».*

A partir de là, et en respectant les caractéristiques essentielles qui ressortent de cette définition souvent un peu laborieuses, nous considérerons plus simplement comme automate programmable un système :

- construit autour d'un processeur numérique, spécifique ou non.
- pouvant être relié à de nombreux signaux physiques.
- fonctionnant dans des conditions industrielles grâce à une protection adaptée ;
- doté d'un logiciel de programmation permettant un traitement rapide des fonctions logiques (TOR).
- doté de possibilités d'échanges avec d'autres processeurs.

Ceci constitue un « noyau » minimal susceptible de nombreuses variantes et extensions propres à un automate donné [1].

#### **4. Rôles de l'API dans un système de production**

L'API est donc d'abord un composant de commande, envoyant des signaux vers les actionneurs, en fonction des informations reçues de l'instrumentation du système matériel, la Partie Opérative, et des ordres reçus (consignes), selon une algorithmique appropriée définie par le programme. La figure 1 explicite ce fonctionnement ; la Partie Commande –l'API– agit dans le cadre d'un système bouclé sur les éléments matériels de l'installation, en vue d'une production, au sens large du terme : fabrication bien sûr, mais aussi emballage, autorisation d'accès, etc. De plus en plus, elle gère aussi l'énergie, incluant donc celle-ci, s'il s'agit d'élaborer matières premières ou objets, dans une Partie Opérative élargie [1].

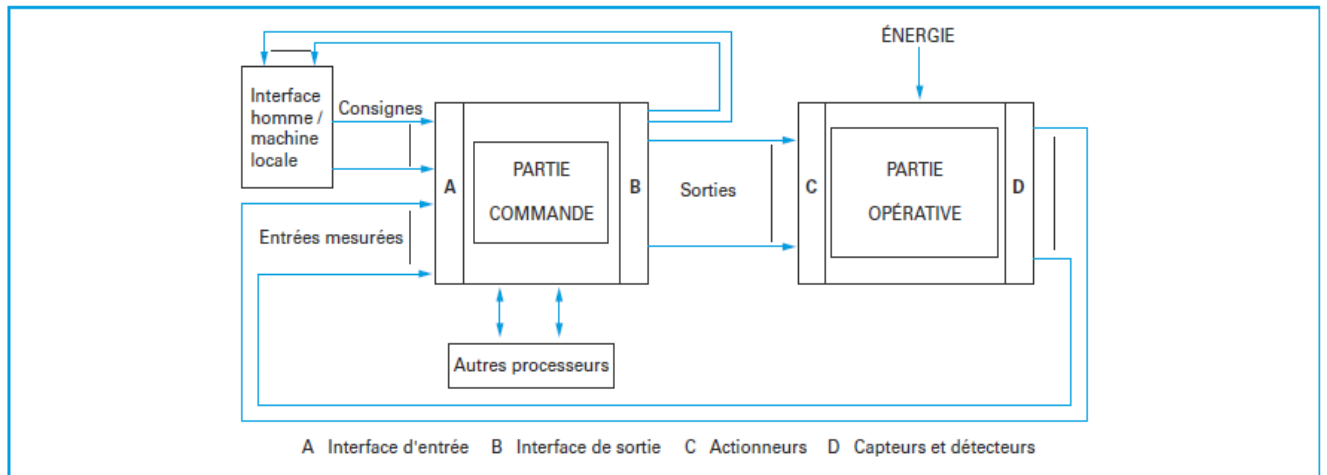


Figure I -1 Automate en commande.

## 5. Architecture des automates

### 5.1. Aspect extérieur

Les automates peuvent être de type **compact** ou **modulaire**.

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

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

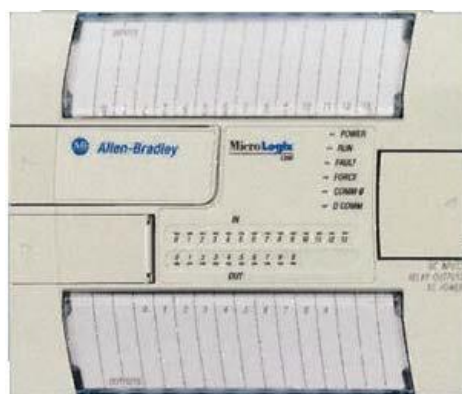


Figure I -2 Automate compact (Allen-Bradley).

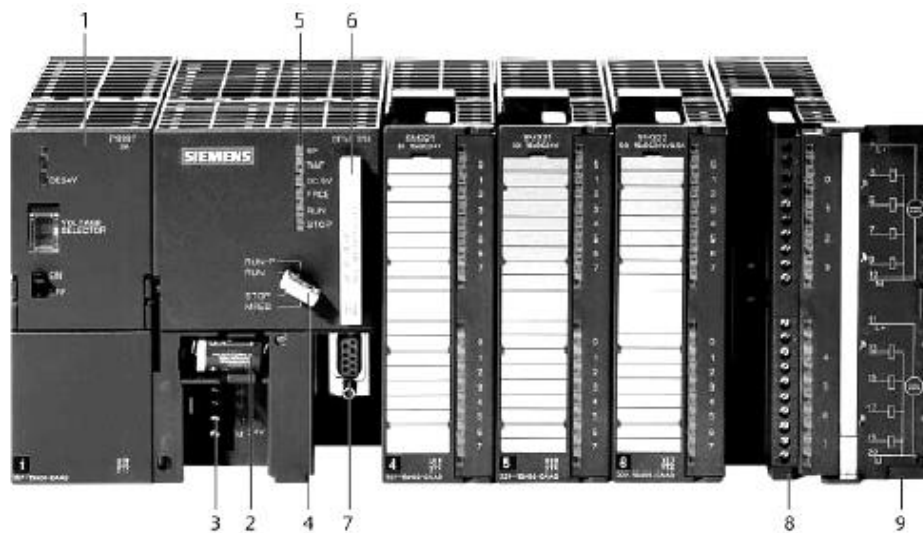


Figure I-3 Automate modulaire (Siemens).

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

### 5.2. Structure interne

La figure suivante représente les différents constituants de la structure interne d'un API [2] :

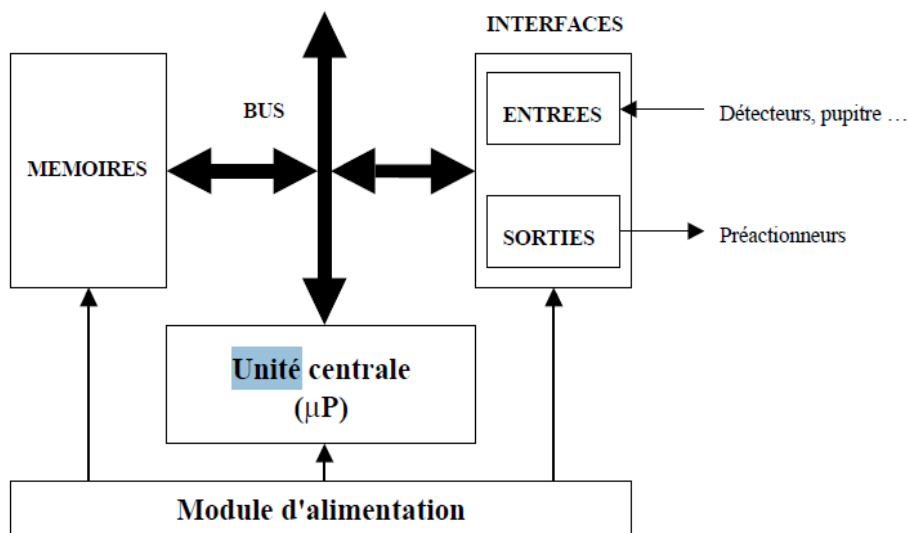


Figure I-4 Structure interne d'un API.

- **Module d'alimentation** : il assure la distribution d'énergie aux différents modules.
- **Unité centrale** : à base de microprocesseur, elle réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...).
- **Le bus interne** : il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.
- **Mémoires** : Elles permettent de stocker le système d'exploitation (ROM ou PROM), le programme (EEPROM) et les données système lors du fonctionnement (RAM). Cette dernière est généralement secourue par pile ou batterie.
- **Interfaces d'entrées / sorties** :
  - \* **Interface d'entrée** : elle permet de recevoir les informations du S.A.P. ou du pupitre et de mettre en forme (filtrage, ...) ce signal tout en l'isolant électriquement (optocouplage).
  - \* **Interface de sortie** : elle permet de commander les divers prés actionneurs et éléments de signalisation du S.A.P. tout en assurant l'isolement électrique.

## 6. Principales fonctions

Chaque composant de l'API a une fonction bien spécifique et il est fait pour accomplir certaines tâches, dans ce qui suit les principales fonctions de différentes cartes de l'API [2]:

- **Cartes d'entrées / sorties** : Au nombre de 4, 8, 16 ou 32, elles peuvent aussi bien réaliser des fonctions d'entrées, de sorties ou les deux. Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...).

Les voies peuvent être indépendantes ou posséder des "communs".

Les cartes d'entrées permettent de recueillir l'information des capteurs, boutons ... qui lui sont raccordés et de la matérialiser par un bit image de l'état du capteur.

Les cartes de sorties offrent deux types de technologies : les sorties à relais électromagnétiques (bobine plus contact) et les sorties statiques (à base de transistors ou de triacs).

- **Cartes de comptage rapide** : elles permettent d'acquérir des informations de fréquences élevées incompatibles avec le temps de traitement de l'automate par exemple : signal issu d'un codeur de position.

- **Cartes de commande d'axe** : Elles permettent d'assurer le positionnement avec précision d'élément mécanique selon un ou plusieurs axes. La carte permet par exemple de piloter un servomoteur et de recevoir les informations de positionnement par un codeur. L'asservissement de position pouvant être réalisé en boucle fermée.



- **Cartes d'entrées / sorties analogiques** : Elles permettent de réaliser l'acquisition d'un signal analogique et sa conversion numérique (CAN) indispensable pour assurer un traitement par le microprocesseur. La fonction inverse (sortie analogique) est également réalisée.

Les grandeurs analogiques sont normalisées : 0-10V ou 4-20mA.

- **Autres cartes** :

- \* Cartes de régulation PID.
- \* Cartes de pesage.
- \* Cartes de communication (Ethernet ...).
- \* Cartes d'entrées / sorties déportées.

## 7. Fonctionnement d'un A.P.I

Lorsque l'API est en fonctionnement, trois phases se succèdent :

- **PHASE 1 : LECTURE DES ENTREES (%I)**

Durant cette phase qui dure quelques microsecondes : les entrées sont « photographiées » et leurs états logiques sont stockés dans une zone spécifique de la mémoire de donnée. Le programme n'est pas scruté. Les sorties ne sont pas mises à jour.

- **PHASE 2 : TRAITEMENT DU PROGRAMME (T)**

Durant cette phase qui dure quelques millisecondes :

- Les instructions de programme sont exécutées une à une. Si l'état d'une entrée doit être lu par le programme, c'est la valeur stockée dans la mémoire de données qui est utilisée.
- Le programme détermine l'état des sorties et stocke ces valeurs dans une zone de la mémoire de données réservée aux sorties.
- Les entrées ne sont pas scrutées. Les sorties ne sont pas mises à jour.
- Pendant cette phase, seules la mémoire de données et la mémoire programme sont mises à contribution. Si une entrée change d'état sur le module d'entrées, l'API ne « voit » pas ce changement.

- **PHASE 3 : ECRITURE DES SORTIES (%Q)**

Durant cette phase qui dure quelques micro-secondes :

- Les états des sorties mémorisés précédemment dans la mémoire de données sont reportés sur le module de sorties.
- Les entrées ne sont pas scrutées.
- Le programme n'est pas exécuté.

L'enchaînement des trois phases se répète sans cesse de façon cyclique lorsque l'API est en fonctionnement.

Avant chaque cycle l'API effectue des traitements internes afin de vérifier ses circuits et les sollicitations extérieures. Le temps de cycle de l'ordre de quelques milli-secondes est surveillé par un circuit électronique appelé « Chien de garde ». Si pour une raison quelconque le temps de cycle mesuré par le chien de garde est supérieur au temps de cycle maxi configuré, l'API signale le défaut et arrête le traitement.

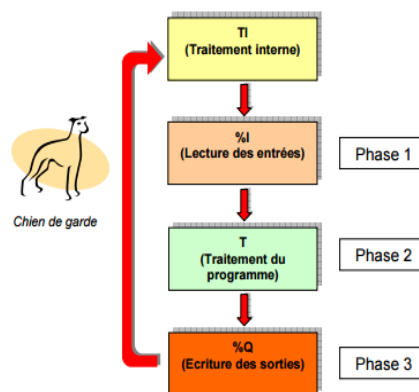


Figure I-5 Cycle d'un API.

## 8. Les langages de programmation

Les langages de programmation des systèmes automatisés sont décrits dans deux normes complémentaires: la norme CEI 61131-3 et la norme CEI 61499.

La norme CE 61131-3 définit entre autres choses, cinq langages qui peuvent être utilisés pour la programmation d'applications d'automates programmables industriels (API). Les cinq langages se divisent en deux grandes catégories [6].

### a. Langages graphiques

- **SFC** (« **Sequential Function Chart** ») : issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels.
- **FBD** (« **Function Block Diagram** » ou **schéma par blocs**) : ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- **LD** (« **Ladder Diagram** » ou **schéma à relais**) : ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true/false).

### b. Langages textuels

- **ST** (« **Structured Text** » ou **texte structuré**) : ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

– **IL (« Instruction List » ou liste d'instructions)** : ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.

Plusieurs langages ne peuvent pas être utilisés ensemble dans le même programme, à l'exception des langages LD et FBD qui peuvent être mêlés dans le même diagramme. Le langage de programmation est sélectionné au moment de la création d'un programme et ne peut pas être changé par la suite.

## 9. Présentation de la gamme SIMATIC de SIEMENS

SIMATIC est le cœur de Totally Integrated Automation, qui est un concept industriel de Siemens, il couvre toute la chaîne de production. L'architecture système est caractérisée par un matériel et un logiciel aux caractéristiques communes : gestion de données cohérente, standards internationaux et interfaces homogènes [4].

La gamme SIMATIC englobe un éventail complet de produits éprouvés pour les applications les plus diverses des industries manufacturières et de process [5].

### 9.1. SIMATIC contrôleurs modulaires

Les contrôleurs modulaires de la gamme SIMATIC sont [4] :

#### 9.1.1. ET 200

Est la station de périphérie multifonctionnelle à modularité granulaire en protection IP20. Elle se connecte aux systèmes de bus PROFIBUS et/ou PROFINET par différents coupleurs.

Les coupleurs à CPU intégrée transposent la puissance de calcul d'une CPU S7-300 directement dans la station périphérique. Ils soulagent ainsi l'automate central et le bus de terrain et permettent de réagir rapidement aux signaux à temps critique.

Les automatismes décentralisés mettent en jeu non seulement des signaux TOR et analogiques mais aussi des fonctions technologiques, des départs-moteurs, des variateurs ou un branchement pneumatique.

#### 9.1.2. S7 300

Il possède une structure modulaire et compacte. Outre les modules, il suffit de disposer d'un profilé support pour l'accrochage. Le bus de "fond de panier" est intégré dans les modules ; sa continuité de module en module est assurée par des connecteurs de bus.

La gamme de modules du S7-300 est utilisable pour des extensions en configuration centralisée, de même que pour la réalisation d'architectures décentralisées avec ET 200M, ce qui se répercute très positivement sur le stock de pièces de rechange.

Si une tâche d'automatisation exige plus que 8 modules, le châssis de base peut être complété avec des châssis d'extension. La configuration maximale possible est de 32 modules à raison de 8 par châssis. La communication entre les différents châssis est assurée de façon totalement autonome par des coupleurs IM. Dans les installations de grande envergure, les châssis de base et d'extension peuvent être installés à plus grande distance (jusqu'à 10 m).

En configuration maximale, on obtient donc 256 E/S sur une rangée, jusqu'à 1024 E/S sur plusieurs rangées et 65 536 E/S en liaison avec une périphérie décentralisée via PROFIBUS DP (jusqu'à 125 stations, par ex. ET 200M via IM 153). Les emplacements sont librement adressables, c.-à-d. qu'il ne faut pas respecter de règles d'embrochage. Donc en résumé le S7 300 est un Mini-automate modulaire et compact pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et Industrial Ethernet.

### 9.1.3. S7-400

Est un automate de haute performance avec une vaste gamme de CPU de puissance échelonnée de la CPU d'entrée de gamme à la CPU haute-performances. Les CPU diffèrent, par exemple au niveau de la mémoire de travail, de l'espace d'adressage, du nombre de blocs chargeables et du temps de traitement. Outre les CPU standard, il existe aussi deux CPU de sécurité et trois CPU à disponibilité élevée. La modularité est sa caractéristique majeure.

Le S7-400 comprend fondamentalement un châssis, une alimentation et une unité centrale. De configuration modulaire, il se prête parfaitement à des extensions. Les modules peuvent être placés librement à droite de l'alimentation. Il se distingue par son fonctionnement robuste sans ventilation, avec possibilité d'embrochage et débrochage de modules d'E/S sous tension.

Selon les tâches à accomplir, le S7-400 peut faire l'objet d'extensions en configuration centralisée ou décentralisée. Pour les configurations centralisées, on dispose de châssis d'extension et de coupleurs. La configuration décentralisée est possible via les interfaces PROFIBUS ou PROFINET intégrées de la CPU. Au besoin, on peut également utiliser des processeurs de communication (CP).

Avec le s7 400 il y a une possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industrial Ethernet.

#### **9.1.4. S7-1200**

L'automate SIMATIC S7-1200 est un mini-contrôleur modulaire utilisé pour les petites performances. Les CPU du système SIMATIC S7-1200 se déclinent en trois classes de performances : CPU 1211C, CPU 1212C et CPU 1214C, chacune d'elles pouvant être étendue en fonction des besoins de la machine. Sur chaque CPU, il est possible de greffer une platine d'extension (Signal Board) pour ajouter des E/S TOR ou analogiques supplémentaires sans avoir à changer de taille d'automate. Des modules de signaux peuvent être ajoutés à droite de la CPU pour étendre la capacité d'E/S TOR ou analogiques. La CPU 1212C supporte deux modules d'E/S supplémentaires, et la CPU 1214C huit. Il est en outre possible, sur toutes les CPU SIMATIC S7-1200, de monter jusqu'à trois modules de communication à gauche de l'automate pour disposer d'une communication optimale.

Le SIMATIC S7-1200 dispose d'une interface Industrial Ethernet/PROFINET intégrée qui garantit une communication optimale avec le système d'ingénierie intégré SIMATIC STEP 7 Basic et avec les périphériques PROFINET IO. Elle prend en charge la programmation ainsi que la communication avec les pupitres SIMATIC HMI Basic Panels pour la visualisation, avec des automates supplémentaires pour la communication de CPU à CPU et avec des appareils d'autres constructeurs pour des possibilités d'intégration élargies.

L'interface de communication SIMATIC S7-1200 se compose d'une connexion RJ45 insensible aux perturbations, avec fonctionnalité d'autocroisement, supportant de nombreuses connexions Ethernet et présentant une vitesse de transmission jusqu'à 10/100 Mbit/s.

Donc en résumé Le contrôleur S7 1200 est composé d'une CPU qui combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôle puissant.

#### **9.1.5. S7 -1500**

Cette nouvelle génération de contrôleurs est caractérisée par une haute performance et une haute efficacité. Il comporte une multitude de fonctions intégrées en standard, y compris le Motion Control, les fonctions sécurité pour une sécurité maximale en production et en développement. Les fonctions de diagnostic configurables permettent de superviser l'état de l'installation, son intégration dans portail TIA permet de concevoir de simplement des projets en optimisant les couts de développement.

## 9.2. Caractéristiques de la CPU 1214C DC/DC/RELAY (6ES7214-1HG40-0XB0)

La CPU utilisée dans notre projet est de type S7-1200 DC/DC/RELAY de référence 6ES7214-1HG40-0XB0, elle se caractérise par sa compacité avec des entrées sorties embarquées et on en trouve: 14 entrées TOR 24VCC; 10 sorties TOR RELAIS 2A; 2 entrées analogiques 0 - 10V CC, son alimentation est en CC: 20,4 - 28,8 V CC, sa mémoire programme/données est de 100 KO, mémoire de chargement 4 MO avec possibilité d'insertion d'une carte de mémoire SIMATIC [7].



Figure I-6 CPU 1214C DC/DC/RELAY.

Temps de traitement de la CPU	
pour opérations sur bits, type.	0,085 $\mu$ s; / instruction
pour opérations sur mots, type.	1,7 $\mu$ s; / instruction
pour opérations à virgule flottante, type.	2,3 $\mu$ s; / instruction
CPU-blocs	
Nombre de blocs (total)	DB, FC, FB, compteurs et temporisations Le nombre maximal de blocs va de 1 à 65535. Il n'y a pas de limitations ; utilisation de

	l'ensemble de la mémoire de travail.
• Nombre de OB, maxi	Limité uniquement par la mémoire de travail pour le code
<b>Configuration matérielle</b>	
Nombre de modules par système, maxi	3 modules de communication, 1 Signal Board, 8 modules d'entrées-sorties
<b>Entrées TOR</b>	
Nombre d'entrées TOR • dont entrées utilisables pour les fonctions Technologiques	14; intégré 6; HSC (compteur rapide)
Tension d'entrée	
• Valeur nominale • pour état log. "0" • pour état log. "1"	24 V DC 5 V à 1 mA 15 V CC à 2,5 mA
Longueur de câble	
• blindé, maxi • Non blindé, max.	500 m; pour les fonctions technologiques 300 m; Pour fonctions technologiques : Non
<b>Sorties TOR</b>	
Nombre de sorties TOR	10; Relais
Pouvoir de coupure des sorties	
• pour charge résistive, max. • pour charge de lampes, maxi	2 A 30 W pour CC, 200 W pour CA
Longueur de câble	
• blindé, maxi • Non blindé, max.	500 m 150 m
<b>Entrées analogiques</b>	
Nombre d'entrées analogiques	2
Etendues d'entrée (valeurs nominales), tensions	
• 0 à +10 V • Résistance d'entrée (0 à 10 V)	Oui ≥100 kOhm
Longueur de câble	
• blindé, maxi	100 m; torsadé et blindé
<b>Alarmes/diagnostic/information d'état</b>	
• LED RUN/STOP • LED ERROR • LED MAINT	Oui Oui Oui

Fonctions intégrées	
Nombre de compteurs 6	6
Fréquence de comptage (compteurs), maxi	100 kHz
Fréquencemètre	Oui
Positionnement en boucle ouverte	Oui
Nombre d'axes de positionnement asservis,	8
Nombre d'axe de positionnement via interface impulsion-direction	jusqu'à 4 avec SB 1222
Régulateur PID Oui	Oui
Nombre d'entrées d'alarme	4
Langages de programmation	
— CONT	Oui
— LOG	Oui
— SCL	Oui
Interfaces	
Type d'interface	PROFINET
Physique	Ethernet
Fonctionnalités	
• Contrôleur PROFINET IO	Oui
• Périphérique PROFINET IO	Oui
• Communication SIMATIC	Oui
• Communication IE ouverte	Oui
• Serveur Web	Oui
Protocoles	
Supporte le protocole pour PROFINET IO	Oui
PROFIBUS	Oui; CM 1243-5 nécessaire
AS-Interface	Oui; CM 1243-2 requis
Protocoles (Ethernet)	<ul style="list-style-type: none"> <li>• TCP/IP, SNMP, DCP, LLDP</li> </ul>
Autres protocoles	<ul style="list-style-type: none"> <li>• MODBUS</li> </ul>
Dimensions et poids	
Largeur	110 mm
Hauteur	100 mm
Profondeur	75 mm
Poids approximatif	435 g

*Tableau I -4 Caractéristiques de la CPU.*



### 9.3. Caractéristiques des modules d'entrées TOR SM 1221

Le tableau suivant contient les caractéristiques des modules d'entrées TOR SM 1221 [8] :

Caractéristiques générales	
Modèle	SM 1221 DI 16 x 24 VDC
Numéro de référence	6ES7 221-1BH32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75
Poids	210 grammes
Dissipation de courant	2,5 W
Consommation de courant (bus SM)	130 mA
Consommation de courant (24 V-)	4 mA/entrée utilisée
Entrées TOR	
Nombre d'entrées	16
Type	P/N (CEI type 1 en mode P)
Tension nominale	24 V CC à 4 mA, nominal
Tension continue admise	30 V- max
Tension de choc	35 V CC pour 0,5 s
Signal 1 logique (min.)	15 V CC à 2,5 mA
Signal 0 logique (max.)	5 V CC à 1 mA
Isolation (site à logique)	500 V CA pour 1 minute
Groupes d'isolation	4
Nombre d'entrées simultanément à 1	16
Longueur de câble (mètres)	500 m blindé, 300 m non blindé

Tableau I-5 Caractéristiques des modules d'entrées TOR SM 1221.

### 9.4. Caractéristiques des modules de sorties TOR SM 1222

Le tableau suivant contient les caractéristiques des modules de sorties TOR SM 1222 [8] :

Caractéristiques générales	
Modèle	SM 1222 DQ 16 RELAY
Numéro de référence	6ES7 222-1hH32-0XB0
Dimensions L x H x P (mm)	45 x 100 x 75
Poids	260 grammes

Dissipation de courant	8.5 W
Consommation de courant (bus SM)	135 mA
Consommation de courant (24 V-)	11 mA / bobine relais utilisée
<b>Sorties TOR</b>	
Nombre de sorties	16
Type	Relais; contact sec
Plage de tension	5 à 30 V CC ou 5 à 250 V CA
Signal 1 logique (min.)	-
Signal 0 logique (max.)	-
Courant (max.)	2A
Charge de lampe	30 W CC / 200 W CA
Résistance état activé (contact)	0,2 $\Omega$ max. lorsque neuf
Résistance d'isolation	100 M $\Omega$ min. lorsque neuf
Courant par commun (max.)	10 A
Isolation entre contacts ouverts	750 V CA pour 1 minute
Groupes d'isolation	4
Fréquence de commutation des relais	1 Hz
Comportement au passage de MARCHE à ARRET	Dernière valeur ou valeur de remplacement (valeur par défaut 0)
Nombre d'entrées simultanément à 1	<ul style="list-style-type: none"> <li>• 8 (pas d'E/S adjacentes) à 60 °C horizontal ou 50 °C vertical</li> <li>• 16 à 55 °C horizontal ou 45 °C vertical</li> </ul>
Longueur de câble (mètres)	500 m blindé, 150 m non blindé

*Tableau I -6 Caractéristiques des modules des sorties TOR SM 1222.*

## 10. Conclusion

L'automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants et langage. Son architecture complexe et modifiable selon la nécessité du projet ainsi que la maîtrise de son software sont considérés comme critères principaux pour mieux choisir son automate.

Dans ce chapitre l'étude de la gamme Siemens a orienté notre choix vers la CPU qui convient le mieux à notre application.

# **Chapitre II : TIA**

## **Portal**

## 1. Introduction

Le TIA portal (Totally Integrated Automation portal), en français Portail d'automatisation totalement intégré, est une plateforme logicielle globale, comme son nom l'indique elle intègre Tous les progiciels requis, (step 7, winCC, startdrive) de la configuration matérielle à la visualisation du processus en passant par la programmation, dans un cadre complet d'ingénierie. Le TIA Portal permet pour la première fois de disposer, au sein d'un cadre, d'un environnement de travail commun pour une ingénierie transparente avec différents systèmes SIMATIC. Il est destiné à la programmation des automates s7 1200, s7 1500 et les CPU récentes de s7 300 et s7 400.

Avec TIA Professional V13, les fonctions suivantes peuvent être utilisées pour automatiser une installation :

- Configuration et paramétrage du matériel.
- Programmation et paramétrage de la communication.
- Test, mise en service et dépannage avec les fonctions d'exploitation et de diagnostic.
- Documentation.
- Génération d'écrans de visualisation pour les Basic Panels SIMATIC avec WinCC Basic intégré. Il est également possible de générer des écrans de visualisation pour les PC et autres Panels à l'aide d'autres progiciels WinCC.

## 2. Vue du portail et vue du projet

Lorsqu'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue :

- La vue du portail : elle est axée sur les tâches à exécuter et sa prise en main est très rapide.
- La vue du projet : elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue.

### 2.1. vue du portail

La vue du portail offre un aperçu de toutes les étapes de configuration du projet et un accès orienté tâche de votre tâche d'automatisation.

Les différents portails ("Démarrage", "Appareils et réseaux", "Programmation API", "Visualisation", " En ligne et diagnostic", etc.) montrent de manière claire et ordonnée l'ensemble des étapes de travail nécessaires à l'exécution d'une tâche d'automatisation. La figure suivante montre la structure de la vue de portail :

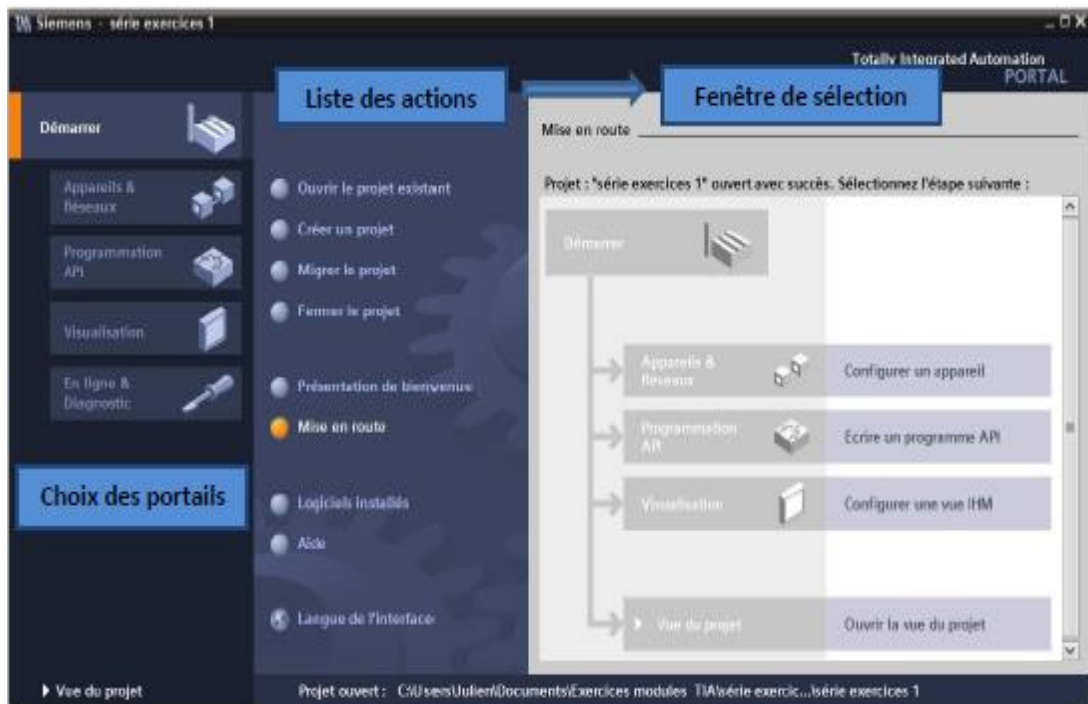


Figure II-1 Vue de portail.

## 2.2. Vue du projet

L'élément « Projet » contient l'ensemble des éléments et de données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée.

**-La fenêtre de travail :** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...

**-La fenêtre d'inspection :** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).

**-Les onglets de sélection de tâches :** ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèques des composants, bloc de programme, instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas. Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres.

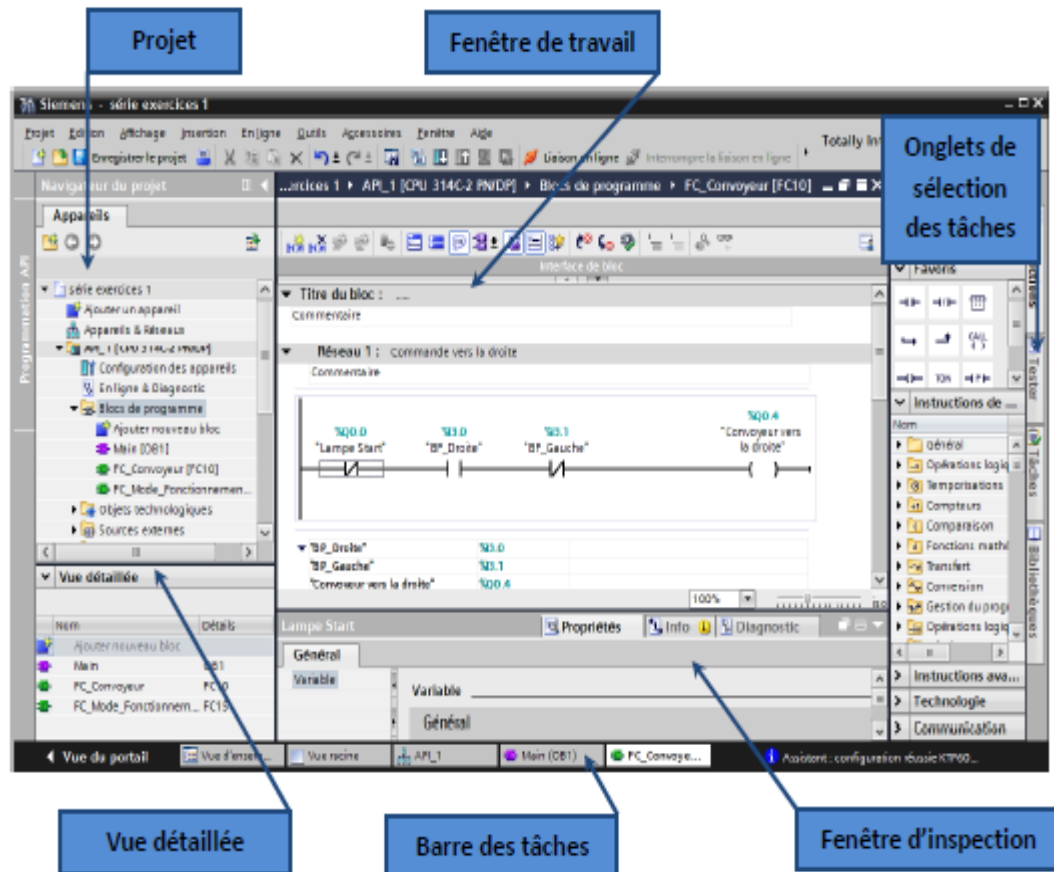


Figure II-2 Vue projet.

### 3. Création d'un projet

La création d'un projet commence toujours par la configuration du matériel. Cette dernière revient à lister tous les modules présents dans le projet. Par exemple l'alimentation, la CPU, les entrées-sorties, les modules de communications, etc... tous ces éléments se trouvent dans la bibliothèque du projet.

Pour créer un projet dans la vue du portail, il faut sélectionner l'action « Créer un projet ». On peut donner un nom au projet, choisir un chemin où il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet. Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « créer ».

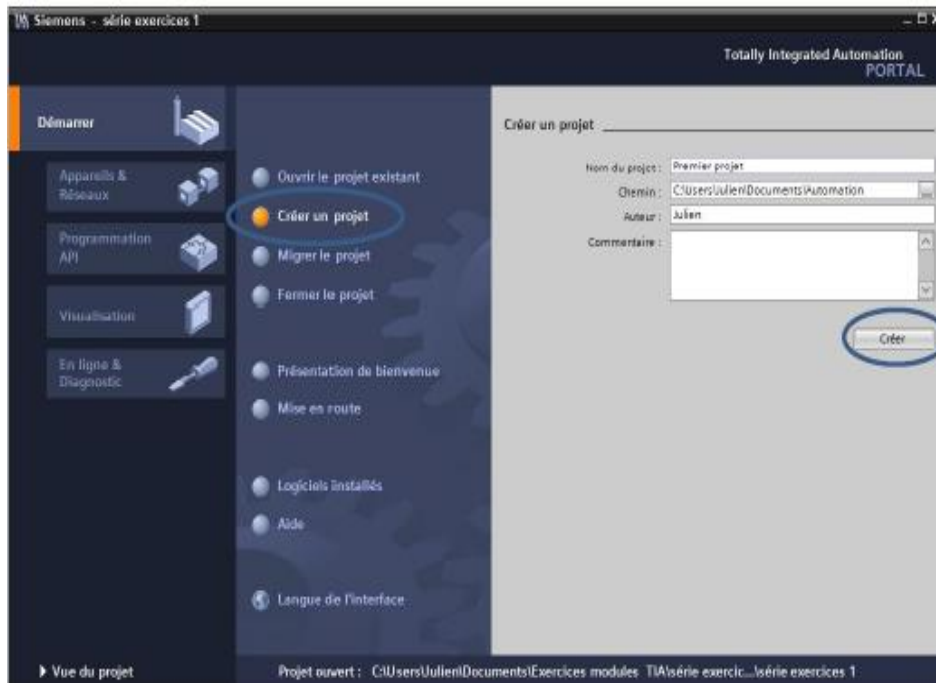


Figure II-3 Vue création de projet.

### 3.1.identification du matériel

Le moyen le plus sûr d'identifier le matériel est d'utiliser le nom et le numéro de référence parfois un numéro de firmware peut-être présent. L'identification de chaque module se fait toujours à l'aide des critères cités comme le montre les exemples suivants :

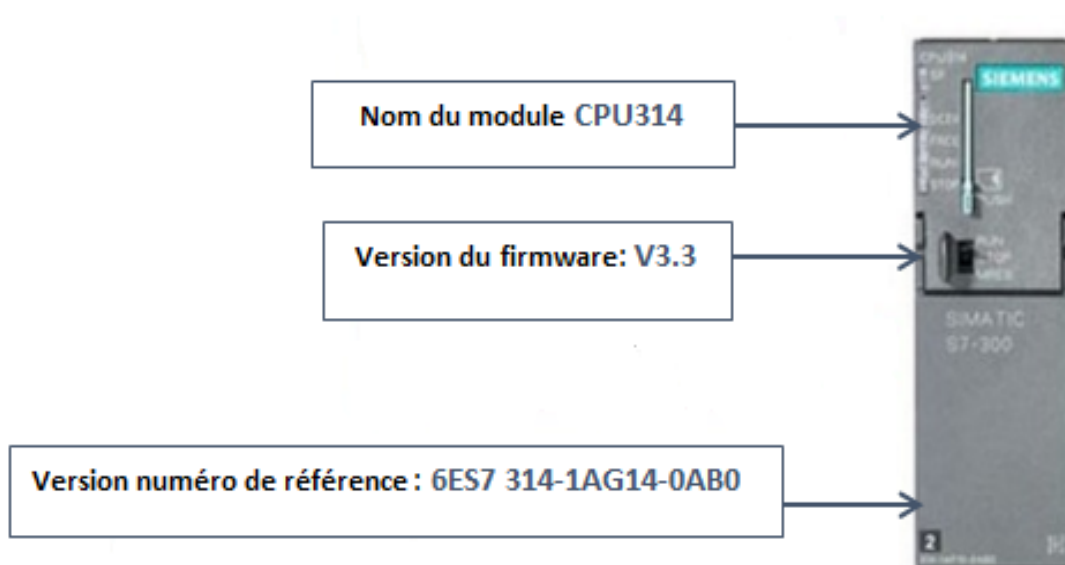


Figure II-4 Identification d'un module.

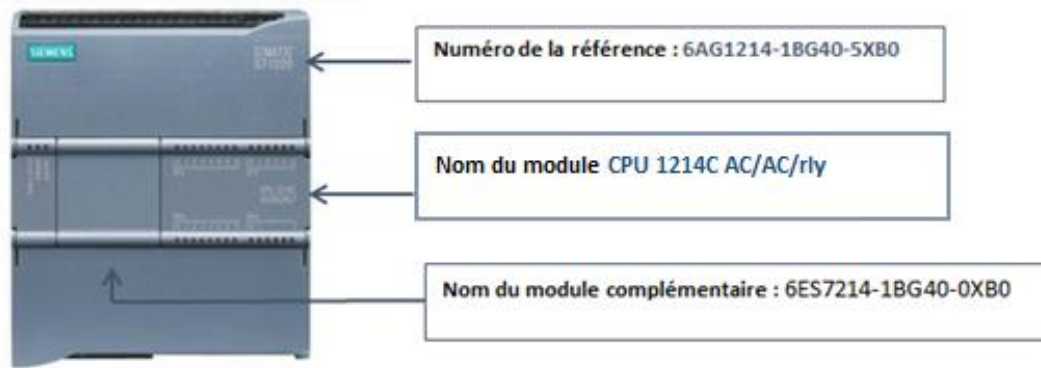


Figure II-5 Identification d'une CPU S7-1200.

### 3.2. Configuration et paramétrage du matériel

Les modules complémentaires de l'API (alimentation, E/S TOR ou analogiques, module de communication AS-i,...) peuvent être ajoutés en utilisant le catalogue. Si l'on veut ajouter un écran ou un autre API, il faut repasser par la commande « ajouter un appareil » dans le navigateur du projet.

Une fois le projet crée, on peut configurer la station de travail. La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la **vue du projet** et cliquer sur « **ajouter un appareil** » dans le navigateur du projet.

Dans cette partie on sélectionne « contrôleurs ». Lorsque l'élément à insérer est sélectionné dans le projet, une description est proposée dans l'onglet information.

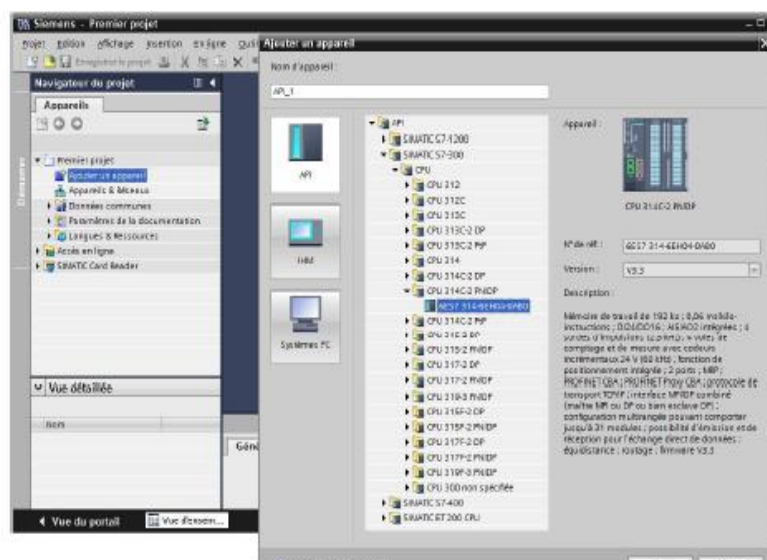


Figure II-6 Onglet d'information.



### 3.3. Les variables API

#### 3.3.1. Adresses symbolique et absolue

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,...) possèdent une adresse symbolique et une adresse absolue.

- L'adresse absolue représente l'identificateur de l'opérande (I, Q, M) et son adresse et numéro de bit et elle revient à utiliser l'adresse physique de l'automate.
- L'adresse symbolique correspond au nom que l'utilisateur a donné à la variable (ex : Bouton Marche).

On utilise généralement l'adressage symbolique pour donner de la clarté au programme. Cela revient à substituer chaque adresse physique par une adresse symbolique qui représente au mieux la fonction. Le lien entre les adresses symbolique et absolue se fait dans la table des variables API. Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

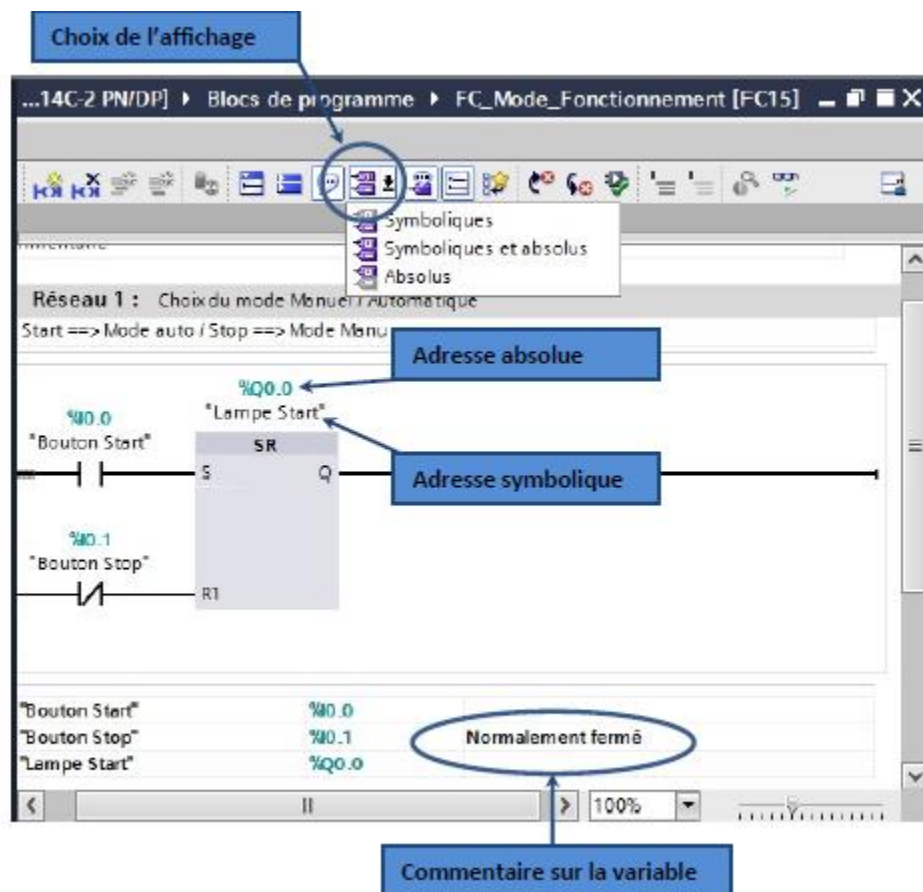


Figure II-7 Exemple d'adresse symbolique et physique.

### 3.3.2. Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matériel, on sélectionne « **appareil et réseau** » dans le navigateur du projet.

Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « **Vue des appareils** » et de sélectionner l'appareil voulu. On sélectionne la CPU puis à l'aide des deux petites flèches (voir figure), on fait apparaître l'onglet « **Vue d'ensemble des appareils** ». Les adresses des entrées et sorties apparaissent. On peut les modifier en entrant une nouvelle valeur dans la case correspondante.

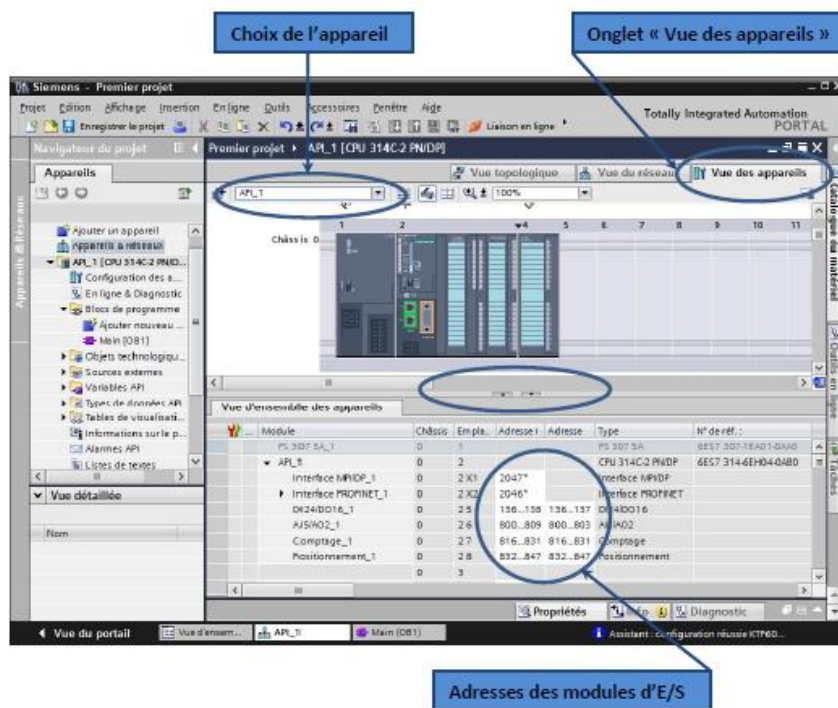


Figure II-8 Vue des appareils.

### 3.4. Table des variables API

Pour TIA Portal, la table de variables est l'outil qui permet de créer un adressage symbolique. C'est dans cette table qu'on va pouvoir déclarer toutes les variables et constantes utilisées dans le programme. Lorsqu'on définit une variable API, il faut définir :

- Un nom : c'est l'adressage symbolique de la variable.
- Le type de donnée : BOOL, INT, REAL... etc.
- L'adresse absolue : par exemple %Q1.5.

On peut également insérer un commentaire qui nous renseigne, sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

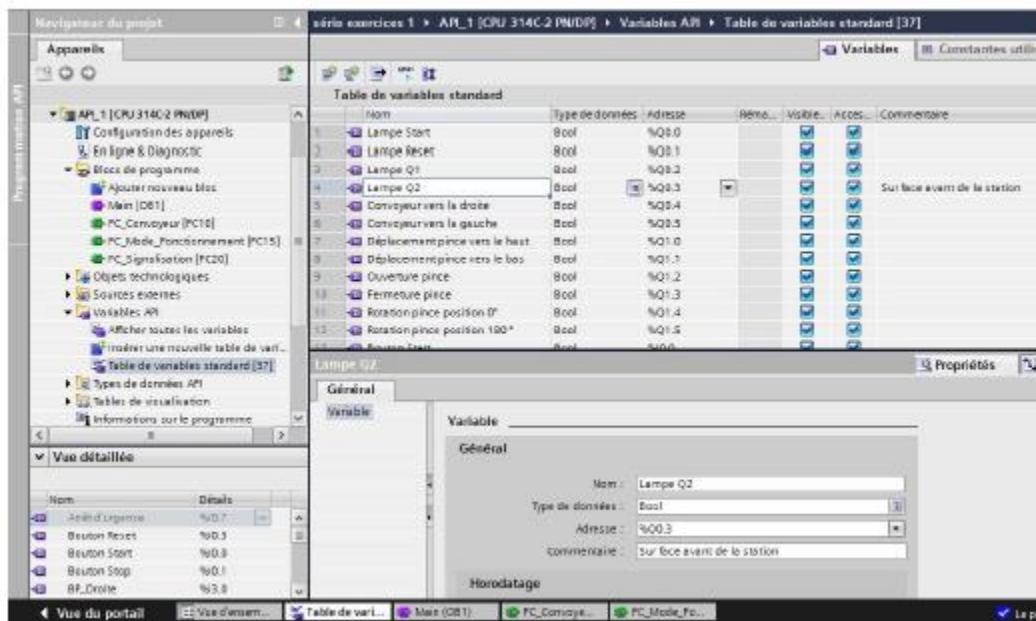


Figure II-9 Table de variable.

### 3.5. Tableau d'affectation

Le tableau d'affectation est très utile pour connaître les entrées, sorties et mémoires déjà utilisées. Pour y accéder on sélectionne d'abord « Sauvegarde en ligne », puis « Information sur le programme » ensuite l'onglet « Tableau d'affectation ».

The screenshot shows the 'Tableau d'affectation de CPU 314' (CPU 314 Allocation Table) with two main sections: 'Entrée, sortie' (Input, output) and 'Mémoire' (Memory).

**Entrée, sortie:**

Adresse	7	6	5	4	3	2	1	0	B	W	DWORD	LWORD
IB0												
IB1												
QB4												
QB5												

**Mémoire:**

Adresse	7	6	5	4	3	2	1	0	B	W	DWORD	LWORD
MB10												
MB20												
MB50												
MB51												
MB60												
MB61												
MB62												
MB63												

Figure II-10 Table d'affectation.

Dans ce tableau d'affectation on voit que les bits d'entrées : I0.1- I0.3, I1.5- I1.7 et les mémoires M10.2- M20.4 sont utilisés. Ainsi que le Word MW50 et le double Word MD60.

### 3.6. Les références croisées

Les références croisées sont des liens qui permettent d'afficher les différents contacts et blocs de fonction lié à une entrée ou à une sortie. Ces liens permettent de retrouver facilement sans avoir à fouiller tout le programme un élément pour faire une modification ou un contrôle lors d'un dépannage par exemple.

## 4. Les blocs dans TIA Portal

L'automate met à disposition différents types de blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC [9].

### 4.1. Les blocs d'organisation – OB

Les OB sont appelés par le système d'exploitation en liaison avec les événements suivants :

- Exécution cyclique du programme et traitement des erreurs.
- Exécution du programme déclenchée par des alarmes (processus, diagnostic...).

Pour que le traitement du programme le projet doit posséder un OB cyclique par exemple OB1

### 4.2. Les fonctions – FC

Ce sont des blocs de code sans mémoire, les données des variables temporaires sont perdues après l'exécution de la fonction, si on veut mémoriser ces données, il faut utiliser des opérandes globaux, Elles sont utilisées pour la programmation de fonction utilisées plusieurs fois.

### 4.3. Les blocs fonctionnels – FB

Ce sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement de blocs.

### 4.4. Les blocs de données – DB

Les blocs de donnée (DB) sont des zones données du programme qui contiennent des données utilisateur. On distingue deux types de bloc de donnée :

**DB globale :** Un DB global peut être créé indépendamment des autres blocs. Tous les blocs (FB, FC et OB) peuvent lire ou écrire les données contenues dans un bloc de données global.

**DB d'instance** : Le DB d'instance est associé à un certain bloc fonctionnel et contient les données locales de ce FB associé. Il ne peut être créé que si le FB associé est déjà présent dans le programme.

#### 4.5. Création et programmation d'un bloc

Quatre genres de blocs peuvent être insérés : OB- FB- FC- DB. A part lors d'une programmation en GRAFCET ou d'une programmation d'un niveau avancé tous les blocs insérés seront des FC.

Sélectionner Bloc de programme double clic sur ajouter nouveau bloc. On sélectionne par exemple FC, on choisit le langage CONT (ou SCL) et on clique sur OK :

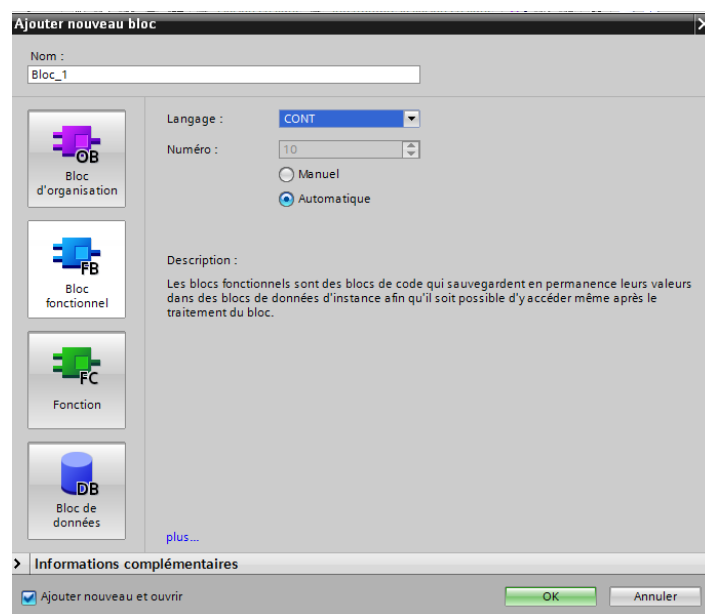


Figure II-11 Vue des blocs disponibles.

La fenêtre du Bloc\_1 est ouverte et prête pour la programmation, pour tester le fonctionnement, saisissons : Entrée : %I0.0, Sortie : %Q0.0.



Figure II-12 Programme d'un bloc.

#### 4.6. Appel d'un bloc depuis l'OB1

Les blocs de type FC et FB seront appelés dans l'OB1. Après l'ouverture de ce dernier on sélectionne le « Bloc\_1 », on le glisse et le déposer sur le réseau 1 de OB1. On procède de la même manière avec les blocs de fonction FB.



Figure II-13 Appel d'un bloc dans OBI.

## 5. Les langages de programmation dans le TIA Portal

Le SIMATIC STEP 7 V13 propose des éditeurs de programmation performants pour la programmation des automates SIMATIC S7 :

- Le langage structuré (SCL), le schéma à contacts (CONT) et le logigramme (LOG) sont disponibles pour tous les automates.
- La liste d'instructions (LIST) et la programmation en graphes séquentiels (GRAPH, SFC) existent en outre pour les gammes d'automates S7-1500, S7-300, S7-400 et WinAC [9].

## 6. Compilation et chargement

La compilation est nécessaire au premier chargement et lors d'une modification de la configuration du matériel par exemple l'ajout ou suppression d'un module ou la modification des paramètres d'un module (mémento de cadence, rémanence, temps de cycle, adresse de byte, etc...).

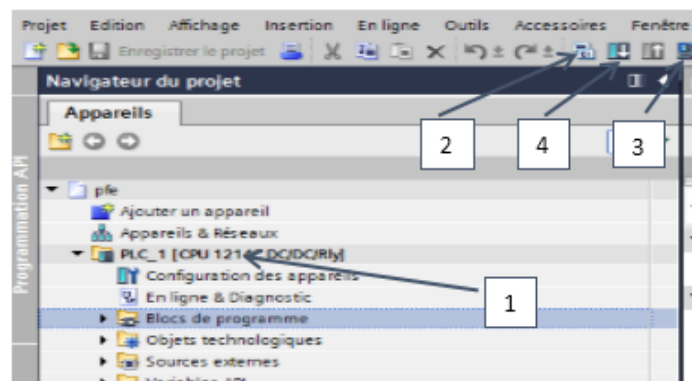


Figure II-14 Les étapes de compilation.

1. Sélectionner le dossier.
2. PLC\_1.Compiler le projet.
3. Démarrer la simulation.
4. Charger dans l'automate.

## 7. La simulation avec PLCSIM

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés dans le programme par exemple : activer ou de désactiver des entrées.

En outre, S7-PLCSIM offre la possibilité de changer l'état de fonctionnement de la CPU (STOP, RUN et RUNP) comme pour une CPU réelle. De plus, il dispose d'une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du programme.



Figure II-15 Etat de la CPU.

On peut créer des « fenêtres » dans lesquelles on a la possibilité d'accéder aux zones mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par adressage symbolique.

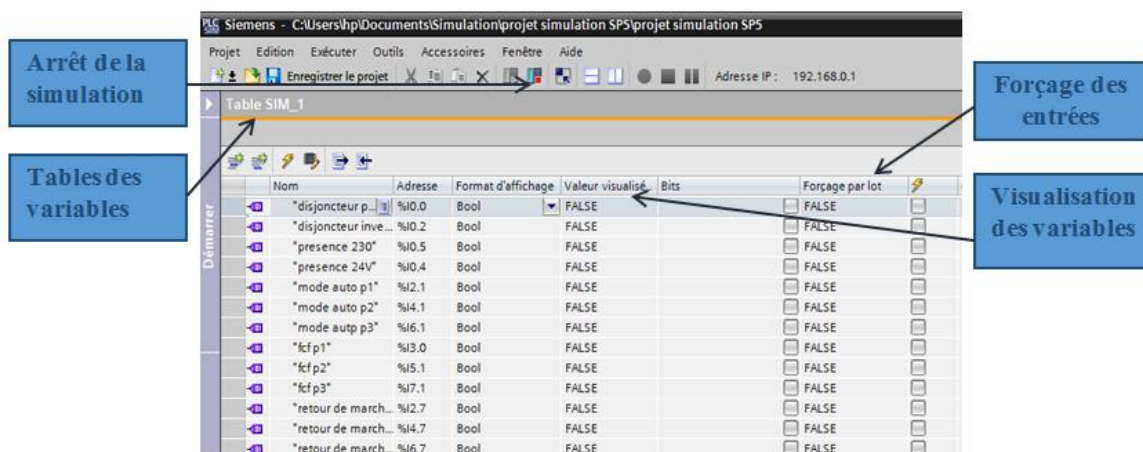


Figure II-16 Vue PLCSIM.

## 8. Supervision du processus

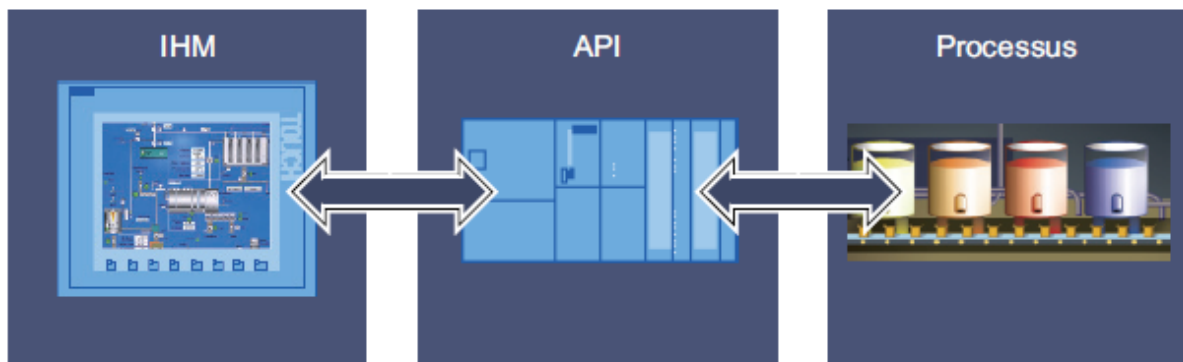
Lors de la conception d'un système automatisé, on a souvent besoin de concevoir une partie contrôle commande ou supervision, permettant de surveiller et de contrôler le système. Dans la nouvelle plateforme TIA le logiciel de supervision et de génération d'écran de visualisation Win CC est intégré ce qui facilite la tâche de conception des vues.

## 8.1. Présentation du WinCC sous TIA Portal

Le SIMATIC WinCC dans le TIA Portal est le logiciel pour la programmation et la configuration de solutions de commande, de visualisation et d'entraînement des applications HMI allant de solutions de commande simples avec des Basic Panels aux applications SCADA pour systèmes multipostes basés sur PC. SIMATIC WinCC se décline en plusieurs versions échelonnées en prix et performances. Ces versions s'appuient l'une sur l'autre et sont adaptées de façon optimale aux différentes classes de pupitres opérateurs. Le package de niveau supérieur englobe toujours les capacités de configuration du package de niveau inférieur [10].

## 8.2. Définition de l'interface homme-machine (HMI)

Le système d'interface homme-machine (IHM) constitue l'interface entre l'opérateur et le processus. Le déroulement du processus est commandé par la CPU. L'opérateur peut visualiser le processus ou intervenir dans le processus en cours par le biais d'un pupitre opérateur.



*Figure II-17 Schéma explicatif.*

Les possibilités suivantes vous sont en autres offertes pour le contrôle-commande des machines et installations :

**a) Représentation du processus :**

Le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue.

**b) Commande du processus :**

L'opérateur peut commander le processus via l'interface utilisateur graphique.

**c) Vue des alarmes :**

Lorsque surviennent des états critiques dans le processus, une alarme est immédiatement Déclenchée.

**d) Archivage de valeurs processus et d'alarmes :**



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

**e) Documentation de valeurs processus et d'alarmes :**

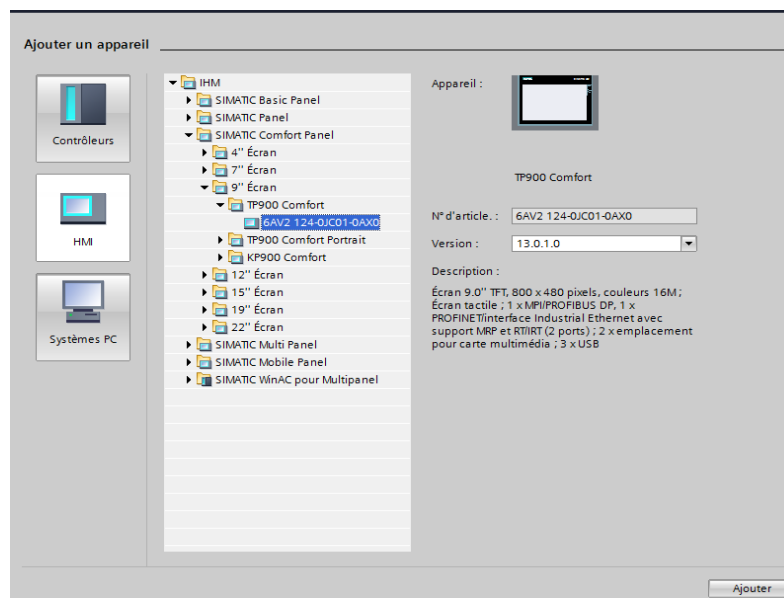
Les alarmes et valeurs processus peuvent être éditées par le système IHM sous forme de journal.

**f) Gestion des paramètres de processus et de machine :**

Les paramètres du processus et des machines peuvent être enregistrés au sein du système IHM dans des recettes [10].

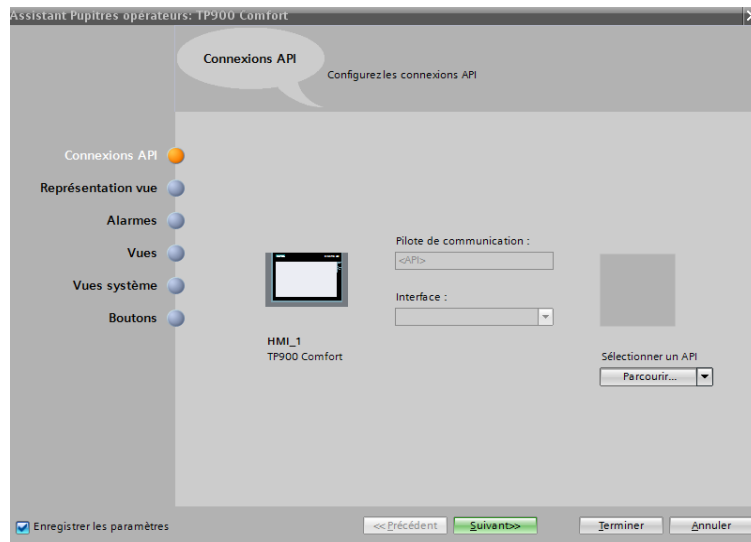
### 8.3. Configuration du HMI dans le TIA portal

Comme précédemment une fois le projet crée, on peut configurer la station de travail. La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la **vue du projet** et cliquer sur « **ajouter un appareil** » dans le navigateur du projet. Et on sélectionne HMI. On bien directement à partir de **la vue portail** en cliquant sur « **Configurer un appareil** », puis sur « **Ajouter un appareil** » et on sélectionne HMI. On peut nommer l’HMI dans le champ « **nom d’appareil** ».



*Figure II-18 Vue configuration HMI.*

On choisit par exemple le TP900 COMFORT une description des caractéristique de matériel s’affiche, on clique sur « ajouter » pour l’ajout de l’appareil. Après le choix de ce dernier une fenêtre d’assistance s’affiche qui contient les paramètres généraux de la configuration de l’HMI choisi par exemple la liaison avec l’API comme la figure suivante le montre :



*Figure II-19 Assistant HMI.*

#### 8.4.Éléments de l'interface utilisateur de WinCC

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

- **La fenêtre de travail :** permet de visualiser l'HMI sélectionnée dans le projet.
- **Les onglets de sélection de tâches :** ont un contenu qui varie en fonction de l'objet sélectionné (accessoire, animation, apparence, instruction du script, tâches bibliothèques).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

- **La 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.
- **Fenêtre de projet :** La fenêtre du projet est le poste central de traitement. Tous les éléments et tous les éditeurs disponibles d'un projet sont affichés sous forme d'arborescence et peuvent être ouverts à partir de cette fenêtre. A chaque éditeur correspond une icône qui permet d'identifier les objets qui lui sont associés. Seuls les éléments pris en charge par le pupitre opérateur sélectionné apparaissent dans cette fenêtre. On peut aussi accéder aux paramètres du pupitre, à la localisation et à la gestion de versions.

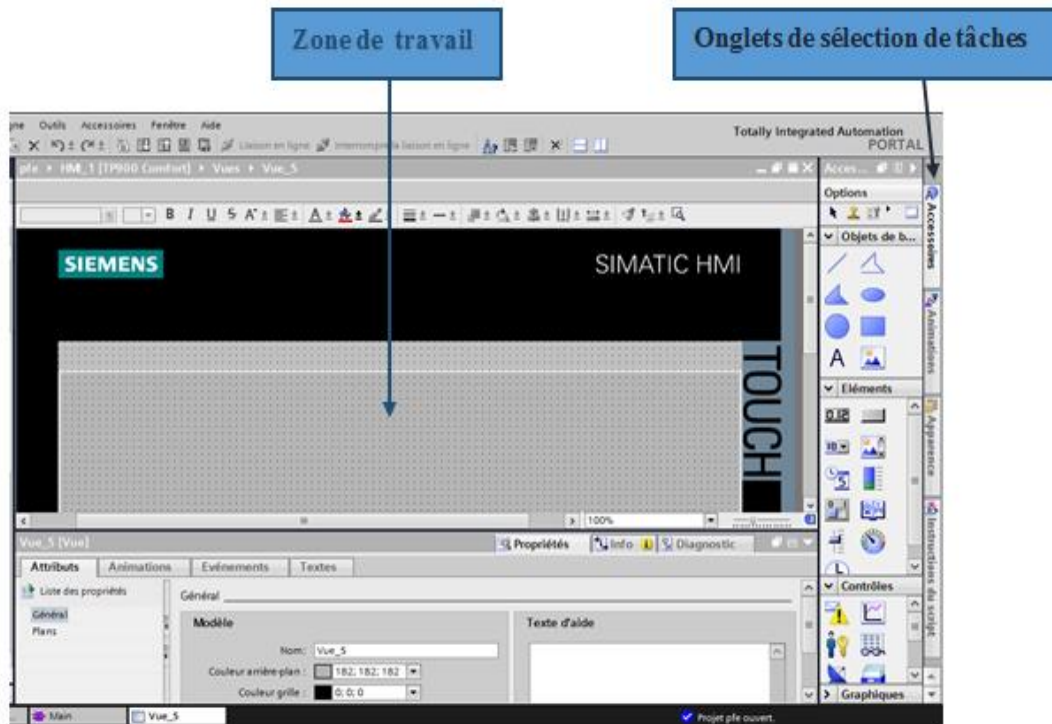


Figure II-20 Vue d'interface WinCC.

## 8.5. Définition des variables

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

### 1. 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.

### 2. 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. La déclaration des variables s'effectue dans l'éditeur "Variables" .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.

## 8.6. Les alarmes dans WinCC

Les alarmes montrent les événements où les anomalies qui peuvent survenir dans le processus, elles peuvent servir par exemple à diagnostiquer les erreurs.

The screenshot shows the 'Alarms HMI' window in WinCC. It contains a table with the following columns: ID, Texte d'alarme, Classe d'alar..., Variable de d..., Bit de..., Adresse de dé..., Variable d'acq..., Bit d'a..., Adresse d'acq..., and Journal. The table lists 15 alarm entries, all with the class 'Errors' and the variable 'mot de défaut'. The reset addresses are also listed as 'mot de reset...'. A '<ajouter>' button is visible at the bottom of the table.

ID	Texte d'alarme	Classe d'alar...	Variable de d...	Bit de...	Adresse de dé...	Variable d'acq...	Bit d'a...	Adresse d'acq...	Journal
7	defaut discordance pompi	Errors	mot de de...	14	*mot de defa...	mot de res...	14	*mot de reset...	
8	defaut vanne 3	Errors	mot de défaut	15	*mot de defa...	mot de reset	15	*mot de reset...	
9	defaut temperature pomp	Errors	mot de défaut	0	*mot de defa...	mot de reset	0	*mot de reset...	
10	defaut discordance pompi	Errors	mot de défaut	1	*mot de defa...	mot de reset	1	*mot de reset...	
11	defaut vanne 4	Errors	mot de défaut	2	*mot de defa...	mot de reset	2	*mot de reset...	
12	defaut temperature pomp	Errors	mot de défaut	3	*mot de defa...	mot de reset	3	*mot de reset...	
1	defaut discordance pompi	Errors	mot de défaut	8	*mot de defa...	mot de reset	8	*mot de reset...	
6	defaut temperature pomp	Errors	mot de défaut	13	*mot de defa...	mot de reset	13	*mot de reset...	
2	defaut vanne 1	Errors	mot de défaut	9	*mot de defa...	mot de reset	9	*mot de reset...	
3	defaut temperature pomp	Errors	mot de défaut	10	*mot de defa...	mot de reset	10	*mot de reset...	
4	defaut discordance pompi	Errors	mot de défaut	11	*mot de defa...	mot de reset	11	*mot de reset...	
5	defaut vanne 2	Errors	mot de défaut	12	*mot de defa...	mot de reset	12	*mot de reset...	
13	defaut discordance pompi	Errors	mot de défaut	4	*mot de defa...	mot de reset	4	*mot de reset...	
14	defaut vanne 5	Errors	mot de défaut	5	*mot de defa...	mot de reset	5	*mot de reset...	
15	defaut temperature pomp	Errors	mot de défaut	6	*mot de defa...	mot de reset	6	*mot de reset...	

Figure II-21 Vue des alarmes HMI.

### 8.6.1. Classes d'alarmes

- \_ **"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"** La classe d'alarmes "System" englobe les alarmes qui indiquent les états du pupitre opérateur et des automates. Les messages de la classe d'alarmes "System" appartiennent aux messages système.
- \_ **"Diagnostic"** La classe d'alarmes "Diagnostic Events" englobe les alarmes qui indiquent les états et les événements des automates SIMATIC S7. L'utilisateur n'acquiesce pas les alarmes appartenant à cette classe [11].

The screenshot shows the 'Alarms IHM' configuration window in WinCC. The window title is 'pfe > HM\_1 [TP900 Comfort] > Alarms IHM'. Below the title bar, there are several tabs: 'Alarmes de bit', 'Alarmes analogiques', 'Alarmes de l'API', 'Alarmes système', and 'Classes d'alarmes'. The 'Classes d'alarmes' tab is active, displaying a table with the following columns: 'Nom d'affichage', 'Nom', 'Modèle d'acquiescement', 'Archive', 'Adresse e-mail', 'Couleur...', and 'Couleur...'. The table contains six rows of alarm classes.

Nom d'affichage	Nom	Modèle d'acquiescement	Archive	Adresse e-mail	Couleur...	Couleur...
!	Errors	Alarme à acquiescement simple	<aucune archiv...		255...	255...
	Warnings	Alarme sans acquiescement	<aucune archiv...		255...	255...
\$	System	Alarme sans acquiescement	<aucune archiv...		255...	255...
S7	Diagnosis events	Alarme sans acquiescement	<aucune archiv...		255...	255...
A	Acknowledgement	Alarme à acquiescement simple	<aucune archiv...		255...	255...
NA	No Acknowledgement	Alarme sans acquiescement	<aucune archiv...		255...	255...

Figure II-22 Vue classe d'alarme.

## 8.6.2. Types des alarmes

WinCC prend en charge les types d'alarme suivants :

1. Les alarmes définies par le système servent à surveiller le pupitre opérateur ou l'automate.
2. Alarmes définies par l'utilisateur qui servent à surveiller l'installation.
  - **Alarmes analogiques** : Les alarmes analogiques indiquent les dépassements de valeurs limites dans le process en cours.
  - **Alarmes de bit** : Les alarmes de bit indiquent un état dans le process en cour.
  - **Alarmes utilisateur** : Les alarmes utilisateur indiquent des actions opérateur dans Runtime. Au besoin, on configure des alarmes utilisateur également dans des scripts dans Runtime pour des applications définies par l'utilisateur.
  - **Alarmes de l'API** : Une alarme de l'API définie par l'utilisateur est programmée dans STEP 7 par le concepteur de l'automate. Les valeurs d'état de l'automate, tels que horodatage et valeurs de process, y sont intégrées. Quand des alarmes de l'API sont configurées dans STEP 7, on les reprend dans le fonctionnement intégré de WinCC dès qu'une connexion à l'automate est établie [11].

## 8.6.3. Configuration d'alarme

### 8.6.3.1. Editeur "Alarmes TOR"

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 M0 .7, M 1.0 à

M 1.7.

De même on crée un autre mot qui va servir à acquitter les alarmes si leur type est à acquitter  
 Pour chaque alarme on doit définir :

- **Numéro (ID):** numéro unique servant à l'identification de l'alarme (valeurs admissibles : de 1 à 32767).
- **Un texte d'alarme :** le texte qui va s'afficher au Runtime.
- **Classe d'alarme :** on choisit pour chaque alarme une classe d'alarme parmi celles définies ci-dessus (erreur, avertissement, acquittée, non acquittée).
- **Variable de déclenchement :** c'est la variable de type INT créée auparavant.
- **Adresse de déclenchement :** on choisit un bit du mot créé.
- **Variable d'acquiescement :** la variable créée pour l'acquiescement.
- **Adresse d'acquiescement :** on choisit un bit du mot d'acquiescement.

ID	Texte d'alarme	Classe d'alar...	Variable de décl...	Bit de décl...	Adresse de déclenche...	Variable d'acquiesceme...	Bit d'acquiescement IHM	Adresse d'acquit...
7	defaut discordance pomp	Errors	mot de défaut	14	"mot de défaut".x14	mot de reset	14	"mot de reset".x...
8	defaut vanne 3	Errors	mot de défaut	15	"mot de défaut".x15	mot de reset	15	"mot de reset".x...
9	defaut temperature pomp	Errors	mot de défaut	0	"mot de défaut".x0	mot de reset	0	"mot de reset".x0
10	defaut discordance pomp	Errors	mot de défaut	1	"mot de défaut".x1	mot de reset	1	"mot de reset".x1
11	defaut vanne 4	Errors	mot de défaut	2	"mot de défaut".x2	mot de reset	2	"mot de reset".x2
12	defaut temperature pomp	Errors	mot de défaut	3	"mot de défaut".x3	mot de reset	3	"mot de reset".x3
1	defaut discordance pomp	Errors	mot de défaut	8	"mot de défaut".x8	mot de reset	8	"mot de reset".x8
6	defaut temperature pomp	Errors	mot de défaut	13	"mot de défaut".x13	mot de reset	13	"mot de reset".x...
2	defaut vanne 1	Errors	mot de défaut	9	"mot de défaut".x9	mot de reset	9	"mot de reset".x9
3	defaut temperature pomp	Errors	mot de défaut	10	"mot de défaut".x10	mot de reset	10	"mot de reset".x...
4	defaut discordance pomp	Errors	mot de défaut	11	"mot de défaut".x11	mot de reset	11	"mot de reset".x...
5	defaut vanne 2	Errors	mot de défaut	12	"mot de défaut".x12	mot de reset	12	"mot de reset".x...
13	defaut discordance pomp	Errors	mot de défaut	4	"mot de défaut".x4	mot de reset	4	"mot de reset".x4
14	defaut vanne 5	Errors	mot de défaut	5	"mot de défaut".x5	mot de reset	5	"mot de reset".x5
15	defaut temperature pomp	Errors	mot de défaut	6	"mot de défaut".x6	mot de reset	6	"mot de reset".x6

Figure II-23 Vue configuration d'alarme de bit.

### 8.6.3.2. Éditeur "Alarmes Analogiques"

Cet éditeur permet de créer des alarmes analogiques et d'en définir les propriétés.

- **Numéro (ID):** numéro unique servant à l'identification de l'alarme (valeurs admissibles : de 1 à 32767).
- **Texte d'alarme:** Le texte du message est affiché au Runtime.
- **Classe d'alarme:** 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.
- **variable de Déclenchement**: 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.

#### 8.6.4. Etats des alarmes

Chaque alarme se trouve dans un état. Les états des alarmes résultent des événements suivants :

- **Apparaissant** : La condition de déclenchement d'une alarme est remplie. L'alarme s'affiche, par ex. "Pression cuve trop élevée".
- **Disparaissant** : La condition de déclenchement d'une alarme n'est plus remplie. L'alarme ne s'affiche plus, car la cuve a été purgée.
- **Acquitter** : L'utilisateur acquitte l'alarme.

#### 8.7. Les 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 [11].

##### 8.7.1. Editeur de recette

On peut créer une recette à partir de la fenêtre "Projet " en cliquant sur " Recettes ".

Les différents éléments de la recette sont définis les paramètres :

**Nom de la recette** : Identifie la recette de manière univoque dans le pupitre opérateur.

**Nom d'affichage** : S'affiche dans Runtime, par exemple dans la vue de recette. On peut configurer le nom d'affichage dans plusieurs langues. On peut Attribuer des noms ou désignations explicites que l'opérateur peut associer directement à une recette.

**Numéro de recette** : Identifie la recette de manière univoque dans le pupitre opérateur.

**Version** : Informations sur la recette. Par défaut, ce sont la date et l'heure de la dernière modification apportée à la recette qui s'affichent.

**Chemin :** Définit le lieu de stockage pour les recettes. Les recettes sont stockées sous forme de fichier.

**Type de taille [non modifiable] :** Par défaut, les enregistrements de recette sont limités à un nombre prédéfini.

**Nombre d'enregistrements [non modifiable] :** Nombre maximal d'enregistrements d'une recette dans Runtime. Ce nombre est limité par la mémoire de recettes du pupitre opérateur.

**Type de communication [non modifiable] :** Les enregistrements de recette sont écrits directement dans les adresses des variables de recette ou y sont directement lus.

**Info-bulle :** Info-bulle sur la recette, affichée pour l'opérateur dans Runtime [11].

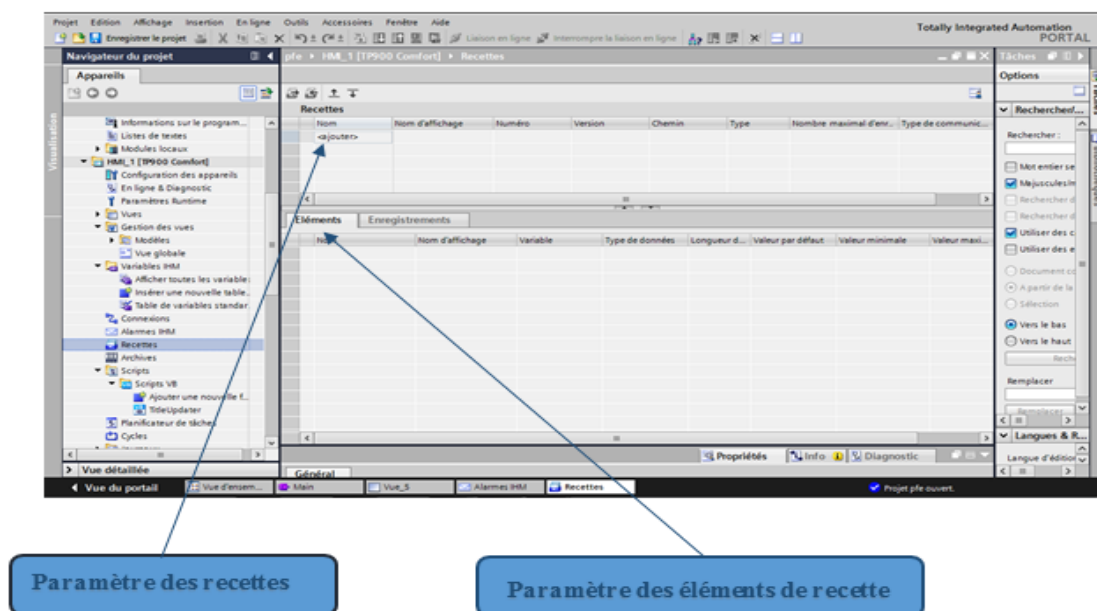


Figure II-24 Vue configuration de recette.

## 8.8. Les archives

Le WinCC propose les types d'archives suivants pour archiver des données de process pour HMI Runtime [11]:

### 8.8.1. Archives d'alarmes

Dans les archives d'alarmes, toutes les données concernant une alarme qui surviennent dans le process visualisé, y compris les données de configuration, sont enregistrées. A partir des archives, on peut lire par exemple toutes les propriétés d'une alarme, comme la classe d'alarme, l'horodatage et les textes de l'alarme. Une modification ultérieure des données de configuration



d'une alarme a pour conséquence qu'un nouveau segment d'archive avec les nouvelles données de configuration est créé. Cette fonction empêche la modification d'avoir une influence sur des alarmes archivées déjà avant la modification. Pour archiver une alarme, réalisez les étapes de configuration suivantes :

- 1- **Définir les "Paramètres d'archives"** : détermination des paramètres pour l'archivage d'alarmes dans les paramètres d'archives.
- 2- **Activation de l'archivage d'alarmes** : décision d'archiver les alarmes de certaines classes d'alarmes.
- 3- **Affectation de l'alarme à la classe d'alarmes archivable.**

### 8.8.2. Archives de variables

Une archive de variables sert à archiver les données de process d'une installation industrielle. Le système d'archives se charge en Runtime de l'archivage des variables. Le système d'archives traite les valeurs du process sauvegardées dans la base de données Runtime et les inscrit dans la base de données d'archives.

Les divers paramètres déterminent si et quand les valeurs du process doivent être acquises et archivées. La méthode d'archivage utilisée détermine ceux des paramètres suivants à configurer :

- **Cycle d'acquisition** : Détermine quand la valeur d'une variable du process sera lue dans l'automate programmable. On configure un cycle d'acquisition par exemple pour l'archivage cyclique des variables.
- **Cycle d'archivage** : Détermine quand la valeur du process sera enregistrée dans la base de données d'archives une fois traitée. On configure un cycle d'archivage par exemple pour l'archivage cyclique des variables.
- **Variable marche-arrêt** : Lance d'archivage des variables quand la variable binaire affectée prend la valeur "1". L'archivage s'arrête dès que la condition de démarrage n'est plus remplie. On configure la variable marche-arrêt pour l'archivage cyclique sélectif des variables.
- **Archivage sur demande** : Les valeurs de process sont archivées à la demande. La demande est pilotée par une variable binaire. Pour la configuration, utiliser le mode de déclenchement "Sur demande".

- **Archivage de variable sur modification** : Les valeurs de processus sont uniquement archivées en cas de modification. Pour la configuration, utiliser le mode de déclenchement "Sur modification".

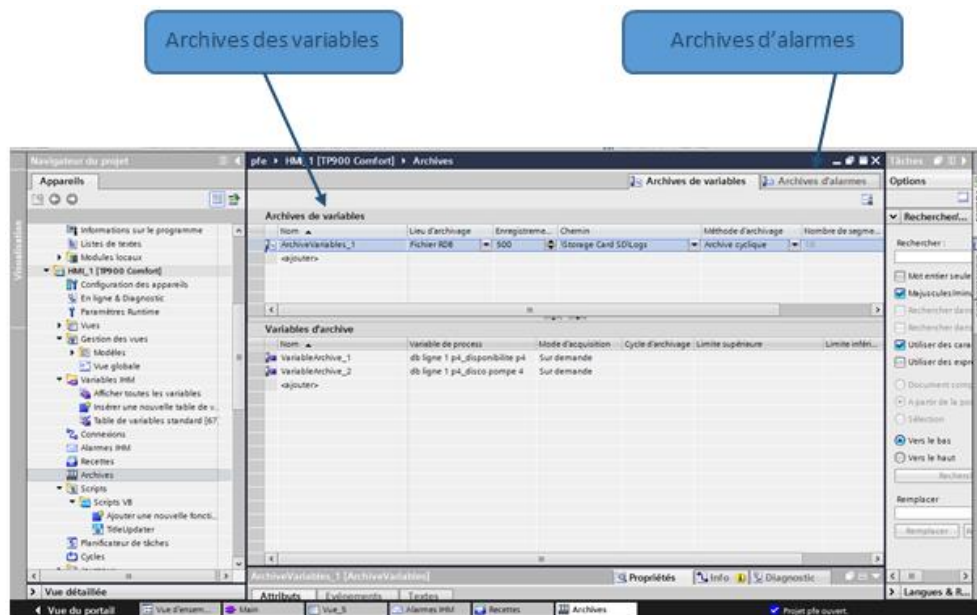


Figure II-25 Vue configuration d'archive.

Les méthodes d'archivage suivantes sont disponibles :

- **Archive cyclique** : Quand une archive cyclique est entièrement remplie, les entrées les plus anciennes sont écrasées.
- **Archive cyclique segmentée** : Dans une archive cyclique segmentée, plusieurs segments d'archive de même taille sont créés et remplis l'un après l'autre. Quand tous les segments d'archive sont entièrement remplis, le segment d'archive le plus ancien est écrasé.
- **Archive avec alarme système en fonction du niveau de remplissage** : Une alarme système se déclenche quand un niveau de remplissage défini est atteint.
- **Archive avec déclenchement d'un événement en fonction du niveau de remplissage** : Quand l'archive est entièrement remplie, l'événement "Débordement" est déclenché. Avec cet événement "Débordement", une fonction système est déclenchée.

### 8.9. Le test de la visualisation du processus

Runtime est un logiciel utilisé pour la visualisation du processus des projets créés avec le logiciel WinCC dans le TIA Portal.

La simulation de l'HMI permet de tester le bon fonctionnement de la visualisation avant de démarrer le processus.

Avant tout la configuration matérielle et les blocs de programme de la CPU doivent être chargés dans PLCSIM.

On clique sur le HMI RT dans la navigation du projet et on démarre la simulation du Runtime dans le menu contextuel.

Les éléments configurés du HMI RT sont automatiquement compilés avant le démarrage du Runtime. L'état de compilation s'affiche dans la fenêtre d'inspection de l'onglet "Info".

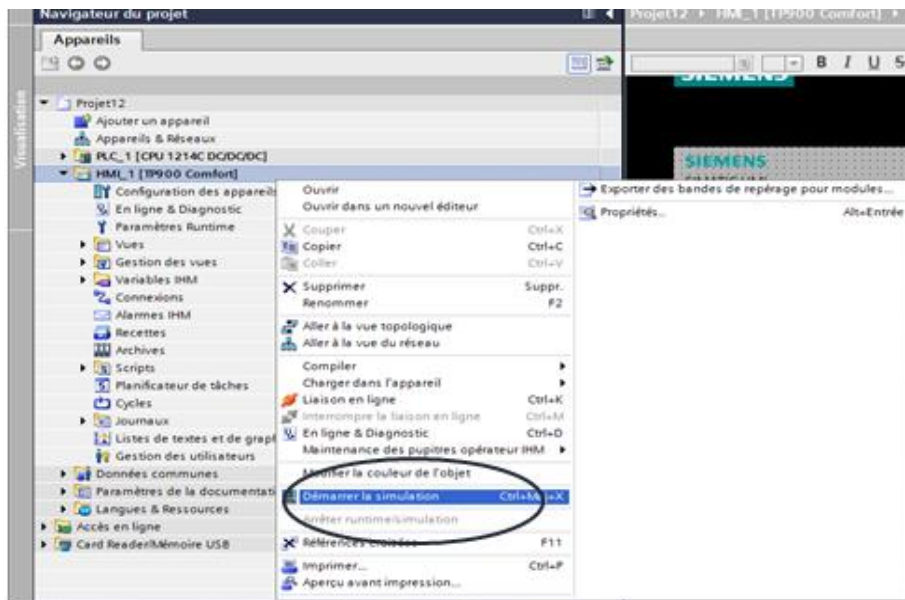


Figure II- 26 Lancement du RUNTIME.

## 9. Conclusion

Ce chapitre regroupe les fonctionnalités essentielles du logiciel de programmation de SIEMENS le TIA Portal, utilisé pour la configuration matériel, la programmation en différents langages, la simulation des programmes ainsi que la création et paramétrage des interfaces HMI pour la visualisation des process.

# **Chapitre III :**

# **Communication et**

# **télégestion**

## 1. Introduction

Avec le développement des systèmes automatisés et de l'électronique (circuits ASICs), la recherche de la baisse des coûts et la nécessité actuelle de pouvoir gérer au mieux la production, c'est à dire : de recevoir les données liées à une application le plus rapidement possible, de consulter, contrôler ou de modifier les paramètres d'une application à distance ; sont apparues de nouvelles technologies de câblage et de communications entre les divers constituants des automatismes.

## 2. définitions primordiales

- **Terrain** : indique un espace délimité géographiquement (usine, atelier, voiture...).
- **Bus** : au sens informatique industriel, conducteur ou ensemble de conducteurs communs à plusieurs circuits permettant l'échange de données entre eux [12].
- **Protocole de communication** : est une spécification de plusieurs règles pour un type de communication particulier. Initialement, on nommait protocole ce qui est utilisé pour communiquer sur une même couche d'abstraction entre deux machines différentes.
- **Réseau** : ensemble d'équipements reliés entre eux pour échanger des informations. Cet ensemble est constitué d'au moins un support de transmission pour l'acheminement des signaux et de protocoles de communication [12].
- **Réseau local** : Un réseau local couvre une zone géographique limitée sur un domaine privé, par opposition aux réseaux publics ou longue distance. Parmi les réseaux locaux, on distingue souvent les réseaux locaux d'entreprise et les réseaux locaux industriels. Ils diffèrent essentiellement par les contraintes d'environnement (temps et sûreté de fonctionnement) et par certains services et protocoles mis en œuvre pour tenir compte des différences de besoins des applications qui les utilisent [12].
- **Réseau local industriel** : il est en première approximation un local utilisé dans une usine ou tout système de production pour connecter diverses machines afin d'assurer la commande, la surveillance, la supervision, la conduite, la maintenance, le suivi de produit, la gestion, en un mot, l'exploitation de l'installation de production [12].
- **Profibus** : (Process Field Bus) est un réseau de terrain ouvert, il se décline en trois variantes de protocoles, répondant chacune à des finalités métiers et applicatives spécifiques [13] :

### 1) Profibus-DP (Decentralized Peripheral) :

Destiné aux applications de type maître-esclave en monomaître pour la gestion des équipements d'entrées-sorties déportées avec des temps d'accès extrêmement courts. Le fonctionnement multimaître est possible.

### 2) Profibus-FMS (Fieldbus Message Specification) :

Destiné aux applications nécessitant l'échange entre maîtres pour la synchronisation d'activités de contrôle-commande, basé sur la messagerie MMS (Manufacturing Message Specification).

### 3) Profibus-PA (Process Automation) :

Destiné aux applications de contrôle de process nécessitant la communication avec des équipements de terrain (capteurs, actionneurs) permettant une télé-alimentation des équipements et un fonctionnement avec sécurité intrinsèque en ambiance explosive.

- **TCP** : Protocole de contrôle de la transmission émetteur-récepteur (absence d'erreur, séquence correcte et complète). TCP fournit un service sûr en mode connecté, une liaison devant être établie entre deux stations avant transmission, puis libérée au terme de l'échange. TCP intègre également des mécanismes de surveillance permanente de la liaison.
- **Ethernet TCP/IP** : Le protocole Ethernet TCP/IP s'est aujourd'hui imposé comme standard utilisé dans l'administration. Il est conçu pour l'échange de données dans un réseau local (LAN) et peut également communiquer avec d'autres réseaux ainsi qu'avec Internet. Pour l'identification des abonnés du réseau, une adresse IP univoque est attribuée à chaque station d'abonné. Le protocole TCP/IP peut échanger des données au-delà des limites de son réseau. Au moyen de commutateurs ou de concentrateurs (hub), il est possible de créer des sous-réseaux (subnet) [14].

## 3. PROFINET

### 3.1. Définition

PROFINET est le nouveau standard ouvert de la communication industrielle sur Ethernet, il facilite la réalisation d'automatismes répartis, l'intégration des équipements de terrain existants et l'exploitation des applications pointues et temps réel du Motion Control (contrôle de position et de synchronisation). Outre l'adoption massive des TI (technologies de l'information), le souci de pérenniser les investissements des industriels (exploitants et propriétaires d'usine, constructeurs de machines et d'installations) est au cœur du concept PROFINET. Il s'agit avant tout d'intégrer les bus de terrain en place, tels PROFIBUS, sans toucher à l'existant [14].

### 3.2. Historique

PROFINET a été créé par PI (PROFIBUS & PROFINET International) - l'organisation des utilisateurs PROFIBUS - et développé par Siemens, Phoenix Contact, Molex et d'autres constructeurs. De par son ouverture et l'utilisation d'un média de communication standard (Ethernet), PROFINET permet l'utilisation de toutes marques de matériel. La première version de ce standard a été publiée en août 2001. La version courante est la version V2.2.

### 3.3. PROFINET et le modèle OSI

Le protocole PROFINET respecte l'organisation en 7 couches du modèle OSI. Les couches 5 et 6 étant inutiles ne sont donc pas utilisées par les équipements PROFINET. Les couches effectivement utilisées dépendent de la nature du trafic. Une station PROFINET, est susceptible de gérer 3 types de trafic :

1. Le trafic IT classique sans rapport avec PROFINET, par ex. accéder à la page web de l'équipement.
2. Le trafic PROFINET non temps réel, par ex. configurer une station ou faire un diagnostic.
3. Le trafic PROFINET temps réel qui concerne les données d'entrées/sorties du procédé.

Pour les deux premiers on utilisera Ethernet, la couche Réseau IP et la couche Transport TCP/UDP. Pour les échanges de données process, on passe directement d'Ethernet (couches 1 et 2) à la couche applicative PROFINET.

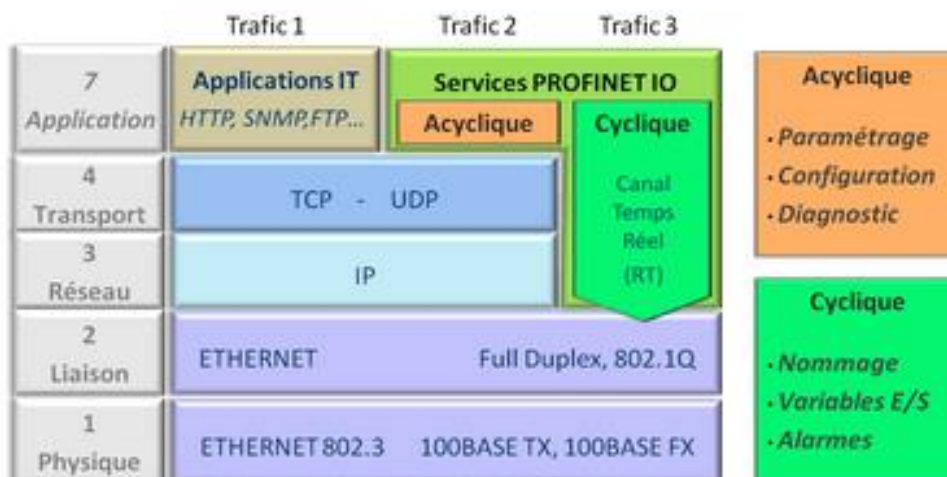


Figure III-2 PROFINET et le modèle OSI.

### **3.4. Caractéristiques de la couche physique**

Le réseau PROFINET étant basé sur Ethernet il respecte les standards correspondants. Les stations supportent des connexions cuivre, fibre optique ou Wi-Fi.

- La version cuivre, la plus répandue, est basée sur la norme 802.3u (Fast Ethernet) qui permet une communication 100 Mb/s en full duplex. L'essentiel des équipements dispose d'une connexion de ce type.
- La fibre optique est essentiellement utilisée dans le cadre de l'infrastructure et sert à interconnecter les switches entre eux.
- Le Wi-Fi est utilisé ponctuellement pour des applications mobiles et/ou difficiles à raccorder physiquement.

### **3.5. Le PROFINET IO et le PROFINET CBA**

Le PROFINET IO est un standard Ethernet industriel ouvert et destiné au monde de l'automatisation. Contrairement au PROFINET CBA (Component Based Automation) qui est destiné aux systèmes distribués, le PROFINET IO se concentre sur l'échange de données entre automates programmables. Le PROFINET CBA fournit un système basé sur DCOM pour l'organisation des systèmes d'automatisation dans les réseaux de dispositifs analogues qui peuvent automatiquement échanger des données à l'aide de relations prédéfinies entre les interfaces des composants d'automatisation.

### **3.6. Définition PROFINET IO**

PROFINET IO est l'évolution du réseau PROFIBUS DP vers une base Ethernet. On y retrouve certains des concepts de base qui ont permis à PROFIBUS de devenir le leader du marché (Simplicité de mise en œuvre, principe de diagnostic, fichier descriptifs...) tout en améliorant les performances et en intégrant les services propres aux technologies IT.

#### **3.6.1. Constituants d'un réseau PROFINET IO**

Le réseau PROFINET IO se compose de 3 constituants :

- **IO Supervisor** : Une station d'ingénierie permettant la configuration et le diagnostic d'un réseau PROFINET IO, typiquement un atelier logiciel sur PC.



- **IO Controller** : IO-contrôleurs sont des dispositifs qui exécutent un programme d'automatisation. Ils effectuent l'échange de données avec les IO-Devices. Il s'agit typiquement d'un automate ou d'un PC.

- **IO Device** : représentent les actionneurs et capteurs distribués connectés à l'IO-Controller via Ethernet typiquement des E/S, variateurs, passerelles, HMI [16].

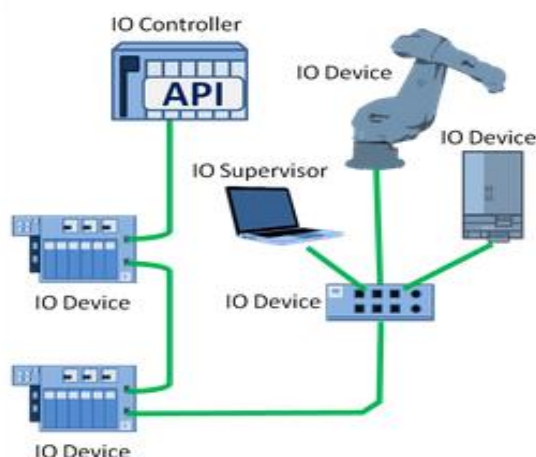


Figure III-2 Les 3 stations de PROFINET.

### 3.6.2. Topologies de PROFINET IO

PROFINET peut être déployé avec les différentes topologies classiques d'Ethernet (Etoile, arborescente, maillé) ou en ligne comme dans les applications bus de terrain conventionnels. Avec ses structures flexibles en ligne, en anneau et en étoile, PROFINET IO s'adapte à l'infrastructure existante des installations, car il est basé sur Ethernet standard, les transmissions filaires et sans fil peut être facilement combinées. Par conséquent, les communications WLAN et Bluetooth peuvent être intégrées de manière transparente dans la solution, y compris pour les données temps réel [16].

### 3.6.3. Echange des données avec le PROFINET IO

PROFINET utilise le mécanisme de Producteur – Consommateur. Les IO Devices produisent les entrées qui sont consommées par l'IO Controller et ce dernier produit les sorties qui sont consommées par les IO Devices. Les données sont transmises de manière autonome sur le réseau en fonction du rythme défini lors de la configuration. PROFINET échelonne la communication sur trois niveaux de performances [16]:

#### a) PROFINET NRT (non temps réel) :

Ce type est utilisé pour les applications où le timing n'est pas critique. Il utilise le protocole Ethernet TCP/IP et aussi UDP/IP pour le transfert de données. Il permet d'obtenir des temps de cycle supérieurs à 100ms.

**b) PROFINET RT (temps réel) :**

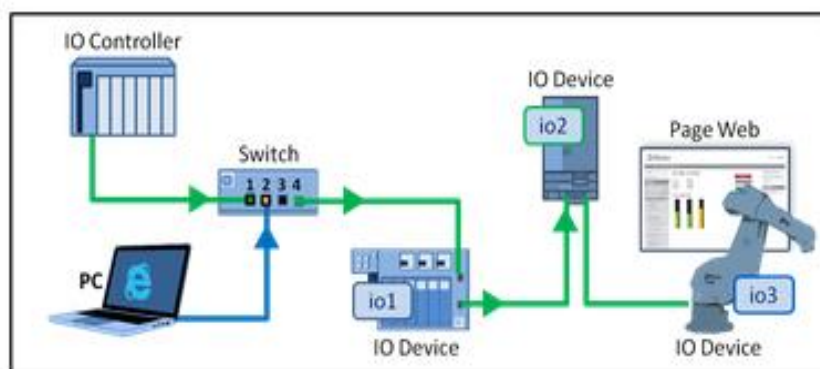
Avec PROFINET RT, les données d'E/S sont échangées directement à l'aide du protocole Ethernet, mais les données de diagnostic et de communication sont toujours transférées via UDP/IP. PROFINET RT permet de créer des applications dont le temps de cycle est supérieur à 10 ms.

**c) PROFINET IRT (temps réel isochrone) :**

Cette communication temps réel isochrone a été développée spécialement pour les applications de motion control (**contrôle du mouvement**). Des temps de cycle inférieurs à 1 ms et des temps de gigue inférieurs à 1  $\mu$ s peuvent être obtenus en cas d'utilisation de matériel spécifique.

### 3.6.4. Priorisation des trames PROFINET

PROFINET utilise le standard Ethernet 802.1q qui définit 8 niveaux de priorité ; 0, moins prioritaire, à 7, plus prioritaire. Les trames de données PROFINET sont émises avec le niveau 6 elles seront donc transmises avant le trafic IT standard. Un réseau comportant des équipements PROFINET est donc susceptible de transporter d'autres données, par exemple pour consulter le serveur Web d'un équipement. Cette ressource que constitue le réseau global est exploitée pour des données process (urgente) et d'autres communications (non urgente). À certain moment, il est donc possible que deux requêtes doivent être acheminées simultanément ce qui est impossible.



*Figure III-3 Priorisation PROFINET.*

Dans l'exemple ci-dessus, l'IO Controller envoie une consigne à destination de IO2, au même moment le PC souhaite consulter la page Web de IO3. Les deux trames arrivent simultanément l'une sur le port 1, l'autre sur le port 2 du switch. Il faut les faire transiter par le port 4 pour qu'elles parviennent à destination.

Grâce à sa haute priorité, la trame PROFINET sera transmise immédiatement alors que la requête Web du PC devra attendre la disponibilité du port

## 4. MODBUS

### 4.1. Définition

MODBUS est un protocole de communication série développé par Modicon en 1979 pour être utilisé avec ses contrôleur logique programmables (PLC).

En termes simples, il s'agit d'une méthode utilisée pour transmettre des informations sur des lignes en série entre des appareils électroniques. L'appareil qui demande l'information s'appelle le MODBUS Master et les périphériques fournissant des informations sont MODBUS Slaves. Dans un réseau MODBUS standard, il existe un Master et jusqu'à 247 Slaves, chacun avec une adresse esclave unique de 1 à 247. Le Master peut également écrire des informations sur les Slaves.

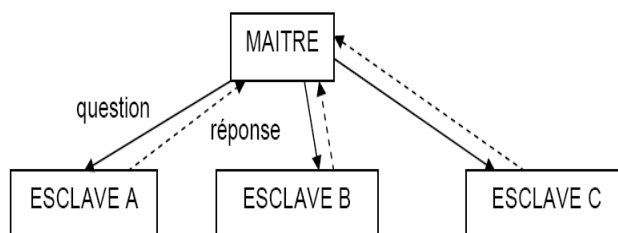


Figure III-4 structure du réseau MODBUS.

- Le maître envoie une **demande** et attend une **réponse**.
- Deux esclaves ne peuvent dialoguer ensemble.
- Le dialogue maître – esclave peut être schématisé sous une forme successive de liaisons point à point.
- Mode de communication : **half-duplex** (2 fils ou 4 fils).
- le maître parle à l'ensemble des esclaves, sans attente de réponse (diffusion générale).

Il ne peut y avoir sur la ligne qu'un seul équipement en train d'émettre. Aucun esclave ne peut envoyer un message sans une demande préalable du maître. Le dialogue direct entre les esclaves est impossible.

## 4.2. Structure des messages dans le MODBUS

Le maître envoie un message constitué de la façon suivante :

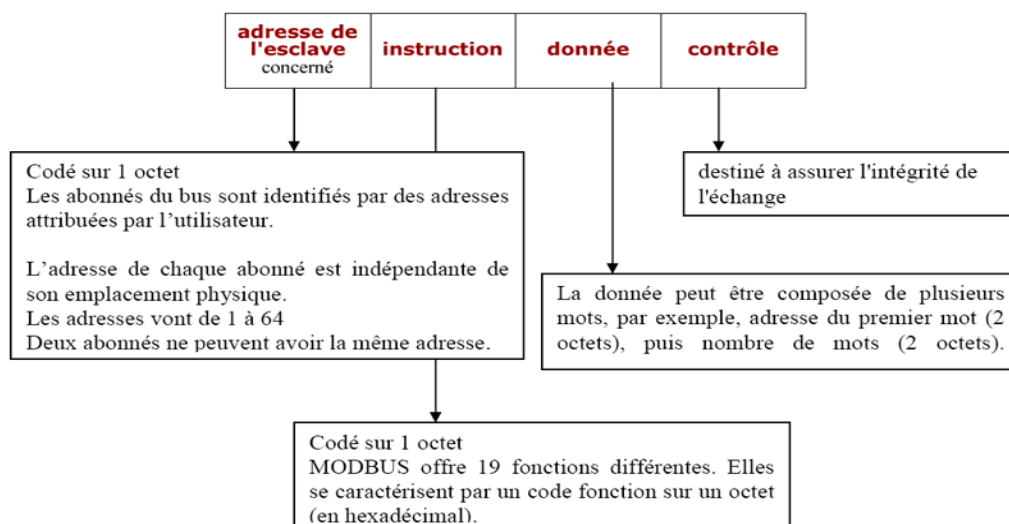


Figure III-5 Structure d'un message dans le protocole MODBUS.

- La détection de fin de message est réalisée sur un **silence** de durée supérieure ou égale à la durée de transmission de 3 octets.
- L'esclave répond par un message du même type que le message question.

Le protocole MODBUS est très utilisé pour les communications des automates programmables industriels et des variateurs de vitesse des moteurs électriques.

## 4.3. Mode de fonctionnement de MODBUS

**4.3.1. En mode RTU (RS232, RS422) :** Il fonctionne sur le mode Maître/Esclave. Seul le maître est actif, les esclaves sont complètement passifs. C'est le maître qui doit lire et écrire dans chaque esclave. Il est constitué de trames contenant le numéro de l'esclave concerné, la fonction à traiter (écriture, lecture), la donnée et le code de vérification d'erreur appelé contrôle de redondance cyclique sur 16 bits ou CRC16.

**4.3.2. En mode TCP (Ethernet) :** Il fonctionne sur le mode Client / Serveur. Les clients sont tous actifs, le serveur est complètement passif. Chaque client lit et écrit dans le serveur, il est constitué de trames contenant la fonction à traiter (écriture, lecture)

et la donnée, l'adresse du serveur concerné est son adresse IP. Le code de vérification d'erreur est inutile en mode TCP, ce mode de transmission comporte déjà un appelé contrôle de redondance cyclique sur 32 bits géré par la carte réseau.

## 5. La télégestion

La télégestion désigne un ensemble de solutions technologiques permettant de piloter à distance des installations autonomes géographiquement dispersées. De quoi, par exemple et de façon basique, permettre à l'exploitant d'un ouvrage d'être alerté en cas de problème technique. C'est toutefois loin d'être le seul atout de ces systèmes, puisqu'ils savent également enregistrer le fonctionnement des équipements surveillés (pompes, vannes, etc.). Il est par exemple possible de suivre en permanence l'état d'un réseau, d'analyser son comportement et d'en optimiser la gestion par l'établissement de bilans périodiques. Objectif : la maintenance préventive, la détection de fuites, la surveillance des consommations et des volumes produits et distribués. Avec à la clé des économies tant en termes d'énergie, que de la ressource elle-même, le tout en limitant les déplacements et les coûts. Progressivement, les technologies de l'information et de la communication (TIC) ont pris une place décisive au sein de ces équipements.

Dans le secteur de l'eau, la télégestion naît, dans les années soixante-dix, pour optimiser le couple station de pompage-réservoir d'eau potable. La télétransmission alors ne sert qu'à déclencher ou stopper le pompage, par radio ou via une liaison filaire. Dans les années quatre-vingt, arrive la téléalarme, autorisant le poste local à prévenir l'exploitant du réseau d'un dysfonctionnement éventuel. L'étape suivante voit la mise en œuvre d'un dispositif permettant de connaître l'état de santé du réseau lui-même, à travers celui de ses équipements, ainsi que leur évolution dans le temps. Cette fonctionnalité est rendue possible par l'enregistrement en local des états des capteurs de l'installation. Complémentairement, des calculs de bilans assistent l'exploitant dans l'analyse du fonctionnement.

Cette approche s'est largement pérennisée depuis, puisqu'en complément des modems RTC (Réseau téléphonique commuté), ont été adoptées des solutions comme le GSM, Data, l'ADSL (*Asymmetric Digital Subscriber Line*) et aujourd'hui le GPRS (*General Packet Radio Service*). Ainsi, de « serveurs Minitel », les équipements de télégestion sont devenus « serveur Web », c'est-à-dire utilisés grâce aux technologies Internet.

## 6. Logiciel de supervision et de télégestion TOPKAPI

TOPKAPI Vision est un superviseur de procédés industriels fonctionnant en environnement Windows, il a été Créé en 1989 par AREAL et commercialisé sous le système d'exploitation DOS, puis en 1991 AREAL a été le premier éditeur français à proposer un logiciel de supervision en environnement WINDOWS.

Topkapi est le logiciel est utilisé dans plusieurs applications comme le bâtiment, l'énergie, l'industrie agro-Alimentaire il est devenu aussi le leader pour la supervision industrielle dans les marchés de l'eau et de l'assainissement.

Très tôt, les besoins particuliers des exploitants de réseaux de distribution d'eau potable, et des réseaux d'eaux usées, ont été pris en compte dans le développement de la solution Topkapi Vision car il permet de créer des interfaces homme-machine, synoptiques, faire un archivage des évènements, établir des rapports de production et garantir la télégestion [15].

### 6.1.Synoptiques graphique

TOPKAPI permet les créations des synoptiques graphiques en garantissant:

- Éditeur intégré avec des fonctions zoom et étendues
- Multifenêtrage et l'ouverture simultanée de plusieurs applications
- Configuration en ligne.
- Orientation objet.
- Bibliothèque d'objets avec héritage.
- Objets avec fenêtre de détail «faceplate».
- Animations couleurs 24 bits (true color).

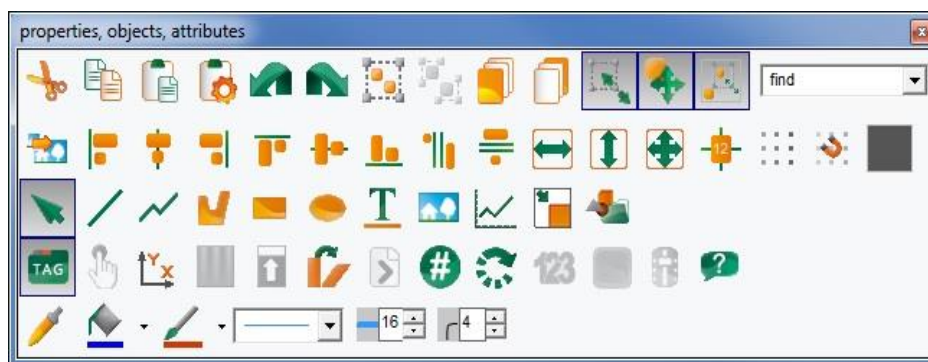


Figure III-6 Bibliothèque de TOPKAPI.

### 6.2.Les courbes

Il est possible dans le logiciel TOPKAPI d'intégrer les courbes aux synoptiques qui peuvent être modifiable en ligne. Ce logiciel garanti un exploitation facile il n'a pas donc une interface a paramétrer.

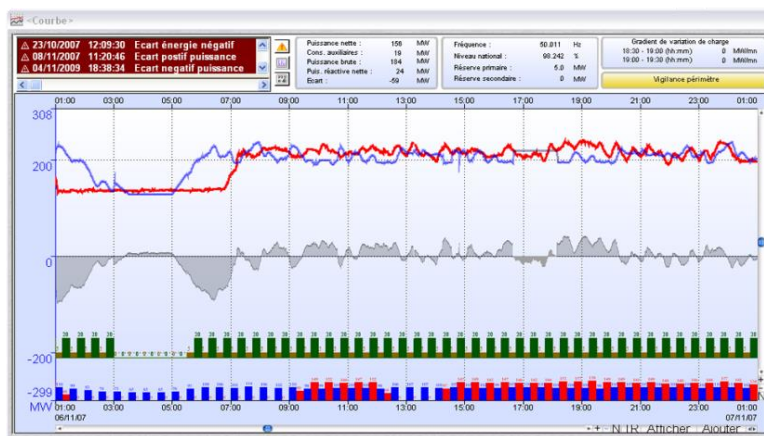


Figure III-7 Vue des courbes dans TOPKAPI.

### 6.3. Alarme

Le TOPKAPI assure la présentation et la création des vues d'alarme, ces vues peuvent être créées et modifiées «on line». Chaque vue alarme indique la présence d'un défaut sur une installation ou plusieurs installation, ce qui est appelé Vues «multipostes» comprenant des données fusionnées issues de plusieurs postes serveurs.

Date	Heure	Ints	Zone	Libellé	Valeur	Date acq	Heure ac
10/02/2004	17:26:00		Départ	Degréleur courbe	Arrêt		
10/02/2004	17:26:12		Pompage	Forage 3 Neully	Marché		
11/02/2004	17:31:09	▲	Départ	Exemple défaut 3 acquité	Défaut	11/05/2004	17:51:38
11/02/2004	17:31:09	▲	Départ	Exemple défaut 2	Défaut	11/05/2004	17:51:38
11/02/2004	17:31:09	▲	Départ	Exemple défaut 3	Défaut		
11/02/2004	17:31:09	▲	Atelier 2	Présence secteur en contact	Non		
11/02/2004	17:31:09		Pompage	Forage 3 Neully	Arrêt		
11/02/2004	17:31:09	▲	Atelier 2	ETAT CONTACT SOL2CBQ	Ouvert		
11/02/2004	17:31:09	▲		03-GFB08 Disjonction des GFB	Défaut	26/02/2004	11:10:09
11/02/2004	17:31:09			Niveau Bache	Normal		
11/02/2004	17:31:09		Atelier 2	Déplacement manuel batis	349		
11/02/2004	17:31:09	▲	Pompage	Marché PI	1		
11/02/2004	17:31:12			03-GFB08 Disjonction des GFB	Normal		

Figure III-8 Vue des alarmes dans TOPKAPI.

### 6.4. Télégestion dans TOPKAPI

Pour les équipements non reliés en permanence (réseau téléphonique ou GSM, Internet, etc.), TOPKAPI traite les données horodatées à la source sans aucun paramétrage spécifique, de façon transparente comme si elles avaient été acquises en temps réel.

L'intégration de la gestion des connexions périodiques, des modems et les capacités d'auto-paramétrage contribuent également à justifier sa réputation de produit de référence pour toutes les applications de télégestion.

## **7. Conclusion**

La communication de l'automate avec son environnement est primordiale pour le bon fonctionnement des processus industriels. On a présenté dans ce chapitre les communications filaire de l'automate S7-1200 (PROFIBUS, PROFINET) et la communication sans fils (la télégestion) en introduisant le logiciel TOPKAPI et ses différentes fonctionnalités, ces deux types de communications seront utilisés dans le chapitre Application.



**Chapitre IV :**  
**Etude descriptive de**  
**la station SP5**

## **1. Introduction**

Les ressources en eau ont toujours été recherchées puis mobilisées pour divers usages concourant au développement socio-économique des populations. Les points d'utilisation sont généralement distants des ressources. Aussi il faut transporter les demandes en eau de leur site naturel aux points d'utilisation ce transport exige de l'énergie.

On essaie toujours de faire véhiculer les eaux gravitairement, si éventuellement la topographie et la nature du terrain le permettent, parfois cette solution devient difficile à cause de certaines contraintes topographiques et géotechniques (exemples : terrains accidentés ou trop plats etc...). Donc on fait appel à des stations de pompage qui sont des installations, y compris des pompes et d'équipements pour le pompage de liquides d'un endroit à l'autre. Notre objective est de faire l'automatisation d'une telle station.

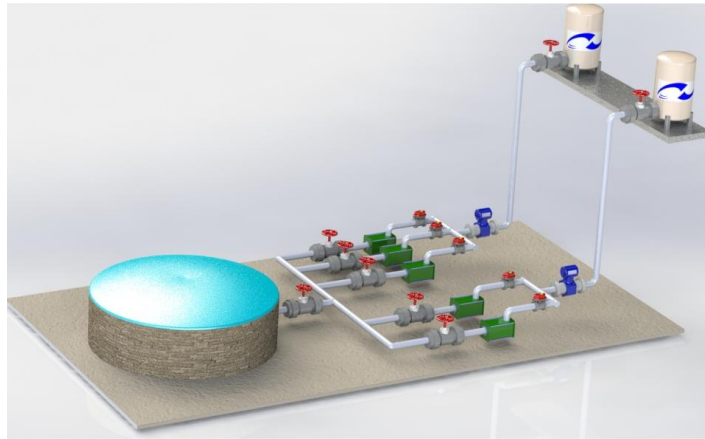
## **2. Description de la station de pompage SP5**

### **2.1. Description générale**

La station de pompage SP5 située à Tessalla El Mardja, fait partie de la chaîne des hauteurs de centre de production de MAZAFRAN qui un centre de production d'eau potable, ce dernier prend en charge la zone Ouest d'Alger. Il est chargé de produire une eau de qualité à travers les différentes ressources disponible de façon optimale et durable. Il est responsable de l'exploitation (transfert et stockage) et de la gestion préventive des équipements de production. Son rôle principal étant l'alimentation des ouvrages hydraulique (réservoirs et châteaux d'eau) des communes qui se trouvent dans sa zone de couverture.

La SP5 assure la surpression nécessaire à l'alimentation du secteur de DEKAKNA et de HADJ YAKOUB, elle est équipée d'un système de pompage à deux refoulements vers deux réservoirs qui sont des ouvrages de stockage d'eau intermédiaires entre les stations de pompes et la distribution d'eau potable.

L'un des réservoirs se situe à DEKAKNA et dessert la localité du même nom. L'autre à HADJ YAKOUB et dessert aussi la localité du même nom.



*Figure IV-1 Schéma représentatif de la station SP5.*

Le refoulement vers le réservoir de HADJ YAKOUB est assuré par deux GEP avec un débit départ de  $190 \text{ m}^3/\text{h}$  et un débit unitaire de  $151,2 \text{ m}^3/\text{h}$ , avec une puissance nominale  $132\text{kW}$  et une tension d'alimentation  $400\text{V}$  pour chaque GEP.



*Figure IV-2 GEP de refoulement ligne HADJ YAKOUB.*

Le refoulement vers le réservoir de DEKAKNA est assuré par trois groupes électropompes (GEP), avec un débit départ de  $160 \text{ m}^3/\text{h}$ , et un débit unitaire de  $151,2 \text{ m}^3/\text{h}$ , avec une puissance nominale  $90\text{kW}$  et une tension d'alimentation de  $400\text{V}$  pour chaque GEP.



*Figure IV-3 GEP de refoulement ligne DEKAKNA.*

Les groupes de pompage aspire depuis une bache d'aspiration qui est un réservoir qui sert à l'alimentation directe en eau de la station de pompage existante sur le site. Elle est composée d'un compartiment de  $150m^3$  et contient une sonde de niveau et deux poires de niveau qui indiquent le niveau Haut et le niveau Bas.



*Figure IV-4 La bache d'aspiration.*

La bache d'aspiration est alimentée par deux forages qui sont des ouvrages de captage vertical permettant l'exploitation de l'eau d'une nappe. Chaque forage est équipé d'une pompe immergée.

## **2.2. Partie électrique**

### **2.2.1. Equipement Electrique MT (transformateur)**

Le transformateur a pour rôle essentiel l'abaissement ou l'augmentation de la tension, le transformateur de la SP5 est un transformateur abaisseur avec une puissance 360 KVA. Il reçoit une tension de de 30kV au côté primaire (à l'entrée) et délivre au côté secondaire (en sortie) une tension de 400V. Il est placé au plus proche des cellules de protection SM6 QM (Combiné interrupteur-fusibles) et sera équipé de relais Bucholz et de thermomètre à 2 seuils.



*Figure IV-5 Transformateur abaisseur.*

### 2.2.2. Equipement Electrique BT

- **Cellule de protection générale CLPG**

L'armoire contient un commutateur volt métrique et ampérométrique, un voltmètre 0-500V, un ampèremètre 0-600A, un relais de protection de courant résiduel VIGIREX et sur l'armoire on observe :

- Des voyants : présence tension 230V ondulée, présence tension 230V non ondulée, présence tension 24Vdc.
- Arrêt d'urgence.

- **Tableau général basse tension TGBT**

Le tableau électrique recouvre les divers organes de commande et de puissance. Il est placé dans le local électrique de la station.

Les armoires sont juxtaposable (extensible), fabriquées à partir d'une feuille d'acier, protégée à l'intérieur et à l'extérieur par une peinture résine polyester époxy et grise.



*Figure IV-6 Vue de l'armoire TGBT.*

- **Armoire départ arrivé**

L'armoire se compose d'une centrale de mesure, un disjoncteur débrochable 1000A, un disjoncteur départ inverseur NSX100, un disjoncteur départ batterie de compensation, disjoncteur départ forage NSX100. Sur l'armoire on observe des voyant de présence des phases L1 L2 L3 et une centrale de mesure de courant et tension.



*Figure IV-7 Armoire départ arrivé.*

▪ **Armoires couloir HADJ YAKOUB et couloir DEKAKNA**

Deux armoires départ pompe une pour alimenter les GEP de HADJ YAKOUB et l'autre pour alimenter GEP de DEKAKNA. Les deux armoires contiennent la partie Puissance ainsi que les Démarreurs progressif Altistart22 de chaque GEP.

Les départs groupes électropompes de HADJ YAKOUB est de 132 kW chacun tandis que les GEP de DEKAKNA est de 90kW chacun.



*Figure IV-8 Armoire couloir HADJ YAKOUB et DEKAKNA*

Sur chaque armoire (couloir DEKAKNA et HADJ YAKOUB) on trouve :

- Un afficheur de centrale de mesure pour chaque pompe.
- Relais PT100.
- Compteur horaire.
- Des Voyants : voyant pompe en service et voyant pompe en défaut.
- Des boutons poussoirs : BP marche pompe, BP reset défaut, BP arrêt pompe.

- Un switch manuel/automatique.

- **Armoire automatisme**

Contient un transformateur 400/220 VAC et un autre 400/24VAC.

Automate programmable LACROIX SOFREL S550 et les composants suivants :

- Écran graphique / molette de dialogue (navigation) pour la consultation et le diagnostic.
- Voyant vert et voyant rouge : montre l'état de l'alimentation et le fonctionnement.
- Port 485 I/O : pour le raccordement des modules d'extensions.
- Alimentation 100/240 V
- Les modules d'extensions entrées/sorties.



*Figure IV-9 vue extérieure et intérieure de l'armoire d'automatisme.*

- **Armoires batteries de compensation**

L'armoire batterie de condensateur est constituée de condensateur, de contacteurs spécifiques pour commander les condensateurs qui sont utilisées pour limiter le transport de puissance réactive et aussi pour la stabilisation de réseau.



**Figure IV-10 Armoire batterie de compensation.**

### ▪ Alimentation secourue

La station SP5 TEM est équipée d'une alimentation secourue par un onduleur et un groupe électrogène 50 kVa. Ils permettront de secourir les appareillages suivants :

Automate programmable, terminal de dialogue, poste de télétransmission, débitmètres, capteurs de niveau analogique, capteurs de pression analogique, indicateurs numériques des débitmètres, gyrophare.



*Figure IV-11 Groupe électrogène.*

### ▪ Armoire auxiliaire

Elle contient un répartiteur tension secourue et un disjoncteur arrivé inverseur, elle se compose de la partie départs auxiliaires, tel que:

- L'alimentation et l'éclairage de la salle de contrôle et la salle machine.
- Eléments chauffants et ventilateurs pour les groupes électropompes.
- Départs vers les coffrets auxiliaires, etc.

## 2.3. Instrumentation

### 2.3.1. Actionneurs

#### a- Le groupe électropompe

Un groupe électropompe (GEP) est un actionneur composé d'un ensemble d'une pompe hydraulique centrifuge et d'un moteur électrique asynchrone qui l'entraîne.

Pour le couloir de DEKAKNA les 3 moteurs utilisés sont de type **Seipee GM 280M** c'est des moteurs asynchrones triphasé à 4 pôles qui entraîne des pompes centrifuges qui sont de types **Caprari PM125H/5A**. Ils se caractérisent par :



Caractéristiques moteur		Caractéristiques pompe	
Constructeur	SEIPEE	Constructeur	CAPRARI
Modèle	GM 280M	Modèle	PM125H/5A
Tension d'alimentation	400 V	Débit	151,2 m <sup>3</sup> /h
Puissance moteur	90 kW	Nombres d'étages	5
Couple nominal	581 Nm	HMT	120 m
Courant nominal	159 A	Diamètre d'aspiration	300 mm
Vitesse nominale	1480 tr/min	Diamètre de refoulement	320 mm

**Tableau IV-1 Caractéristiques des GEP de la ligne DEKAKNA.**

Pour le couloir de HADJ YAKOUB les 2 moteurs utilisés sont de type **SEIPEE GM 315Ma** c'est des moteurs asynchrones triphasé à 4 pôles qui entraîne des pompes centrifuges sont de types **CAPRARI PM125H/8A**. Ils se caractérisent par :

Caractéristiques moteur		Caractéristiques pompe	
Constructeur	SEIPEE	Constructeur	CAPRARI
Modèle	GM 315Ma	Modèle	PM125H/8A
Tension d'alimentation	400 V	Débit	151,2 m <sup>3</sup> /h
Puissance moteur	132kW	Nombres d'étages	8
Couple nominal	852 Nm	HMT	185 m
Courant nominal	228 A	Diamètre d'aspiration	300 mm
Vitesse nominale	1480 tr/min	Diamètre de refoulement	320 mm

**Tableau IV-2 Caractéristiques des GEP de la**

**ligne HADJ YAKOUB.**

**b- Démarreur progressif**

Les démarreurs utilisés dans la station sont de type Altistar22 qui est un démarreur-ralentisseur de Schneider Electric qui assure le démarrage et l'arrêt progressifs en tension et en couple des moteurs asynchrones triphasés à cage, pour des puissances comprises entre 4 et 400Kw.

LIGNE	DEKAKNA	HADJ YAKOUB
Constructeur	Schneider Electric	Schneider Electric
Modèle	ATS22C21Q	ATS22C25Q
Tension d'alimentation	230..440 V	230..440 V
Puissance moteur	110 kW	132 kW
Fréquence d'alimentation	50 Hz	50 Hz

*Tableau IV-3 Caractéristiques des démarreurs progressifs.*

### c- Compresseur d'air

Un compresseurs d'air est un actionneur qui aspire le gaz de basse pression et refoule à haute pression. Il est utilisé pour garder un niveau de 1/3 d'air dans l'anti béliet.

La station dispose deux anti-béliet, un sur chaque conduite de refoulement, chaque anti béliet est équipé d'un compresseur d'air.



*Figure IV-12 Anti-béliet ligne HADJ YAKOUB.*

### d- Pompe d'épuisement

Une pompe d'épuisement (également appelée pompe refoulente) a pour rôle l'évacuation d'eau envahissant un point bas : cave, sous-sol, fond de terrain. Le dispositif est constitué de la pompe en question et d'un long tuyau d'évacuation afin d'éjecter l'eau puisée assez loin pour qu'elle ne revienne pas inonder la zone protégée par la pompe, ainsi d'un flotteur réglé de façon à commander la mise en route de la pompe (via l'automate) dès que l'eau atteint un seuil bien déterminé, le dispositif se trouve dans le caniveau de la salle machine.

### e- Vanne manuelle

Une vanne est un dispositif destiné à contrôler (stopper ou modifier) le débit d'un fluide liquide, gazeux, pulvérulent ou multiphasique, en milieu libre (canal) ou en milieu fermé (canalisation). La station SP5 dispose de 10 vannes manuelles, 5 vannes sur la conduite d'aspiration et 5 pour le refoulement.

### 2.3.2. Capteurs

#### a- Capteur de débit (Débitmètre électromagnétique)

Le débitmètre électromagnétique est un capteur de débit limité aux liquides conducteurs. Il est constitué d'un tube en matériaux non magnétique, revêtu intérieurement d'un revêtement isolant. Deux bobines d'induction sont disposées de part et d'autre de la conduite. Elles créent un champ magnétique alternatif. Quand un liquide conducteur s'écoule perpendiculairement à travers le champ magnétique, une différence de potentiel électrique est créée au sein du liquide. Cette différence de potentiel, captée à l'aide de deux électrodes fixés perpendiculairement sur les côtés de la conduite permet, par le calcul, d'en déduire la vitesse puis le débit du fluide. La vitesse d'écoulement du liquide et son débit sont proportionnels.

Les débitmètres utilisés dans la station SP5 sont de type **Tecofi** MUT2200ELPN10, ils délivrent un signal 4-20 mA, ils sont placés sur les conduites de refoulement principaux possédant un diamètre de 200 mm leurs convertisseurs/afficheurs sont fixés dans la salle machine [16].



*Figure IV-13 débitmètre électromagnétique et son convertisseur.*

#### b- Capteur de température

##### ▪ Sonde PTC

Les sondes PTCs (Positive Température Coefficient) appelées ainsi Thermistances sont des détecteurs de température. Elles sont constituées d'une résistance dont la valeur ( $\Omega$ ) augmente rapidement et linéairement avec la température et cela une fois qu'un seuil de cette dernière est dépassé [16].

Les sondes PTC utilisées sont de type A99BC-25C Silicon Temperature Sensors d'une plage de mesure de 0 à 100°C, elles se trouvent au niveau des moteurs plus exacts sur les enroulements, elles sont utilisées comme des dispositifs de surveillance de température interne et cela pour protéger le moteur de problème de ventilation ou de température ambiante excessive, on trouve 10 Sondes PTC deux dans chaque moteur de la SP5.

- **Sonde Pt100**

La sonde Pt100 (Platine 100) est un capteur de température dit résistive parce qu'elle est constituée d'une résistance en Platine donc la valeur initiale est de 100  $\Omega$  pour une température correspondant à 0 °C ainsi la résistance change au fur et à mesure du changement de la température. Le rapport Température/Résistance n'est pas constant mais il est assuré par l'électronique de l'appareil de mesure [16].

Les sondes Pt100 utilisées dans la station est de type Nom : A99BC-25C Silicon Temperature Sensors pour la surveillance de la température du palier des pompes. Chaque pompe contient deux Sondes, une aux extrémités de chaque palier. En total on trouve 10 capteurs.

- c- **Capteur de niveau**

- **Électrode de niveau**

Une électrode de niveau (ou bougie de niveau), est un détecteur de niveau pour liquide conducteur. Elle est constituée d'un tube de protection non conducteur ouvert à une de ses extrémités, à l'intérieur se trouve une électrode fixée à l'autre extrémité et connectée par un câble électrique [16].

Pour assurer la détection du niveau, la station est équipée de 3 électrodes de niveau. Ces 3 détecteurs sont placés horizontalement aux niveaux seuils désirés. Une faible tension est envoyée dans la première électrode dite de référence (de masse), lorsque le liquide est en contact avec cette dernière et une autre positionnée à un niveau supérieur (NB) la tension passe par le liquide conducteur ainsi ferme le circuit, l'immersion du troisième détecteur placé plus haut (NH) établit un contact entre l'électrode en question et celui de référence. Ces 3 bougies sont connectées à un relai électrique qui donne à sa sortie deux signaux de type TOR. Ces détecteurs sont équipés aux anti-béliers pour assurer le niveau 2/3 eau, donc au total il existe 6 électrodes de niveau reliées à 2 relais électriques.

- **Le flotteur**

Le flotteur, est un détecteur de niveau pour liquides, il est constitué d'un boîtier non conducteur contenant une bille en acier qui réalise un contact électrique en fonction de la position du détecteur dans le plan vertical.

Le détecteur est suspendu à une distance bien définie au-dessus du liquide, en absence de ce dernier le flotteur est suspendu dans le plan vertical au-dessus du liquide grâce à son câble. Quand le niveau monte, le capteur est poussé vers le haut et le contact s'établit [16].

Le flotteur est utilisé en mesure de sécurité comme indicateur de plein ou vide de la bache d'eau de la station SP5, il délivre un signal TOR et commande l'arrêt des GEP en cas de débordement d'eau ou de défaillance de la sonde de niveau.

#### ▪ Sonde de niveau

La sonde de niveau, est un capteur de pression piézo-résistif immergé. Il est constitué d'une membrane soumise sur l'une de ses faces à la pression au fond du réservoir (P) et sur l'autre à la pression de la surface (Po). La pression exerce une contrainte sur la membrane créant une variation mesurable de résistance. Ainsi la mesure du niveau est obtenue par la formule:

$$H = \frac{P - P_0}{\mu g}$$

Où  $g$  : L'accélération de la pesanteur et  $\mu$  : La masse volumique du liquide.

C'est l'électronique embarquée dans le capteur qui effectue ce calcul. Une fois H calculée, le capteur joue le rôle de régulateur de courant dans la " boucle de courant". Le courant de la boucle est alors l'image de la hauteur d'eau [16].

La station est équipée d'une sonde de niveau placé dans la bache d'eau, sa plage de mesure est de 0-6m et délivre un signal 4-2-mA.

#### d- Capteur de pression (Sonde de pression)

Les sondes de pression, sont des capteurs de pression piézo-résistifs, tous comme la sonde de niveau précité, elles sont constituées d'une membrane sensible créant une variation mesurable de résistance l'hors d'un changement de pression [16].

L'SP5 contient 6 sondes de pression ; 4 sondes, une sur chaque une des conduites de refoulement et d'aspiration principaux ainsi une sonde pour chaque anti-bélier, la plage de mesure de ces sonde est de 0-25 bars et délivre un signal de 4-20 mA.



Figure IV-14 sonde de pression.

### 3. Cahier des charges de la station

Ce cahier de charge décrit les conditions de la mise en marche du système, et les commandes nécessaires afin de garantir un bon fonctionnement de la station de pompage. On peut le traduire comme suit :

1. Le fonctionnement de la station est géré par l'automate, cependant l'opérateur dispose des boutons et des commutateurs placés sur les armoires pour intervenir durant le Process en cas de besoin.
2. L'Automate Siemens S7-1200 aura pour rôle essentiel de gérer les actionneurs tels que les groupes électropompe, les vannes motorisées. Il assurera la sécurité de l'installation afin d'éviter toute fausse manipulation de la part de l'opérateur, ou d'autres défauts qu'ils soient physiques ou programmés, il permettra aussi la communication avec la supervision (HMI) et la salle de contrôle à distance.
3. Pour chaque ligne (HADJ YAKOUB et DEKAKNA), les groupes seront asservis comme suit:

- **La commande de démarrage**

La commande de démarrage des GEP est activée si un niveau bas dans la cuve alimentées (réservoir Hadj YACOUB ou DEKAKNA) ou un niveau haut dans la bêche d'aspiration ont été détecté. La mise en marche se fait après la vérification des conditions suivantes :

- Absences des défauts électriques générales.
- La bêche d'aspiration non vide.
- La vanne d'aspiration individuelle en position ouverte et la vanne de refoulement en position fermée.
- Le groupe de pompage en question disponible (GEP et vanne).

- **La gestion des GEP**

En cas de demande de pompage :

- Pour la ligne de HADJ YAKOUB un seul GEP sur deux doit démarrer.
- Pour la ligne de DEKAKNA deux GEP sur trois doivent démarrer.

Le choix de la pompe à démarrer se base sur deux critères la disponibilité et le temps d'arrêt le plus grand.

En cas de défaut sur un GEP en marche, un basculement vers un autre GEP doit être fait.

- **Description de la disponibilité**

Les conditions de disponibilité d'un GEP sont l'absence de défauts physiques électriques ou programmés sur la pompe et sur la vanne et l'activation du MODE AUTO.

- **La séquence de démarrage**

Sur ordre de marche et si la vanne de refoulement est en position ouverte on lance tout d'abord la commande de sa fermeture, si un fin de course fermeture est détecté on lance après la commande de démarrage de la pompe. Après avoir reçu le retour de marche de la pompe on lance la commande d'ouverture de la vanne jusqu'à sa position ouverte.

On appelle cette procédure « le démarrage à vanne fermée » et cela pour éviter toutes éventuelles surchauffes des pompes. Le démarrage d'une pompe passe par régime transitoire court (ordre de quelques secondes) et c'est le temps nécessaire pour atteindre sa vitesse nominale. Une fois que la pompe atteint son régime permanent (c'est-à-dire que l'appel de courant a disparu et le courant s'est stabilisé à sa valeur nominale), l'automate actionnera alors l'ouverture de la vanne jusqu'à sa position ouverte et le groupe de pompage est considéré en marche.

- **La commande d'arrêt**

La mise à l'arrêt des GEP se fait en cas de :

- Détection de niveau haut dans la cuve alimentée (réservoir Hadj YAKOUB ou DEKAKNA) ou de niveau bas dans la bache d'aspiration. Tous les GEP doivent être arrêtés successivement et le choix du GEP à arrêter en premier se base sur le temps de marche plus grand.
- Défaut au niveau du GEP en marche, ce dernier est mis à l'arrêt.
- Défaut au niveau de la station tous les GEP doivent être arrêtés simultanément.

- **La séquence d'arrêt**

Suite à ordre d'arrêt on lance la commande fermeture de la vanne jusqu'à sa position fermée ensuite on arrête la pompe. Le GEP dans ce cas est considéré en arrêt.

- **Condition de bon fonctionnement**

- Les pompes ne doivent jamais démarrer ou s'arrêter au même temps.
- Le démarrage et l'arrêt des pompes se fait en cascade.
- La permutation se fait en cas d'indisponibilité.

- **Gestion des défauts**

On appelle défaut, un événement dont l'origine est une anomalie physique et on définit l'alarme comme une représentation visuelle du défaut, et c'est l'information issue du défaut, elle est transmise à l'opérateur par l'intermédiaire du terminal HMI. On distingue deux types de défauts :

- **Défaut physique (Défaut électrique)**

C'est un défaut câblé qui regroupe les dysfonctionnements liés à la chaîne de commande du moteur : (défaut thermique). Il est généré quel que soit le mode de marche de l'actionneur (manuel ou automatique). En principe ce défaut est traité par les armoires électriques. Au niveau programme l'équipement concerné passe en position de sécurité (arrêt dans la plupart des cas). Une alarme est immédiatement générée.

- **Défaut programmé (Défaut discordance)**

Ce défaut a pour objet de détecter les incohérences qui peuvent se produire entre l'ordre émis par l'automate et le retour d'état réel de l'actionneur. Lorsque l'automate donne un ordre à un actionneur (exemple : marche/ arrêt du GEP ou ouverture/fermeture de la vanne), une temporisation est lancée, si à la fin de la temporisation le système n'a pas reçu le retour de l'information attendu (retour marche pour le GEP et fin de course pour la vanne), alors celui-ci est déclaré en défaut discordance.

Ce défaut provoque immédiatement le passage en position de sécurité du moteur. Une alarme est immédiatement générée, lorsque l'état de l'automate et celui du site sont de nouveau en concordance, l'actionneur reste en position de sécurité tant que l'opérateur n'a pas effectué une procédure d'acquiescement.

#### **4. Conclusion :**

Ce chapitre contient une description générale de la station SP5 (fonctionnement et instrumentation) ainsi que description du cahier des charges élaboré selon l'exigence de l'entreprise d'accueil.

La compréhension du fonctionnement de la station est l'étape primordiale pour l'édition et le développement d'un programme pour sa gestion et supervision.



# **Chapitre V: Programmation et télégestion**

## **1. Introduction**

Dans ce chapitre on traduira le cahier de charge qu'on a décrit dans le chapitre précédent. En premier lieu on décrit les différents constituants du programme (FB, FC, DB), les vues créées pour la visualisation du process avec le TIA Portal, la configuration de la communication à distance et la télégestion avec le TOPKAPI à l'aide des routeurs 3G (ERICSSON). Finalement le chargement du programme dans l'API et l'HMI

## **2. Objectif du travail**

- 1) Remplacement de l'automate LACROIX SOFREL par un automate SIEMENS S7-1200.
- 2) développement du programme avec le logiciel TIA PORTAL en introduisant de nombreuses améliorations par rapport à ce qui existe sur le site.
- 3) développement d'une application de supervision avec le logiciel WINCC ADVANCED.
- 4) l'introduction et la configuration des routeurs 3G au lieu de la communication GSM utilisée et cela pour la récupération des données à partir des sites de refoulement ainsi que les forages, et aussi pour la télégestion.
- 5) la télégestion à l'aide du logiciel TOPKAPI.
- 6) Utilisation des vannes motorisées au lieu des vannes manuelles pour les conduites de refoulement de chaque pompe pour que le processus soit gérable automatiquement voir plus sécurisé, car le démarrage à vanne de refoulement fermée permet de diminuer la puissance nécessaire ainsi que le pic de courant lors de démarrage ce qui permet de remplir progressivement les conduites, en chasser l'air, limiter le débit et éviter ainsi la cavitation. chaque vanne est pilotée par l'intermédiaire d'un moteur électrique qui, en fonction des impulsions de commande permet l'ouverture ou la fermeture de la vanne. Ces deux actions sont détectées par deux fins de course (fin de course ouverture et fin de course fermeture) qui permettent à l'électrovanne un fonctionnement de type TOR. Il faut prévoir 5 électrovannes pour chaque GEP sur chaque conduite de refoulement.

## **3. Configuration du matériel**

Dans notre projet on a utilisé la CPU 1214c DC/DC/RELAY, 5 modules de 16 entrées TOR et un module de 16 sorties relais. Les caractéristiques du CPU et des modules entrée/sortie sont détaillée dans le chapitre API.

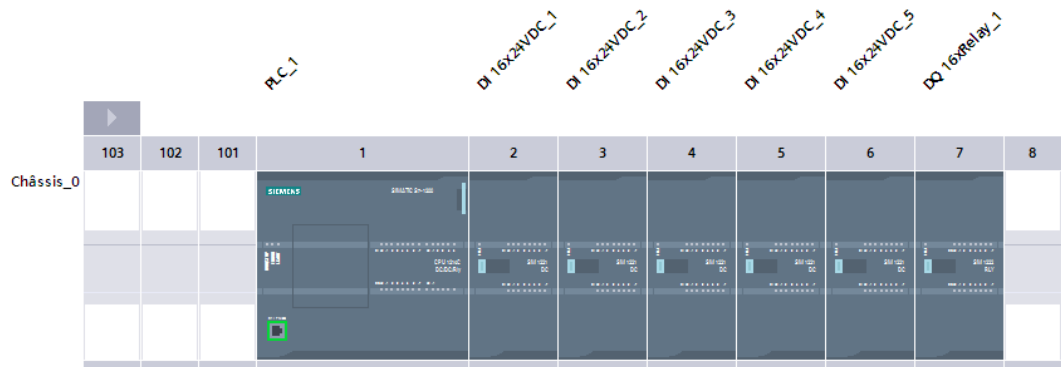


Figure V-1 Configuration du matériel.

## 4. Développement du programme

### 4.1. Description des FB

#### 4.1.1. Calcul du temps

Ce bloc reçoit deux signaux en entrée et envois 3 signaux en sortie.

**Les entrées :** Entrée, Reset.

**-Les sorties :** Sec, Min, Heure.

Le but de ce FB écrit en SCL est de calculer le temps (temps de marche et temps d'arrêt). L'entrée de déclenchement du calcul du temps de marche des GEP est le « retour de marche», en cas de disparition de ce dernier le temps de marche est mis à zéro. Pour le calcul de temps d'arrêt du GEP l'entrée de déclenchement est l'absence du retour de marche, et la remise à zéro est déclenché par la présence de ce dernier.

#### Le programme

```

1 // les entrees :
2 //entree: active le calcul du temps.
3 //reset :remise a zero du calcul .
4 // les sorties:
5 //sec :temps en secondes
6 //min:temps en minutes
7 //H: temps en heures
8 #R_TRIG Instance(CLK:="Clock_1Hz"AND #entree,Q=>#tmp);
9 IF #tmp THEN
10     #sec := #sec + 1;
11 END_IF;
12 IF #sec = 60 THEN
13     #min := #min + 1;
14     #sec := 0;
15 END_IF;
16 IF #min = 60 THEN
17     #H := #H + 1;
18     #min := 0;
19 END_IF;
20 IF #reset THEN
21     #sec := 0;
22     #min := 0;
23     #H := 0;
24 END_IF;

```

### 4.1.2. Défaut vanne

Ce FB reçoit 5 signaux en entrée et envoie 4 signaux en sortie.

#### Les entrées :

**Fermer** : c'est la commande «fermer la vanne » ce signal provient d'un autre programme.

**Ouvrir** : c'est la commande «ouvrir la vanne » ce signal provient d'un autre programme.

**Fco** : fin de course ouverture, ce signal est généré par la vanne.

**Fcf** : fin de course fermeture, ce signal est généré par la vanne.

**Reset** : l'entrée d'acquiescement des défauts.

#### Les sorties :

**disco**: Discordance ouverture.

**cmdo** : Commande d'ouverture de la vanne (signal envoyé à la vanne)

**cmdf** : Commande de fermeture de la vanne (signal envoyé à la vanne)

**discf** : Discordance fermeture

Le but de ce FB écrit en SCL est de surveiller l'ouverture et la fermeture de la vanne motorisée et il doit assurer :

- L'envoi des signaux d'ouverture et de fermeture de la vanne en cas d'absence de défaut.
- L'envoi des signaux discordance d'ouverture et de fermeture :
- Le défaut de discordance ouverture est généré lorsque on envoie la commande «ouvrir la vanne » et qu'on ne reçoit pas le signal « fco » indiquant que la vanne est complètement ouverte

et cela après un certain temps (la vanne prend un temps pour s'ouvrir cela est due à son inertie). Pour des raisons de sécurité ce temps sera une variable locale et non pas une entrée du bloc. De même le défaut discordance fermeture est généré lorsqu'on envoie la commande «fermer la vanne » et qu'on ne reçoit pas le signal « fcf » indiquant que la vanne est complètement fermée et cela après un certain temps.

- L'acquiescement des défauts : les défauts de discordance ouverture et fermeture seront acquiescés lorsqu'on appuie sur le bouton «reset ».

## Le programme

```

1 //ouverture de la vanne
2 IF #ouvrir THEN
3
4 #cmdo := 1; // après la reception de l'ordre d'ouverture on envoie la commande d'ouverture à la vanne
5 #cmdoo := 1; ;
6 END_IF;
7
8 #IEC_Timer_0_Instance_4(IN:=#cmdoo, PT:=#c,Q=>#q_1);
9
10 IF #q_1 AND NOT #fco THEN
11 #disco := 1; // si la temporisation est terminée et la vanne n'est pas ouverte alors on arrête la commande d'ouverture
12 #predisc o" := 1; // et on envoie un signal qui indique qu'il y a un défaut de discordance lors de l'ouverture
13 END_IF;
14
15 IF #fco OR #predisc o" OR #fermer THEN
16 #cmdo := 0;
17 END_IF;
18
19 //fermeture de la vanne
20 IF #fermer THEN
21
22 #cmdf := 1; // après la reception de l'ordre de fermeture on envoie la commande de fermeture à la vanne
23 #cmdff := 1;
24
25 #IEC_Timer_0_Instance_9(IN:=#cmdff,PT:=#c,Q=>#q2);
26
27 IF #q2 AND NOT #fcf THEN
28 #discf := 1; // si la temporisation est terminée et la vanne n'est pas fermée alors on arrête la commande de fermeture
29 #predisc f" := 1; // et on envoie un signal qui indique qu'il y a un défaut de discordance de fermeture
30
31
32 END_IF;
33 IF #fcf OR #predisc f" OR #ouvrir THEN
34 #cmdf := 0;
35 ;
36 END_IF;
37
38 //reset des défauts
39 IF #reset THEN
40 #disco := 0; // si on appuie sur le bouton reset le défaut discordance ouverture sera désactivé
41 END_IF;
42 IF #reset THEN
43 #discf := 0; // si on appuie sur le bouton reset le défaut discordance fermeture sera désactivé
44 END_IF;

```

### 4.1.3. Défaut discordance pompe

Ce FB reçoit 4 signaux en entrée et envoie un seul signal en sortie.

a)-Les entrées :

**Commande:** commande de démarrage du GEP

**Fcf:** fin de course fermeture de la vanne motorisée de refoulement

**Retour de marche:** un signal venant de démarreur progressif indiquant que la pompe a démarré et que la vitesse nominale est atteinte

**Reset:** pour l'acquiescement du défaut

**b)-Les sorties :**

**Défaut de discordance:** signal indiquant qu'il y a un défaut de discordance au niveau de la pompe.

Le but de ce FB écrit en SCL est de surveiller le démarrage de la pompe, il doit assurer :

- La génération de signal défaut de discordance dans le cas où une commande de démarrage de la pompe a été lancée et la vanne est complètement fermée « fcf » et qu'après une certaine temporisation le bloc ne reçoit pas le retour de marche.

- L'acquiescement des défauts : le défaut de discordance de la pompe sera acquiescé lorsqu'on appuie sur le bouton «reset ».

**Le programme**

```

1 //les entrées:
2 //commande
3 //fin de course fermeture
4 //retour de marche
5 //reset:acquiescement de défaut
6 //sortie:
7 //défaut discordance
8
9
10 □#IEC_Timer_0_Instance(IN:=#commande AND #"fin de course fermeture",
11 | PT:=#"temp de disc",Q=>#predisco);
12
13 □IF #predisco AND NOT #"retour-marche" THEN
14 |   #"défaut discordance" := 1; // si la temporisation est terminée et la pompe n'a pas démarré on envoie
15 |   END_IF; // un signal qui indique qu'il y a un défaut de discordance démarrage
16
17 □IF #reset THEN
18 |   #"défaut discordance" := 0 // si on appuie sur le bouton reset le défaut sera désactivé
19 |   ;
20 |   END_IF;

```

#### 4.1.4. Défaut température pompe

Ce FB reçoit 7 signaux en entrées et envoie 2 signaux en sortie.

**a)-Les entrées :**

**Température b1 et Température b2 :** Températures des bobinages du moteur.

**Température p1 et Température p2 :** Températures des paliers de la pompe.

**Seuil b et Seuil p :** Température seuil de bobinage et Température seuil des paliers de la pompe.

**Reset:** pour l'acquiescement du défaut.

**b)-Les sorties : Défaut temp b et Défaut temp** indiquent qu'il y'ait un défaut de température au niveau de bobinage ou au niveau de palier de la pompe.

Le but de ce FB écrit en SCL est la détection des défauts au niveau du GEP et il doit assurer :

- La génération d'un signal indiquant un défaut de température au niveau du bobinage ou du palier et cela après la comparaison des valeurs des capteurs de température avec celles des seuils.

- Si après un défaut, la température reprend sa valeur normale et qu'on appuie sur le bouton reset le défaut sera acquitté.

## Le programme

```

1 //les entrées:
2 //temperature b1 ,temperature b2 : temperature bobinage let 2
3 //temperature p1,temperature p2 : temperature des deux palier
4 //seuil b :seuil de temperature de bobinage
5 //seuil p:seuil de temperature des paliers
6 //reset: acquittement des defauts
7 //les sorties:
8 //defaut temp b: defaut temperature bobinage
9 //defaut temp p: defaut temperature palier
10 IF ("temperature b1">#"seuil b")OR ("temperature b2">#"seuil b") THEN
11     #"defaut temp b" := 1;// si les temperatures de bobinage dépassent le seuil alors on envoie
12         // un signal de defaut temperature bobinage
13
14 ELSIF #reset THEN //si on appuie sur le reset et que la temperature de bobinage ne dépasse pas le seuil
15     #"defaut temp b" := 0; //alors on désactive le defaut temperature de bobinage
16 END_IF;
17
18 IF ("temperature p1" > #"seuil p") OR ("temperature p2" > #"seuil p") THEN
19     #"defaut temp p" := 1; // si les temperatures des paliers dépassent le seuil alors on envoie
20         // un signal de defaut temperature paliers
21 ELSIF #reset THEN
22     #"defaut temp p" := 0; //si on appuie sur le reset et que la temperature des paliers
23     //ne dépasse pas le seuil alors on désactive le defaut temperature paliers
24 END_IF;

```

### 4.1.5. Demande pompage

Ce FB reçoit 4 signaux en entrée et envoie un seul signal en sortie

a)-Les entrées :

**NB bache** et **NH bache** : indique le niveau bas et le niveau haut au niveau de la bache.

**NB réservoir** et **NH réservoir** : indique le niveau bas et le niveau haut au niveau du réservoir.

b)-Les sorties :

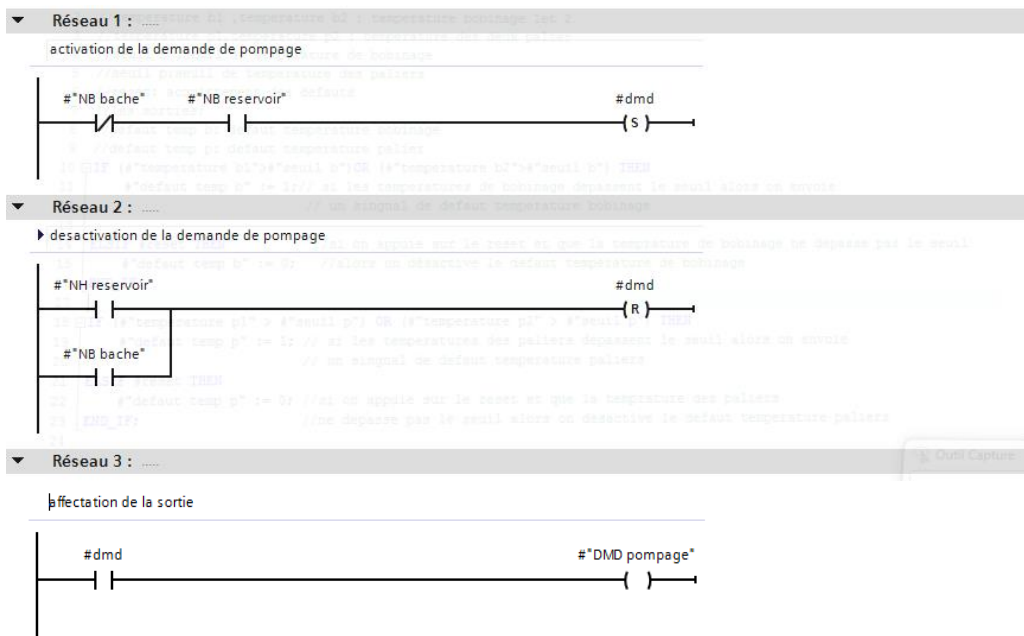
**DMD pompage** : demande pompage

Le but de ce FB écrit en langage CONT est l'activation et la désactivation de la demande de pompage et cela comme suit :

- Si le niveau bas est détecté dans le réservoir et que la bache n'est pas vide alors la demande de pompage sera activée.

- La désactivation de la demande de pompage est assurée par l'atteinte du niveau haut dans le réservoir ou bien le niveau bas dans la bache d'aspiration.

### Le programme



### 4.1.6. Disponibilité

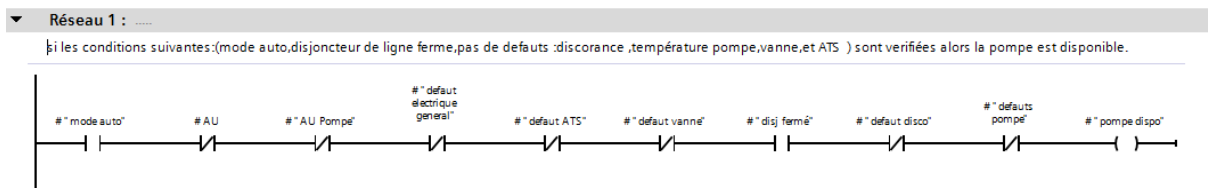
Ce FB reçoit 9 signaux en entrée et envoie un seul signal en sortie

a)-Les entrées : mode auto, AU principal, AU pompe, disjoncteur fermé, défaut électrique, défaut ATS, défaut vanne, défaut discordance pompe, défaut température

b)-Les sorties : pompe dispo.

Le but de ce FB écrit en langage CONT est de vérifier si une pompe est disponible ou non après la vérification de l'ensemble de conditions.

### Le programme



### 4.1.7. Démarrage

Ce FB reçoit 16 signaux en entrée et envoie 5 signaux en sortie.



**a)-Les entrées :** cmd, retour de marche, fco, fcf, reset, à l'arrêt, en cours d'arrêt.

**b)-Les sorties :** cmd ouverture vanne, défaut discordance pompe, défaut discordance vanne, en marche, en cours de démarrage.

Ce bloc doit assurer :

-Lancement de la commande physique à la pompe si la vanne est fermée et on considère que la pompe en cours de démarrage

-Traitement de défaut de discordance de la pompe et délivrance d'un signal qui indique la présence d'un défaut dans le cas.

-En cas d'absence de défaut discordance pompe, une commande d'ouverture vanne sera lancée et surveillée.

-Délivrance d'un signal qui indique la présence d'un défaut discordance vanne si bloc ne reçoit pas le fin de course ouverture.

-A la fin de la séquence de démarrage, la pompe sera considérée en marche.

#### **4.1.8. Gestion de démarrage de la ligne HADJ YAKOUB**

Ce FB reçoit 11 signaux en entrée et envoie 2 signaux en sortie.

**a)-Les entrées :**

**dispo4, dispo5 :** disponibilité des pompes 4 et 5.

**s4, s5, m4, m5, h4, h5 :** temps d'arrêt des pompes 4 et 5.

**Retour de marche4, retour de marche5 :** les retours de marche des pompes 4 et 5.

**b)-Les sorties :** cmd p4, cmd p5.

Le but de ce FB écrit en langage CONT, est de gérer les commandes de démarrage des pompes de la ligne HADJ YAKOUB et il doit assurer :

-Une comparaison des temps d'arrêt des GEP 4 et 5 lors de l'activation de la demande de pompage.

-Le lancement de commande pour une seule pompe, si la pompe choisie par la comparaison est disponible sa commande sera activée, si elle n'est pas disponible et l'autre pompe est disponible donc la commande de cette dernière sera activée.

-Si une pompe en marche perd sa disponibilité et que l'autre pompe est disponible, un basculement doit être fait après une temporisation, le temps que la pompe indisponible s'arrête (priorité à l'arrêt).

#### 4.1.9. Gestion de démarrage de la ligne DEKAKNA

Ce FB reçoit 11 signaux en entrée et envoie 2 signaux en sortie.

**a)-Les entrées :**

**dispo1, dispo2, dispo3** : disponibilité des pompes 1, 2 et 3.

**s1, s2, s3, m1, m2, m3, h1, h2, h3** : temps d'arrêt des pompes 1,2 et 3.

**Retour de marche1, retour de marche2, retour de marche3** : les retours de marche des pompes 1,2 et 3.

**b)-Les sorties** : cmd p1, cmd p2, cmd p3.

Le but de ce FB est de gérer les commandes de démarrage des pompes de la ligne DEKAKNA et il doit assurer :

- Une comparaison des temps d'arrêt des GEP 1,2 et 3 lors de l'activation de la demande de pompage.

-Selon le cahier de charge deux pompes sur trois doivent être mise en marche.

- la commande de démarrage est activée pour la pompe disponible et possédant le temps d'arrêt le plus grand, après un certain temps (car les pompes ne doivent jamais démarrer en même temps) et si la deuxième pompe sélectionnée est disponible on lance sa commande.

- En cas d'indisponibilité des pompes sélectionnées un basculement se fait vers une pompe disponible.

- Si une pompe en marche perd sa disponibilité et qu'une autre pompe est disponible, un basculement doit être fait après une temporisation le temps que la pompe indisponible s'arrête (priorité à l'arrêt).

#### 4.1.10. Gestion arrêt ligne DEKAKNA

Ce FB reçoit 5 signaux en entrée et envoie 3 signaux en sortie.

**a)-Les entrées** : retour de marche des pompes 1, 2, 3, NB bêche et NH réservoir.

**b)-Les sorties : cmd arrêt p1, cmd arrêt p2, cmd arrêt p3.**

Le but de ce FB écrit en langage SCL est de gérer les commandes d'arrêt des pompes de la ligne DEKAKNA car dans le cas normal deux pompes seront en marche, l'arrêt de ces dernières se base sur le temps de marche le plus grand .Ce bloc doit assurer :

- Comparaison des temps de marche des 2 GEP qui sont en marche sur le niveau bas de la bêche d'aspiration ou sur le niveau haut du réservoir.
- Le choix de la pompe possédant le temps de marche le plus grand et lancement de la commande d'arrêt, après son arrêt, le lancement de la commande pour arrêter l'autre pompe qui est en marche.

**4.1.11. Démarrage arrêt de la ligne DEKAKNA**

Ce FB contient les entrées/sorties physiques de chacune des 3 pompes ainsi que les entrées générales.

<b>Les entrées pour chaque pompe</b>	<b>Les entrées générales</b>	<b>Les sorties pour chaque pompe</b>
Arrêt d'urgence pompe Mode automatique Disjoncteur pompe Défaut ATS Fin de course ouverture vanne Fin de course fermeture vanne Retour de marche. Reset des défauts Température bobinage 1. Température bobinage 2. Température palier 1. Température palier 2.	Arrêt d'urgence principal Disjoncteur principal Défaut phase Disjoncteur inverseur Présence 24V Présence 230V NH ; niveau haut réservoir NB ; niveau bas réservoir NH niveau haut bêche NB niveau bas bêche	Commande fermeture vanne Commande démarrage /arrêt Commande ouverture vanne

*Tableau V-1 Les entrées-sorties physiques pour chaque pompe.*

En cas de demande de pompage on fait appel au FB gestion démarrage de la ligne de DEKAKNA pour choisir les 2 pompes à démarrer, en traitant tous les défauts (discordance pompe, discordance vanne, défauts températures, défauts électrique), après le choix on fait appel au bloc démarrage.

En cas de commande d'arrêt de pompage, on fait appel au bloc «Gestion arrêt ligne DEKAKNA» pour choisir l'ordre d'arrêt des pompes en marche. Une fois le choix est fait la pompe est considérée en cours d'arrêt, on ferme d'abord la vanne de refoulement, après on arrête la commande, apres la disparition du retour de marche la pompe est considérée à l'arrêt.

En cas de défaut au niveau d'un des GEP en marche, il sera arrêté immédiatement et un basculement se fait vers un autre GEP qui est disponible (suivant la gestion de démarrage).

S'il y a une présence d'un défaut au niveau de la station, tous les GEPs en marche se mettent à l'arrêt.

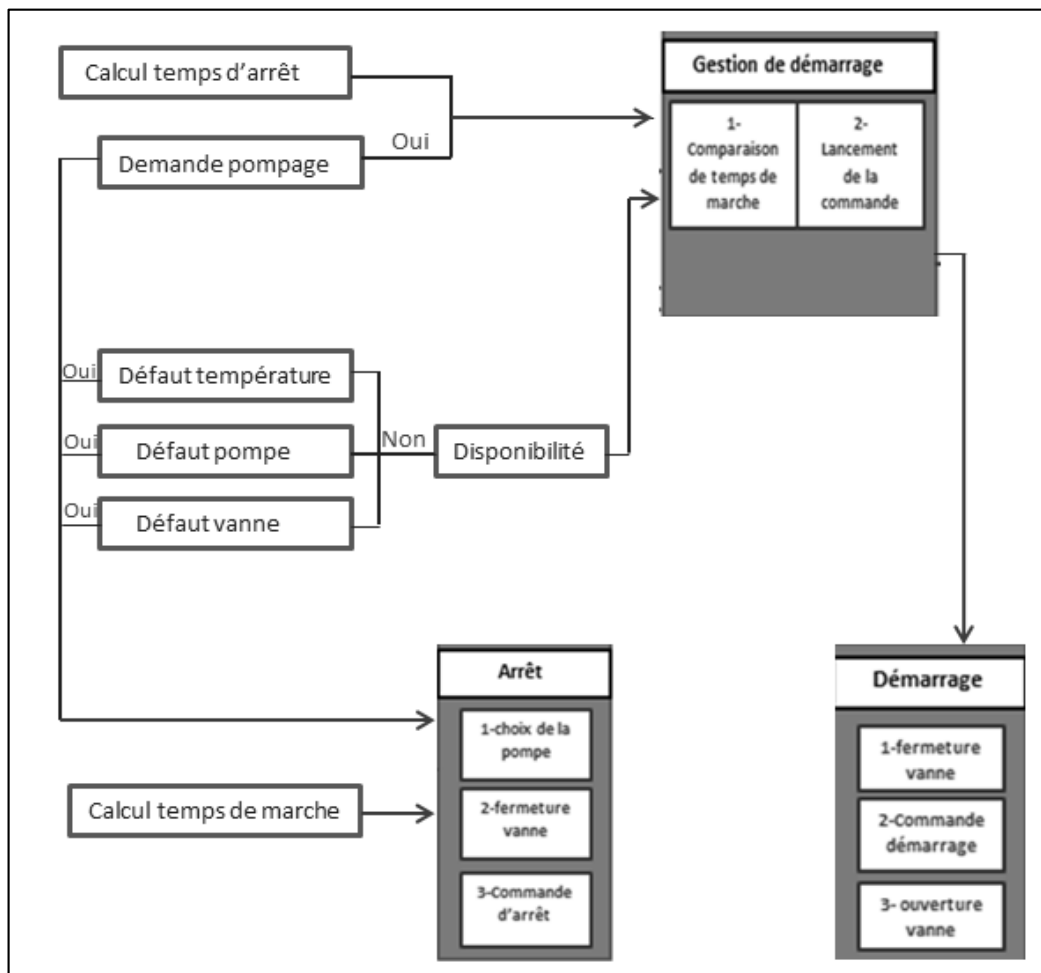


Figure V-2 Schéma descriptif du démarrage-arrêt de la ligne DEKAKNA.

#### 4.1.12. Démarrage arrêt de la ligne HADJ YAKOUB

Ce FB contient les entrées/sorties physiques de chacune des 2 pompes ainsi que les entrées générales physiques citées auparavant.

En cas de demande de pompage on fait appel au bloc «Gestion démarrage de la ligne de HADJ YAKOUB» pour choisir la pompes à démarrer en traitant tous les défauts (discordance pompe, discordance vanne, défauts températures, défauts électrique), une fois le choix est fait on appelle le bloc «Démarrage».

En cas de commande d'arrêt pompage, la pompe sera considérée en cours d'arrêt, une commande de fermeture de la vanne de refoulement est lancée et la commande de la pompe est désactivée. On considère que la pompe est à l'arrêt après la disparition du retour de marche.

En cas de défaut au niveau de GEP en marche il sera arrêté immédiatement et un basculement se fait vers le GEP disponible restant. En cas de présence d'un défaut au niveau de la station, le GEP en marche se met à l'arrêt.

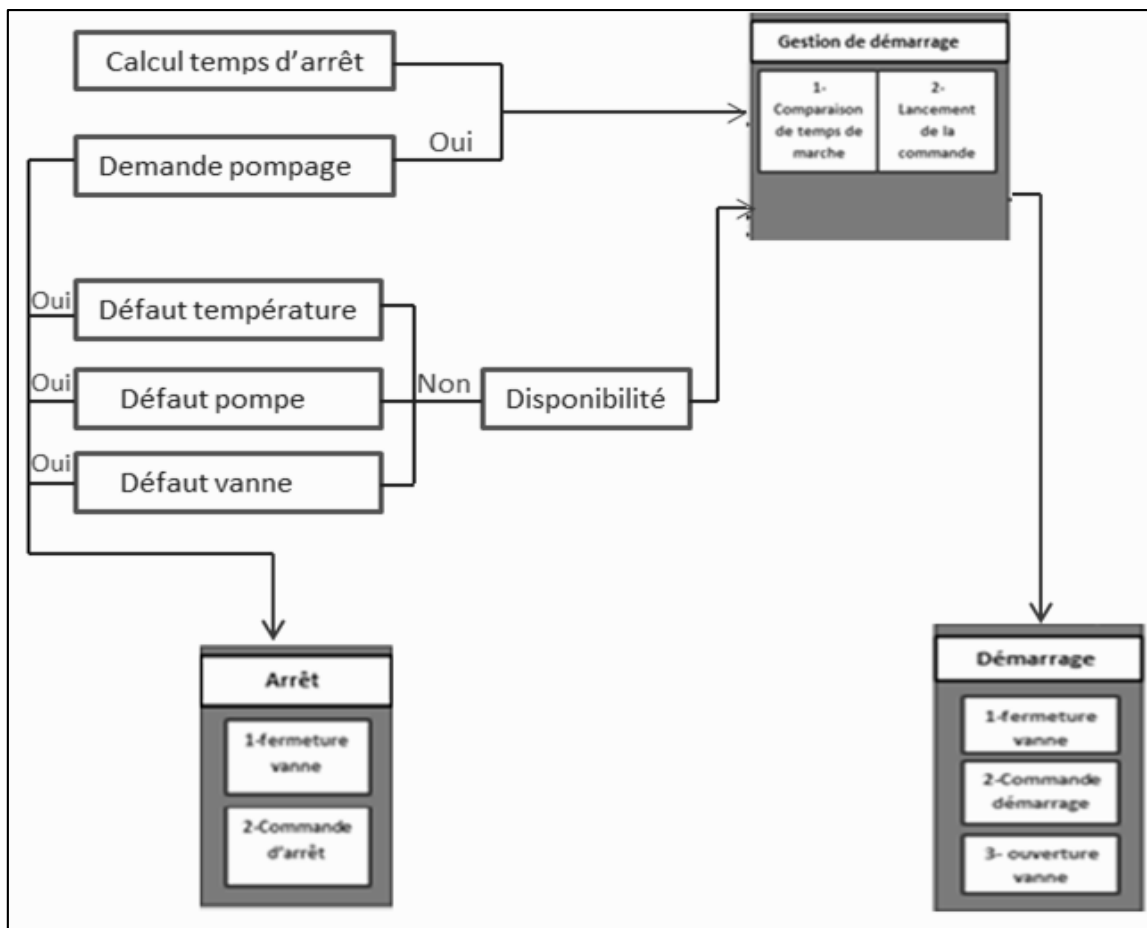


Figure V-3 Schéma descriptif du démarrage-arrêt de la ligne HADJ YAKOUB.

#### 4.1.13. MB Server

Ce bloc comprend plusieurs paramètres :

**Les entrées :**

**Disconnect:** L'instruction "MB\_SERVER" établit une connexion passive avec un module partenaire, c'est-à-dire que le serveur réagit à chaque demande de connexion TCP provenant de chacune des adresses IP. Ce paramètre permet de commander le moment où une demande de connexion doit être acceptée.

**Connect\_ID :** Définit un ID à la liaison (au choix).

**IP\_Port:** Définit le port de connexion.

**MB\_HOLD\_REG :** Définit le registre à lire ou sur lequel on va écrire (peut être un memento ou un DB).

**Les sorties :**

**NDR :** A "1" si nouvelles données écrites par le client Modbus.

**DR :** A "1" si les données sont lues par le client Modbus.

**Error :** A "1" s'il y'a erreur pendant la communication.

**Status :** code erreur.

L'instruction "MB\_SERVER" communique en tant que serveur Modbus TCP via la connexion PROFINET de la CPU S7-1200. L'instruction "MB\_SERVER" traite les demandes de connexion d'un client Modbus TCP, reçoit les requêtes des fonctions Modbus et envoie les messages de réponse.

## 4.2. Description des FC

### Adressage

Le but de cette FC est d'affecter au DB données à transmettre les variables du process qu'on veut envoyer à la supervision à distance ainsi que les commandes reçues par la station travers la télégestion du TOPKAPI.

**Données à transmettre :**

**Les pompes :** en marche, en arrêt, en défaut, disponible.

**Les vannes :** ouvertes, fermée, en défaut.

**Les niveaux :** niveau haut et bas de la bache d'aspiration.

**Energie :** présence 24V, présence 230V, état disjoncteur inverseur et principal, défaut phase.

Données à recevoir :

Les pompes : commandes marche-arrêt.

### 4.3. Description des DB

#### 4.3.1. DB données station

Ce DB contient des informations générales de la station : la demande pompage des deux lignes ainsi que les défauts électriques de la station qui doivent être accessible sans faire appel aux FB internes du programme.

donnée station								
	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Visible da...	Valeur de ..	Commentaire
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	demande pompage ligne 1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	demande pompage ligne 2	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	defaut électrique	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

#### 4.3.2. DB données pompe

Ce DB est créé pour chaque pompe, contient les défauts de la pompe (discordance, température), l'état de la pompe (en marche, en arrêt, ou encours de démarrage ou arrêt), les défauts de la vanne (discordance ouverture ou fermeture), les temps de marches et les temps d'arrêt de la pompe ainsi que des informations sur vanne.

données pompe1								
	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Visible da...	Valeur de ..	Commentaire
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	disponibilité 1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	defaut disco1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	defaut ouverture vanne1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	defaut fermeture vanne1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	defaut vanne1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	defaut tmp b1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	defaut tmp p1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	p1 en cours de démarrage	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	p1 en cours d'arrêt	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	pompe1 en marche	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	pompe1 en arrêt	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	temp-marche h1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	temp-marche m1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	temp-marche s1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	temp-arret h1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	temp-arret m1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	temp-arret s1	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

#### 4.3.3. DB donnée télégestion

Ce DB se compose d'un tableau de 4 mots, chaque bit de ces mots contient une information qui va être transmise vers la supervision TOPKAPI par exemple le défaut d'une pompe ou son état.

données telegestion									
	Nom	Type de données	Décalage	Valeur de départ	Rémanence	Accessible ...	Visible da...	Valeur de ..	Com...
1	▼ Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	▼ données	Array[0..3] o...	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	données[0]	Word	0.0	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	données[1]	Word	2.0	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	données[2]	Word	4.0	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	données[3]	Word	6.0	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

## 5. Développement de la supervision

### 5.1. Création de la station HMI

Dans notre projet on introduit une station HMI en choisissant le pupitre TP900 COMFORT sur lequel on crée des vues pour la visualisation globale de la station ce pupitre se caractérise par le tableau de la page suivante :

Série fabricant	TP900.
Type d'affichage	TFT.
Taille de l'affichage	9 pouces.
Résolution de l'affichage	800 x 480pixels.
Couleur de l'afficheur	Coloré.
Type de port	Ethernet, MPI, Profibus DP, USB.
Mémoire embarquée	12 Mo.
Tension d'alimentation	24 VCC.
Dimensions	274 x 190 x 63mm.
Indice IP	IP20.

Tableau V-2 Caractéristiques de l'HMI.

### 5.2. Configuration des alarmes

Pour créer les alarmes, on doit créer d'abord un mot dans les variables API 'mot de de défaut' qui va contenir les alarmes bit de notre système, puis on accède à alarme HMI dans le navigateur projet. Un tableau s'affiche on le remplit comme la figure suivante le montre :



ID	Texte d'alarme	Classe d'alar...	Variable de d...	Bit de ..	Adresse de dé..	Variable d'acq..	Bit d'a...
7	defaut discordance pompe 1	Errors	mot de de...	8	*mot de defa...	<aucune v...	0
8	defaut vanne 1	Errors	mot de defaut	9	*mot de defa...	<aucune vari...	0
9	defaut temperature pompe 1	Errors	mot de defaut	10	*mot de defa...	<aucune vari...	0
10	defaut discordance pompe 2	Errors	mot de defaut	11	*mot de defa...	<aucune vari...	0
11	defaut vanne 2	Errors	mot de defaut	12	*mot de defa...	<aucune vari...	0
12	defaut temperature pompe 2	Errors	mot de defaut	13	*mot de defa...	<aucune vari...	0
13	defaut discordance pompe 3	Errors	mot de defaut	14	*mot de defa...	<aucune vari...	0
14	defaut vanne 3	Errors	mot de defaut	15	*mot de defa...	<aucune vari...	0
15	defaut temperature pompe 3	Errors	mot de defaut	0	*mot de defa...	<aucune vari...	0
16	defaut discordance pompe 4	Errors	mot de defaut	1	*mot de defa...	<aucune vari...	0
17	defaut vanne 4	Errors	mot de defaut	2	*mot de defa...	<aucune vari...	0
18	defaut temperature pompe 4	Errors	mot de defaut	3	*mot de defa...	<aucune vari...	0
19	defaut discordance pompe 5	Errors	mot de defaut	4	*mot de defa...	<aucune vari...	0
20	defaut vanne 5	Errors	mot de defaut	5	*mot de defa...	<aucune vari...	0
21	defaut temperature pompe 5	Errors	mot de defaut	6	*mot de defa...	<aucune vari...	0

Figure V-4 Tableau de création des alarmes.

### 5.3. Création des vues

Les éléments représentatifs de la station sont utilisés à partir de la bibliothèque du WinCC, à chaque composant on affecte une variable, une adresse, une représentation et un évènement qui lui est assigné.

- **Vue synoptique** : c'est une vue générale de la station qui indique l'état des 5 GEP (en marche, en arrêt, disponibilité) ainsi que l'état de la vanne (fermée ou ouverte).

Cette vue affiche aussi le niveau d'eau dans la bache d'aspiration et dans les deux réservoirs DEKAKNA et HADJ YAKOUB, au-dessous de la vue deux boutons ont été inséré pour accéder à la vue des défauts électriques et la vue des alarmes.

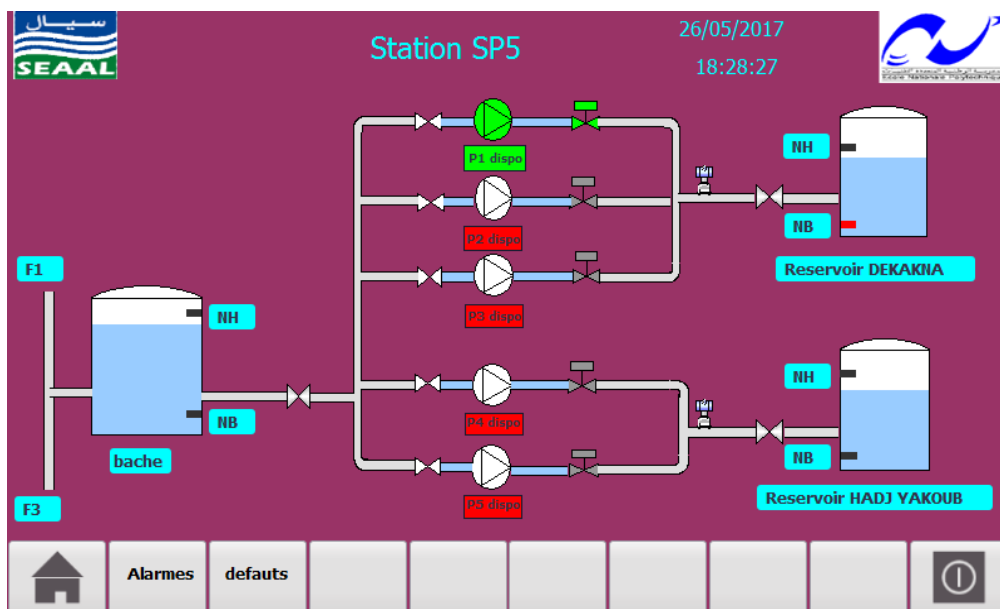


Figure V-5 Synoptique de la station SP5.







symbole	Représentation	Affiliation
	Pompe	<b>Blanc</b> : pompe en arrêt <b>Vert</b> : pompe en marche
	Vanne manuelle	<b>Pas d'affiliation</b>
	Vanne motorisée	<b>Vert</b> : ouverte <b>Gris</b> : fermée
	Flotteur	<b>Rouge</b> : niveau détecté (1) <b>Noir</b> : pas de détection de niveau (0)
	Débitmètre	<b>Pas d'affiliation</b>
	Disponibilité	<b>Rouge</b> : pompe indisponible <b>Vert</b> : pompe disponible

Tableau V-3 Affiliation de la vue synoptique.

- **Vues pompes** : En cliquant sur l'icône d'une pompe dans la vue synoptique une autre vue s'affiche contenant les conditions électriques, l'état de la pompe (en marche en arrêt, en défaut), la vanne (fermée, ouverte, en défaut) ainsi que le temps d'arrêt et de marche de la pompe.



Figure V-6 Vue des pompes.

Case	Affiliation
Disjoncteur	<b>Vert</b> : disjoncteur fermé <b>Rouge</b> : disjoncteur ouvert
Mode auto	<b>Vert</b> : le mode auto est activé <b>Rouge</b> : le mode auto est désactivé
Défaut ATS	<b>Vert</b> : pas de défaut démarreur <b>Rouge</b> : présence d'un défaut ATS
AU principal, AU pompe	<b>Vert</b> : pas d'arrêt d'urgence <b>Rouge</b> : arrêt d'urgence
Vanne ouverte, fermée,	<b>Vert</b> : présence fin de course <b>Gris</b> : pas de fin de course
En défaut (vanne et pompe)	<b>Rouge</b> : présence défaut vanne/pompe <b>Vert</b> : pas de défaut
En marche	<b>Vert</b> : pompe en marche <b>Gris</b> : la pompe n'est pas en marche
A l'arrêt	<b>Vert</b> : pompe à l'arrêt <b>Gris</b> : la pompe n'est pas en arrêt
Temps de marche (arrêt)	Indique le temps de marche (arrêt) en heure, minute, seconde

Tableau V-4 Affiliation de la vue de pompe.

- **Vue défaut électrique** : cette vue englobe l'état des disjoncteurs (fermé ou ouvert) ainsi que la présence des tensions 24 et 230 V et du défaut phase.



Figure V-7 Vue défauts électriques générales.

Case	Affiliation
Disjoncteur principal, inverseur	<b>Vert</b> : fermé <b>Rouge</b> : ouvert
Présences 24V, présence 230V	<b>Vert</b> : présence de tension <b>Rouge</b> : absence de tension
Défaut phase	<b>Vert</b> : pas de défaut <b>Rouge</b> : présence d'un défaut phase

Tableau V-5 Affiliation de la vue défaut électrique.

- **Vue alarmes** : cette vue comporte un tableau qui affiche les alarmes actuelles du process, leurs état, la date et l'heure d'apparence. On peut acquitter l'alarme en cliquant sur le bouton acquittement qui se trouve au-dessous du tableau à droite, cela fera disparaître l'alarme de ce tableau mais pas réellement.

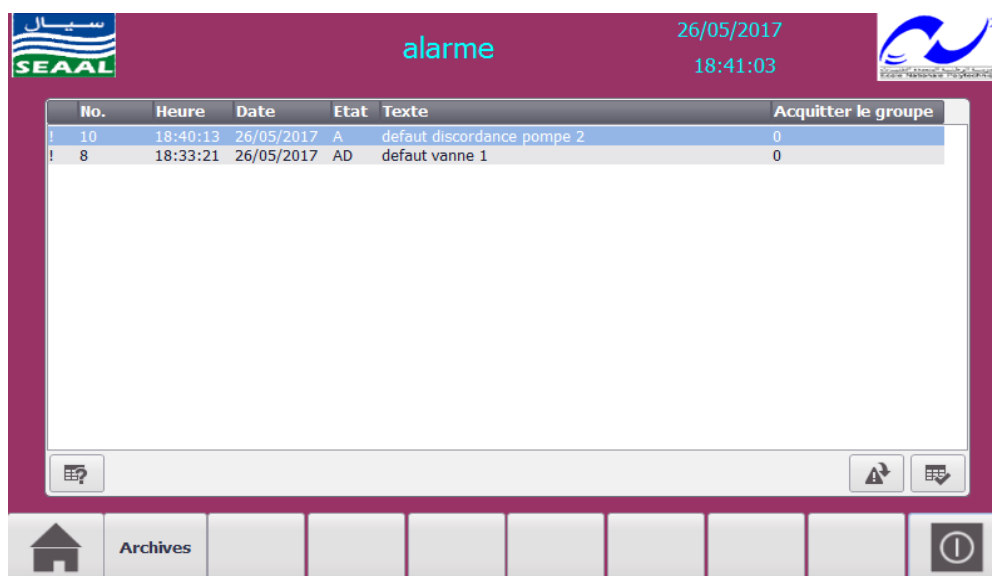


Figure V-8 Vue des alarmes.

- **Vue alarme tampon** : en cliquant sur le bouton Archives de la vue d'alarme précédente un autre tableau s'affiche contenant l'archive des alarmes. Comme l'exemple ci-dessous le tableau montre les alarmes leurs date et heure d'apparition ainsi que leurs états (acquittée, apparente...)

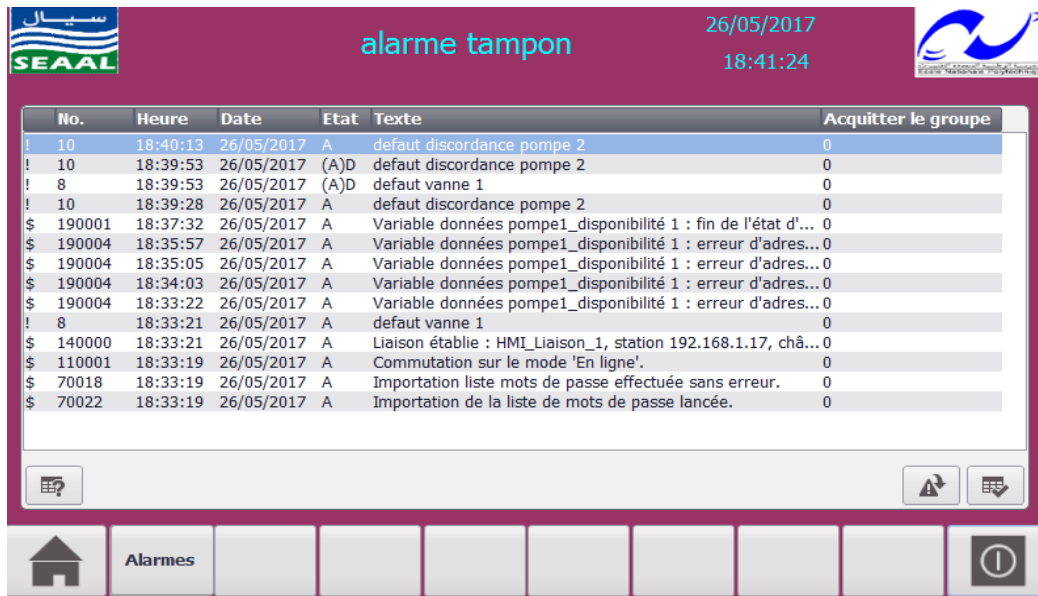


Figure V-9 Vue des alarmes tampons.

- **Vue caractéristiques pompe et moteur :** cette vue affiche des informations sur la pompe (modèle, débit et d’autres informations) ainsi que les caractéristiques du moteur de chaque ligne.



Figure V-10 Vue caractéristiques pompe et moteur.

## 6. Développement de l’application télégestion avec TOPKAPI

Dans le but d’établir une communication de la station de pompage SP5 avec la salle de contrôle qui se situe à la direction générale de SEAAL a Kouba à travers le TOPKAPI et après avoir configuré les routeurs 3G, on a édité une fiche tableur de la station qui contient les variables qui vont être transmis ou reçus par le TOPKAPI.

Lors de l'ouverture du logiciel TOPKAPI une fenêtre principale s'affiche, on commencera par la suite les configurations nécessaires pour garantir l'échange des données entre l'automate et la supervision a traves ce logiciel.

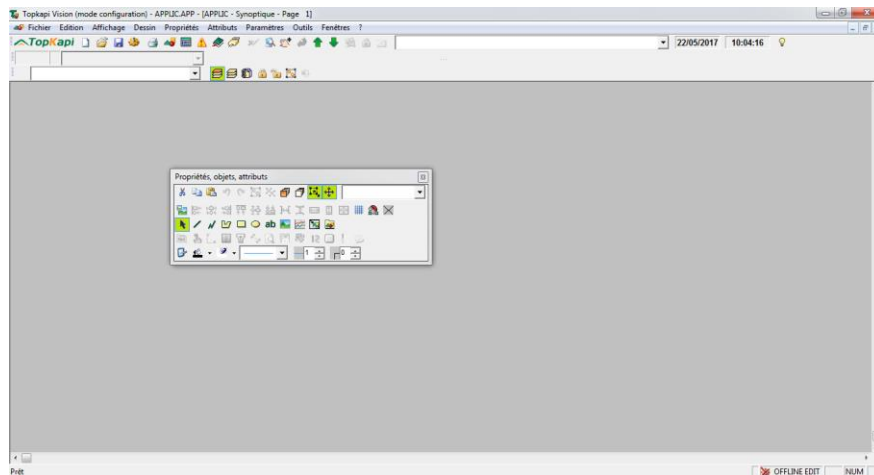


Figure V-11 Fenêtre principale du logiciel TOPKAPI.

## 6.1. Configuration de protocole de communication

On ajoute un port de communication ,parmi les different types de communications disponible sur le TOPKAPI ,on choisit le protocole MODBUSIP qui est compatible avec l'autoamte S7-1200 et on clique sur OK.

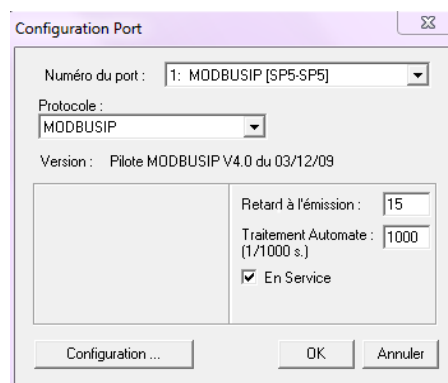


Figure V-12 Configuration du port.

## 6.2. Configuration de l'automate

On ajoute un automate puis on introduit le nom de la station, l'adresse IP de la carte SIM utilisée dans le routeur de la station et les zones mémoires et on valide.

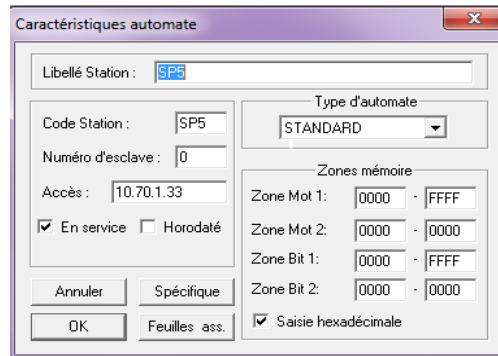


Figure V-13 Configuration de la station automate.

### 6.3. Paramétrages des données et remplissage de la fiche tableur

#### 1) Les listes système

On crée d’abord les différentes listes système qui vont être utilisées dans le paramétrage.

- zone de localisation : SP5.
- profils d’archivage : pompe, capteurs, énergie.
- profils d’échantillonnage : TOR.

#### 2) Les données TOR

Pour chaque donnée on introduit : le type (TOR, bit de mot), un titre, une adresse, libellé, une zone de localisation, un profil d’archivage et un profil d’échantillonnage. Lorsque tous les propriétés sont choisis on clique sur « Appliquer ». Les données TOR servent à recevoir les informations TOR de la station de pompage ex : pompe en marche, vanne fermée

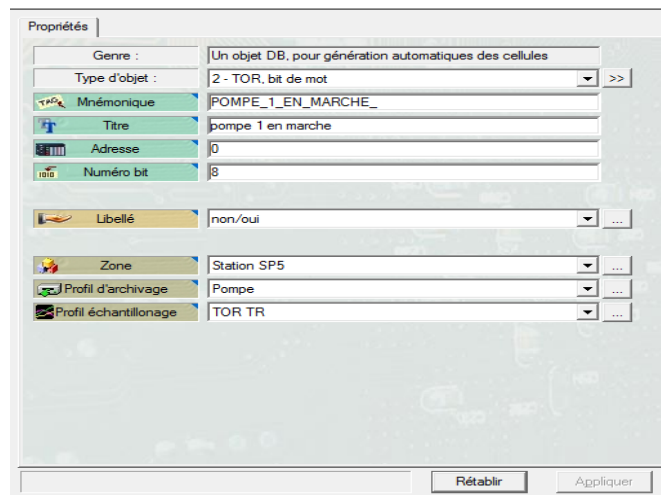


Figure V-14 Configuration des données TOR.

### 3) Les données alarmes

Pour chaque donnée alarme, on introduit : le type (TOR, bit de mot d'alarme), un titre, une adresse, libellé, rang défaut, une zone de localisation, un profil d'archivage et un profil d'échantillonnage. Lorsque tous les propriétés sont choisi on clique sur « Appliquer ». Les données alarme représente dans notre cas les défauts sur les pompes que la station communiquera a la supervision.

Genre :	Un objet DB, pour génération automatiques des cellules
Type d'objet :	4 - TOR, bit de mot alarme
Mnémonique :	POMPE1_EN_DEFAULT_
Titre :	POMPE 1 EN_DEFAULT
Adresse :	0
Numéro bit :	10
Libellé :	non/oui
Rang défaut :	Avertissement
Groupe Astreinte :	Aucun
Opérateurs autorisés :	
Zone :	Station SP5
Profil d'archivage :	Pompe
Profil d'échantillonnage :	TOR TP

Figure V-15 Configuration des données alarmes.

### 4) Les données télécommande

Pour chaque variable télécommande, on introduit : le type (télécommande manuelle bit de mot), un titre, une adresse, libellé, opérateur autorisé, une zone de localisation, un profil d'archivage. Lorsque tous les propriétés sont choisi on clique sur « Appliquer ». Les variables télécommandes sont utilisées pour envoyer des commande de la supervision vers l'automate dans notre cas on a choisi d'envoyer les commandes d'arrêt et de démarrage des pompes à la station SP5.



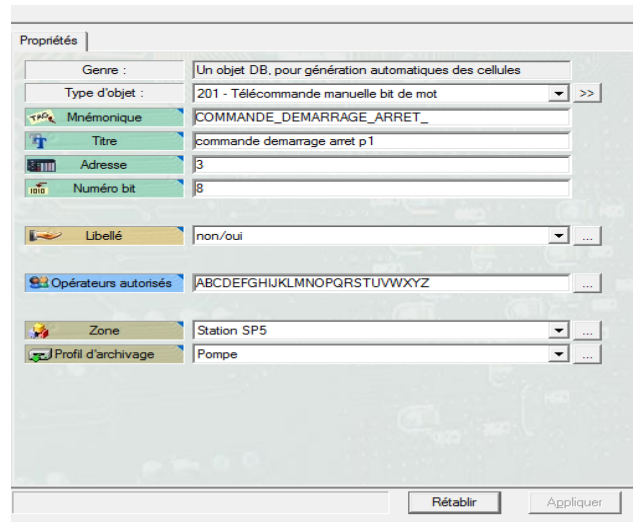


Figure V-16 Configuration des données télécommande.

Après avoir configuré et paramétré toutes les données à recevoir et à transmettre on aura la fiche tableau suivante :

	A	B	C	D	E	F
1						
2	1 pompe 1					
3	1 pompe 1 en marche	NON				
4	1 pompe 1 en arrêt	NON				
5	1 COMPE 1_EN_DEFAULT	NON				
6	1 disponibilité pompe 1	NON				
7						
8	2 pompe 2					
9	2 pompe 2 en marche	NON				
10	2 pompe 2 en arrêt	NON				
11	2 COMPE _2_EN_DEFAULT	NON				
12	2 disponibilité pompe 2	NON				
13						
14	3 pompe 3					
15	3 pompe en marche 3	NON				
16	3 pompe 3 en arrêt	NON				
17	3 COMPE _3_EN_DEFAULT	NON				
18	3 disponibilité pompe 3	NON				
19						
20	4 pompe 4					
21	4 pompe 4 en marche	NON				
22	4 pompe 4 en arrêt	NON				
23	4 COMPE _4_EN_DEFAULT	NON				
24	4 disponibilité pompe 4	NON				
25						
26	5 pompe 5					
27	5 pompe 5 en marche	NON				
28						
29						

Figure V-17 La fiche tableau SP5 après remplissage.

## 7. Réalisation matérielle

### 7.1. Présentation du montage

Afin de charger le programme édité dans l'automate S7-1200 et dans l'HMI TP900 COMFORT, on a réalisé une maquette simulant l'armoire d'automatisme de la station qui se compose de :

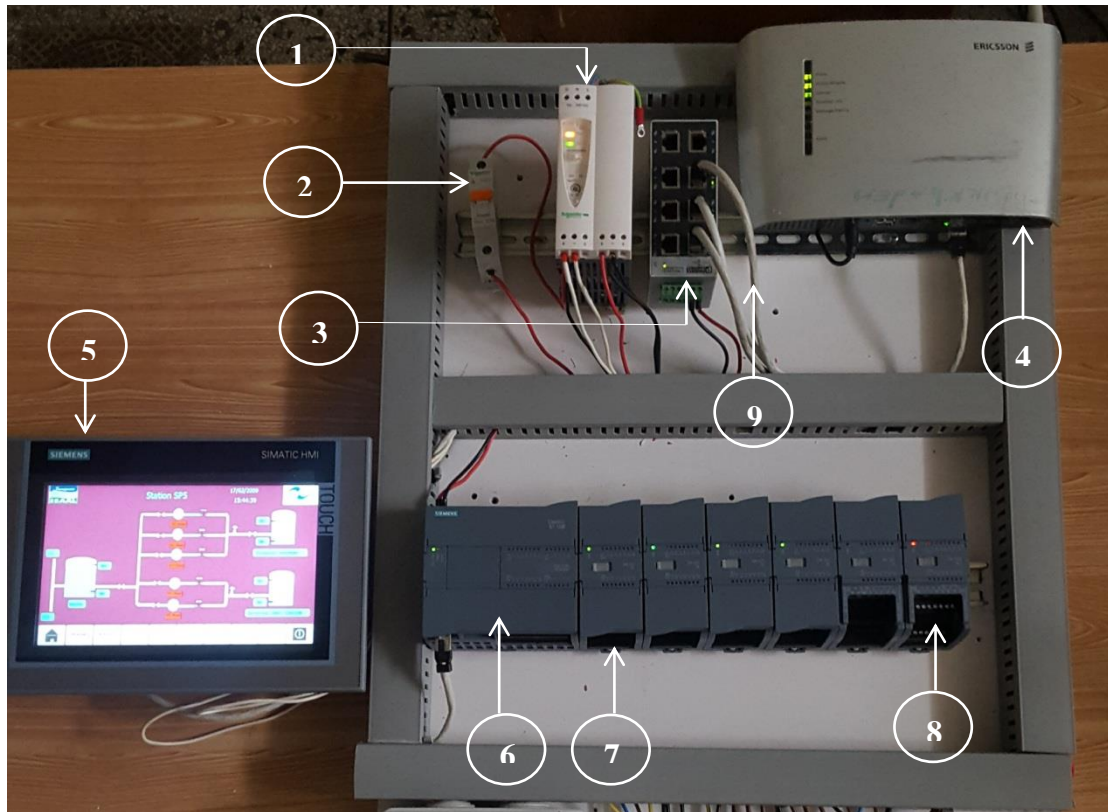


Figure V-18 la maquette d'armoire d'automatisme.

- |                           |                                       |
|---------------------------|---------------------------------------|
| 1-Alimentation Schneider  | 6-Un automate S7-1200                 |
| 2-Disjoncteur.            | 7-Module d'entrées 16 entrées TOR.    |
| 3-Un switch.              | 8- Module de sortie 16 sorties RELAY. |
| 4-Un routeur 3G ERICSSON. | 9- Câble réseaux.                     |
| 5- HMI TP900 COMFORT.     |                                       |

## 7.2. Chargement du programme dans le matériel

Pour transférer le programme vers la CPU et l'HMI on suit les étapes suivantes :

- Compilation logiciel et matériel du projet.
- Connection de la CPU et l'HMI avec le PC via PROFINET :

Les deux appareils alimentés de 24 VDC et le pc sont connectés au switch à travers des câbles réseaux

- Recherche des appareils connectés avec le PC :

En cliquant sur l'icône Abonnés accessible dans le TIA Portal qui sert à identifier les périphériques connectés au pc et afficher leurs adresses IP. Tout d'abord il faut choisir le PROFINET comme type interface PG/PC et la carte réseau du pc comme interface PG/PC

ensuite on lance la recherche. Après la recherche, la carte réseau du pc détecte les deux appareils (API-HMI).

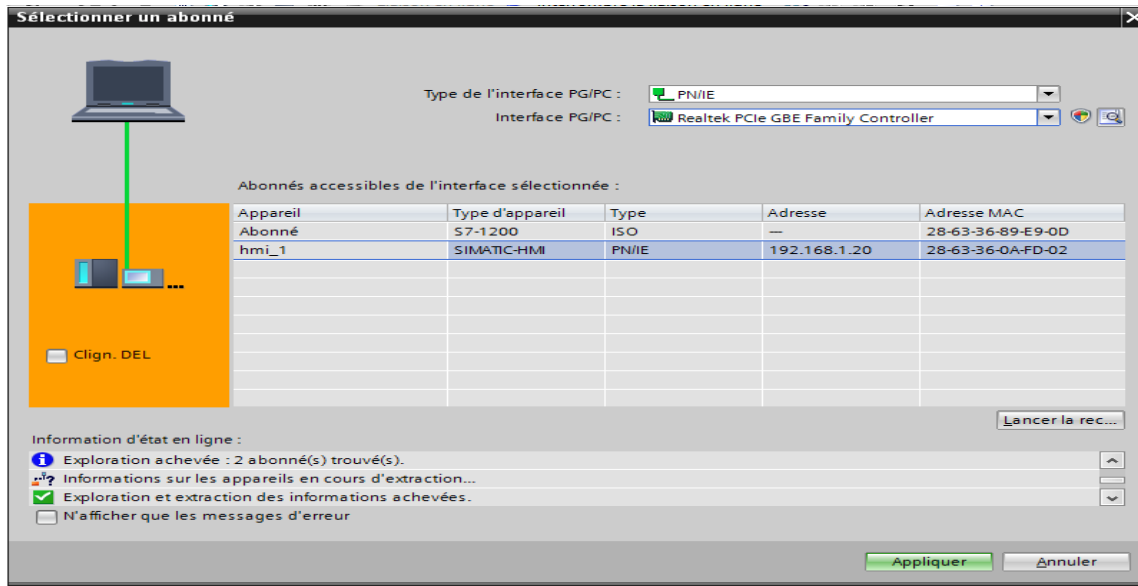


Figure V-19 Les abonnés accessibles.

- Configuration des adresses IP :

A la première utilisation l'automate ne possède pas d'adresse IP, il est donc nécessaire de lui affecter une adresse statique à partir de la configuration matérielle à charger.

Pour cela on accède à la configuration matérielle puis «propriétés » on sélectionne « adresses Ethernet », on lui affecte une adresse et un masque sous réseau.

Dans notre cas l'automate est connecté à un routeur, donc on doit introduire l'adresse IP du routeur utilisé.

Pour la configuration de l'adresse IP de l'HMI on suit les mêmes étapes.

Les adresses IP doivent être dans le même sous réseau que celle du PC.

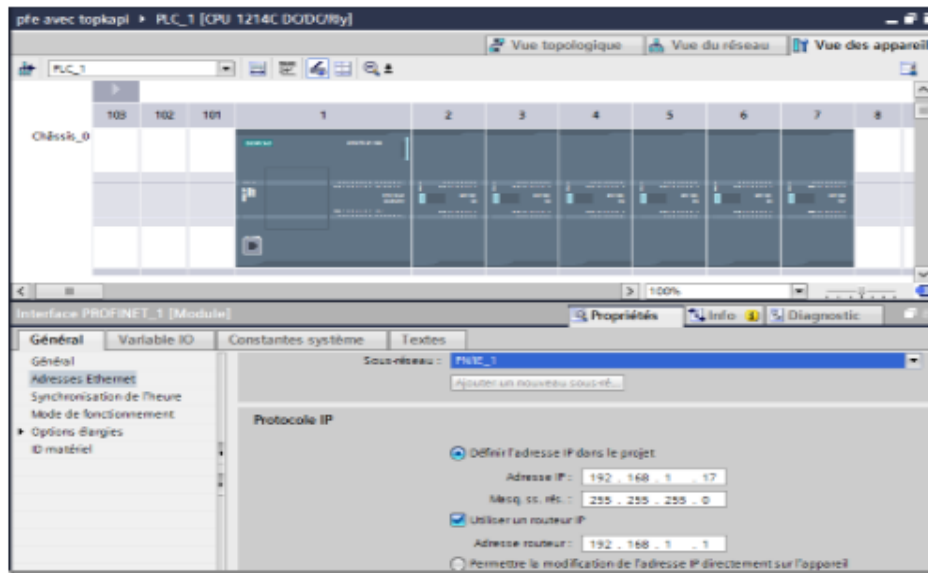


Figure V-20 Configuration des adresses IP.

- Chargement dans la CPU et l’HMI :

On sélectionne la CPU /HMI dans la vue projet et on clique sur charger, la fenêtre de la figure suivante s’affiche.

Après le choix de l’interface PG/PC et la carte réseau, on charge le programme cliquant sur «charger ».

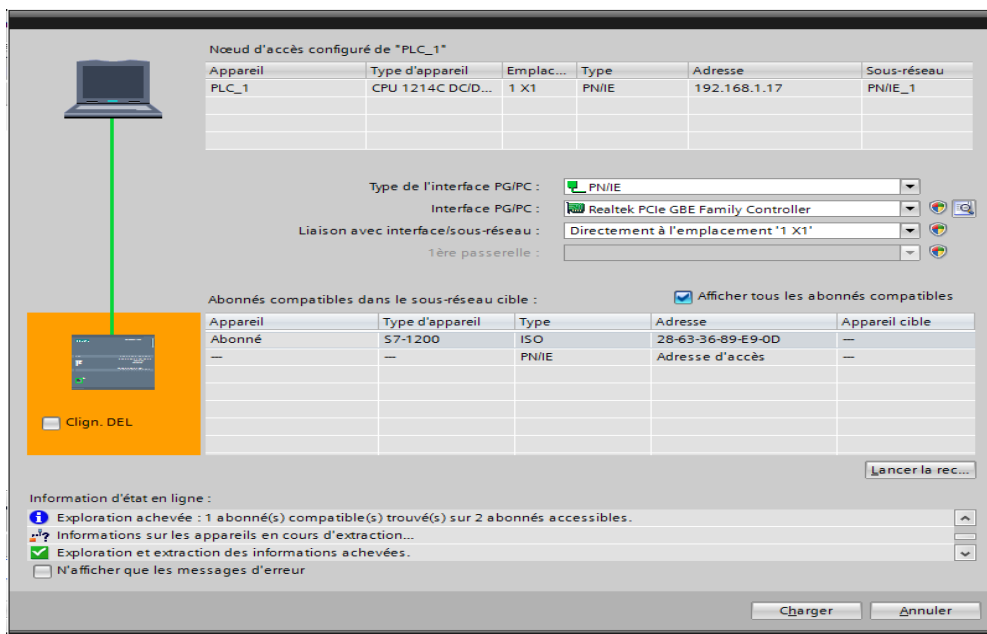


Figure V-21 Chargement du programme dans l’API.

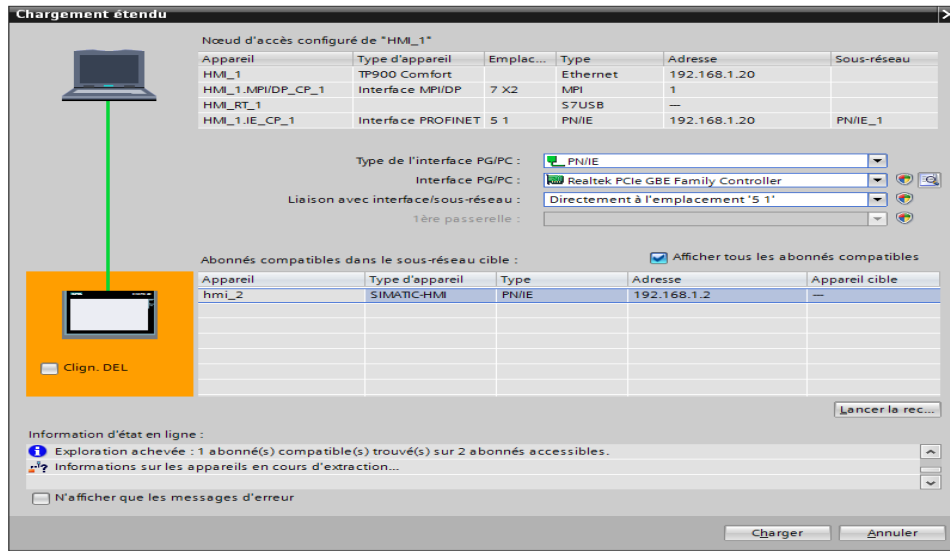


Figure V-22 Chargement du programme dans l'HMI.

### 7.3. Le test de la télégestion

Pour la télégestion on a connecté l'automate de la station SP5 et le pc de la supervision de la direction à des routeurs ERICSSON 3G.

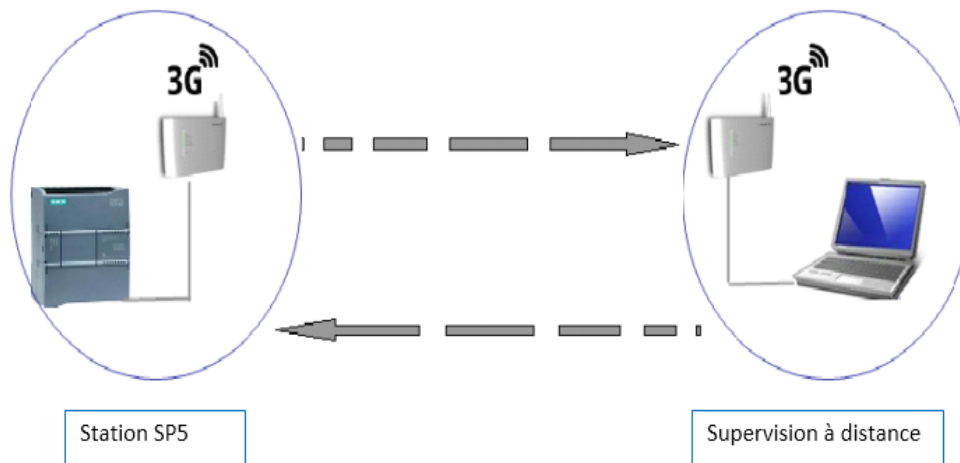


Figure V-23 Communication 3G entre la station et la supervision.

L'état des variables du programme de la station est visualisé dans la fiche tableau du TOPKAPI, dans la figure suivante on voit que la pompe 1 est disponible et en marche tandis que les autres pompes sont en arrêt et en défaut.

	A	B	C	D	E	F	G	H	I	J	K	L
1	pompe 1											
2	pompe 1 en marche	oui										
3	pompe 1 en arrêt	non										
4	POMPE 1_EN_DEFAULT	non										
5	disponibilité pompe 1	oui										
6												
7	pompe 2											
8	pompe 2 en marche	non										
9	pompe 2 en arrêt	oui										
10	POMPE__2_EN_DEFAULT_	oui										
11	disponibilité pompe 2	non										
12												
13	pompe 3											
14	pompe 3 en marche 3	non										
15	pompe 3 en arrêt	oui										
16	POMPE_3_EN_DEFAULT	oui										
17	disponibilité pompe 3	non										
18												
19	pompe 4											
20	pompe 4 en marche	non										
21	pompe 4 en arrêt	oui										
22	POMPE_4_EN_DEFAULT_	oui										
23	disponibilité pompe 4	non										
24												
25	pompe 5											
26	pompe 5 en marche	non										
27	pompe 5 en arrêt	oui										

Figure V-24 La fiche tableau en mode exploitation.

Une fenêtre affichant les alarme indiquant les défauts des pompes de la station SP5 ainsi que l’archive des évènements contenant les variables échangées.

Date	Heure	F	Cel	Mnemonique	Zone	Libellé	Valeur	Défaut	Valeur numérique	Date acc
23/05/2017	14:35:25	SP5B10		POMPE_2_EN_DEFAULT_	Station SP5	POMPE_2_EN_DEFAULT	oui	3éfaut 1	1	
23/05/2017	14:48:43	SP5B16		POMPE_3_EN_DEFAULT_	Station SP5	POMPE_3_EN_DEFAULT	oui	3éfaut 1	1	
23/05/2017	14:48:43	SP5B22		POMPE_4_EN_DEFAULT_	Station SP5	POMPE_4_EN_DEFAULT	oui	3éfaut 1	1	
23/05/2017	14:48:43	SP5B28		POMPE_5_EN_DEFAULT_	Station SP5	POMPE_5_EN_DEFAULT	oui	3éfaut 1	1	
23/05/2017	15:37:17	SP5B03		POMPE_1_EN_ARRET	Station SP5	Pompe 1 en arrêt	oui		1	
23/05/2017	15:37:17	SP5B04		POMPE1_EN_DEFAULT_	Station SP5	POMPE 1 EN_DEFAULT	oui	3éfaut 1	1	
23/05/2017	15:37:17	SP5B05		DISPONBLITE_P1	Station SP5	disponibilité pompe 1	non		0	
23/05/2017	15:37:36	SP5B03		POMPE 1 EN_ARRET	Station SP5	Pompe 1 en arrêt	non		0	
23/05/2017	15:37:36	SP5B04		POMPE1_EN_DEFAULT_	Station SP5	POMPE 1 EN_DEFAULT	non	NORMAL	0	
23/05/2017	15:37:36	SP5B05		DISPONBLITE_P1	Station SP5	disponibilité pompe 1	oui		1	
23/05/2017	15:37:59	SP5B02		POMPE_1_EN_MARCHE_	Station SP5	pompe 1 en marche	oui		1	
23/05/2017	15:37:59	SP5B32		VANNE_1_OUVERTE_	Station SP5	vanne 1 ouverte	oui		1	
23/05/2017	15:38:51	SP5B72		COMMANDE_DEMARRAGE_ARR	Station SP5	commande demarrage arret p1	non	TELEC	0	
23/05/2017	15:38:58	SP5B72		COMMANDE_DEMARRAGE_ARR	Station SP5	commande demarrage arret p1	non		0	
23/05/2017	15:39:10	SP5B32		VANNE_1_OUVERTE_	Station SP5	vanne 1 ouverte	non		0	

Figure V-25 Fenêtre des alarmes et d’archives.

## **8. Conclusion**

On a édité un programme qui gère la station de pompage SP5 à base d'un API SIEMENS S7-1200 ainsi qu'une supervision globale et détaillée de différents éléments du process à travers l'interface graphique TP900 COMFORT, le programme de fonctionnement de la station et la supervision ont été développés à l'aide du logiciel TIA Portal.

On a aussi développé l'application de supervision et commande à distance à l'aide du logiciel TOPKAPI en utilisant la communication 3G.

On a configuré les routeurs pour la communication 3G entre la station et la supervision, ensuite on a réalisé une maquette de l'armoire d'automatisme de la station de pompage dont les éléments essentiels sont l'API, l'HMI sur lesquels on a chargé le programme, simulé son fonctionnement et finalement tester la télégestion.

## **Conclusion générale :**

Le travail présenté a pu répondre à un besoin industriel, qui est l'automatisation et la télégestion d'une station de pompage. La gestion de cette station est assurée par un automate Siemens S7-1200, qui forme une unité de traitement et de commande d'une grande flexibilité, qu'on a pu exploiter d'une manière très optimale grâce au logiciel de programmation et de supervision TIA Portal. La supervision de la station de pompage est assurée à travers une interface HMI TP900 COMFORT.

Pour la télégestion et le transfert de données entre la station de pompage et la supervision à distance, on a choisi le logiciel TOPKAPI.

Ce projet nous a permis de concevoir de nouvelles compétences dans la programmation et la réalisation des interfaces HMI ainsi on a développé nos connaissances dans l'instrumentation et le câblage électrique ainsi que les réseaux industriels. On a appris à résoudre divers problèmes, à gérer le temps, à travailler en équipe développer, un esprit d'analyse et la mise en application des connaissances acquises.

En fin, nous espérons que notre travail sera utile aux étudiants en automatique, afin d'appliquer leurs connaissances théoriques et apprendre à maîtriser la programmation et la supervision sous TIA Portal.

Dans des travaux futurs on pourra envisager :

-développer une communication entre les intersites de la station (réservoirs et forage ) à base de la 3G pour assurer le transfert des données nécessaire pour le bon fonctionnement de la station SP5 .

- Améliorer la supervision par l'ajout de fonction de notification par envoi de sms et mail.



## **Bibliographie :**

- [1] M. BERTRAND, « Automates programmables industriels », Techniques de l'ingénieur, Vol. S 8 015.
- [2] A. GONZAGA, « Automates programmables industriels », publié le 7\_11\_2004 geea.org
- [3] P. HOARAU, « API ».
- [4] SIEMENS, « contrôleurs SIMATIC», SIMATIC 2013.
- [5] SIEMENS, « contrôleurs SIMATIC», SIMATIC 2011.
- [6] N.JOUVRAY, « langages de programmation pour systèmes automatisés : norme CEI61131-3 », Technique de l'ingénieur, Vol2. S 8030.
- [7] SIEMENS, « Fiche technique 6ES7214-1HG40-0XB0», SIMATIC 2017.
- [8] SIEMENS, «SIMATIC S7 1200 modules TOR », 2013.
- [9] SIEMENS, « SIMATIC Step7 dans le TIA Portal », SIMATIC 2013.
- [10] SIEMENS, « SIMATIC Wincc dans le TIA Portal », SIMATIC 2013.
- [11] Help TIA Portal.
- [12] J-P. THOMESSE, « Réseaux locaux industriels », Techniques de l'ingénieur, Vol. S 7574
- [13] E.BAJIC et B.BOUARD, « Réseau PROFIBUS », Techniques de l'ingénieur, Vol. S 8 190.
- [14] PROFIBUS International Centre technique Allemagne.
- [15] <http://www.topkapi-scada.com>
- [16] G. ASCH, «Les capteurs en instrumentation industrielle », Edition DUNOD 2010

## **Annexe A: SEAAL**

### **1. Présentation SEAAL :**

La Société des Eaux et de l'Assainissement d'Alger (SEAAL), est une Société Publique par actions, détenue à 70% par l'Algérienne des Eaux (ADE) et à 30% par l'Office National de l'Assainissement (ONA).

La principale mission de SEAAL est de produire et de desservir en Eau potable les wilayas d'Alger et de Tipasa, et de collecter et traiter les Eaux usées au niveau de ces deux périmètres. Elle dessert ainsi 3.8 millions d'habitants soit environ 10% de la population Nationale.

SEAAL assure le service auprès de 680.000 clients (ménages, administrations, commerces, industriels et sites touristiques) et le nombre de ses salariés s'élève à 6.202.

SEAAL est née en 2006 de la volonté politique des Autorités Algériennes d'améliorer rapidement la qualité et le cadre de vie des citoyens, en particulier dans la Capitale.

### **2. Défis et enjeux :**

Les défis de SEAAL consistaient, d'une part, à moderniser rapidement le service (eau, assainissement, clientèle, patrimoine...) à un niveau de standard international, et, d'autre part, d'autonomiser, à terme, la structure publique grâce à un transfert de savoir-faire de SUEZ Environnement au profit des équipes locales. Le dispositif organisationnel mis en place assure une communication transparente, claire et fiable entre les acteurs pour atteindre une confiance durable.

### **3. Les valeurs :**

Les valeurs à travers lesquelles se définit SEAAL procèdent de la mise en place d'une véritable « Culture d'entreprise » où la notion d'appartenance prend tout son sens. Le développement de cette « Culture d'entreprise » ne pouvait se concrétiser qu'à travers le partage de références incontournables telles que la compréhension des enjeux de la modernisation du Service Public, la mobilisation pour en améliorer la performance et l'exigence de donner une image correspondant aux attentes des clients. Cela s'est matérialisé par trois Valeurs qui résument l'identité de SEAAL dans laquelle chacun de nos collaborateurs se retrouve :

- Le Professionnalisme, générateur de reconnaissance.
- Le Dynamisme, vecteur de visibilité et donc de fierté.

- La Confiance réciproque
- Ces valeurs sont l'aboutissement d'une démarche de concertation au sein de l'entreprise, seule à même de garantir une adhésion de l'ensemble des collaborateurs.

#### **4. Les métiers de SEAAL :**

**4.1. Production :** La première mission, capter l'eau dans son milieu naturel, la traiter et la stocker en veillant en continu à sa qualité. Le challenge est de préserver les ressources en eau et d'assurer leur disponibilité pour les générations actuelles et futures. Pour ce faire, l'Etat et SEAAL œuvrent au quotidien à la sécurisation des capacités de production et de transfert et à la gestion à long terme des ressources.

**4.2. Distribution :** Exploiter, entretenir et moderniser 6 455 km de réseaux de distribution jusqu'aux robinets des clients, mais également leur apporter une écoute de chaque instant et un service d'une qualité irréprochable.

**4.3. Assainissement :** Collecter puis dépolluer les eaux usées, mais aussi améliorer sans cesse la qualité des eaux de baignade et préserver la biodiversité : une mission essentielle pour laquelle la SEAAL œuvre au quotidien.

## Annexe B : Démarreur progressif ALTISTART 22

### B.1. Démarreur progressif :

Les démarreurs de la station SP5 sont de type Altistar22 qui est un démarreur-ralentisseur de Schneider Electric .L'ATS22 offre une commande d'accélération et de décélération pour les moteurs à induction asynchrones triphasés standard (à cage d'écreuil). Il contrôle les performances du moteur en fonction du couple moteur et non en fonction d'un simple contrôle de tension ou de courant. Des algorithmes de contrôle avancés permettent un meilleur contrôle pendant la phase de démarrage et réduisent les instabilités mécaniques en fin de démarrage. L'ATS22 est disponible dans 15 calibres de 17 à 590 A et est dimensionné pour fonctionner sur des moteurs de 208 à 600 V. Il s'ajuste automatiquement sur la fréquence du réseau.

### B.2. Spécifications électriques des démarreurs ATS22 :

Borne	Fonction	Spécifications
CL1	Alimentation puissance de l'ATS22	230 V CA, +10 %
CL2		220 V CA, -15 %
R1B	Relais 1 normalement fermé	Capacité de commutation maximale : 5 A – 250 V CA ou 30 V CC sur charge résistive (p.f. = 1) 2 A – -250 V CA ou 30 V CC sur charge inductive (p.f. = 0,4) Capacité de commutation minimale : 100 mA, 12 V CC
R1C	Relais 1, commun	
R1A	Relais 1 normalement ouvert	
R2B	Relais 2 normalement fermé	
R2C	Relais 2, commun	
R2A	Relais 2 normalement ouvert	
LI1	Entrée logique 1	3 entrées logiques 24 V avec impédance de 4,3 kΩ U <sub>max</sub> = 30 V, I <sub>max</sub> = 8 mA État 1 : U > 11 V - I > 5 mA État 0 : U < 5 V - I < 2 mA L'alimentation 24 V est limitée à 42 mA (pour utilisation interne et externe). Précision 24 V CC : 24 V, ±6 V CC Délai d'activation/de désactivation : • Matériel : < 15 ms • Logiciel : < 70-85 ms (anti-rebond)
LI2	Entrée logique 2	
LI3	Entrée logique 3	
+24 V CC	24 V CC flottant (+) (1)	
COM	24 V CC flottant (-)	
PTC1	PTC (+)	Connexion des sondes PTC : Résistance totale du circuit de sondes : 750 Ω à 25 °C (77 °F).
PTC2	PTC (-)	
(⊕)	Terre (blindage)	Connecteur Modbus RJ45 pour : • Terminal déporté • Logiciel SoMove • Bus de communication
RJ45 broche 1	Non raccordée	
RJ45 broche 2	Non raccordée	
RJ45 broche 3	Commun	
RJ45 broche 4	D1	
RJ45 broche 5	D0	
RJ45 broche 6	Non raccordée	
RJ45 broche 7	12 ±0,5 V CC (2)	
RJ45 broche 8	Commun	
RJ45 blindé	Mise à la terre du signal (SNG)	

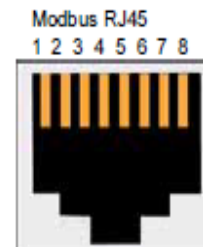


Tableau B-1 Spécification électriques des démarreurs ATS22.

### B.3.Types de commandes

**LI1 sur stop :** La borne LI1 est définie sur stop. Ce réglage n'est pas modifiable via l'IHM ou une liaison série.

Cette entrée est active sur le niveau (niveau bas (0) = stop).

**Commandes de marche et de démarrage :** Ces deux commandes peuvent être affectées uniquement à la borne LI2 (mais pas à la borne LI3).

**Marche en mode 2 fils :** à la mise sous tension ou en cas de réinitialisation manuelle après un défaut, le moteur redémarre si la commande de marche est active.

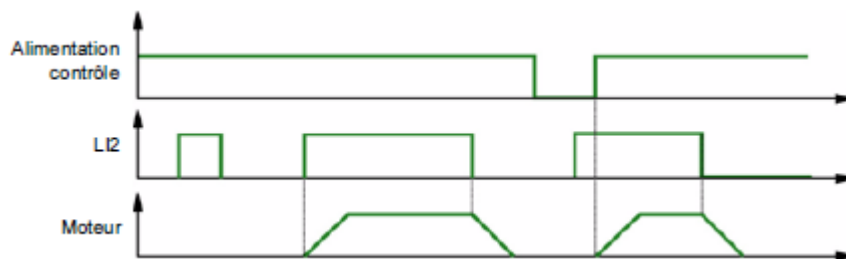


Figure B-2 commande en mode deux fils.

### Marche en mode 3 fils

À la mise sous tension, en cas de reset défaut ou après ordre d'arrêt ou après la modification d'une affectation, le moteur peut uniquement être mis en marche une fois la commande de démarrage exécutée (état 0), suivie d'une nouvelle impulsion (état 1).

Lorsque vous passez d'un fonctionnement à distance à un fonctionnement local et qu'une commande de marche est active sur le terminal, le moteur ne démarre pas en mode 3 fils. La commande de marche doit être supprimée puis de nouveau exécutée.

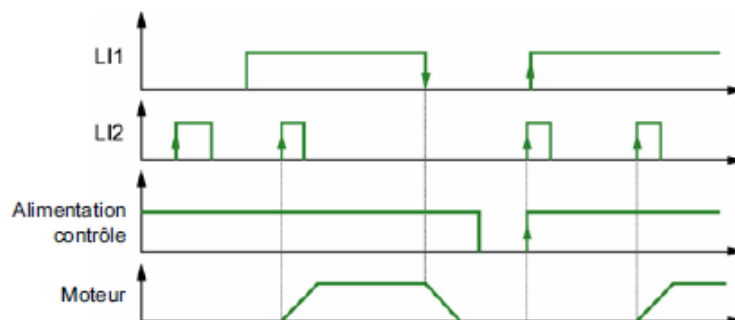
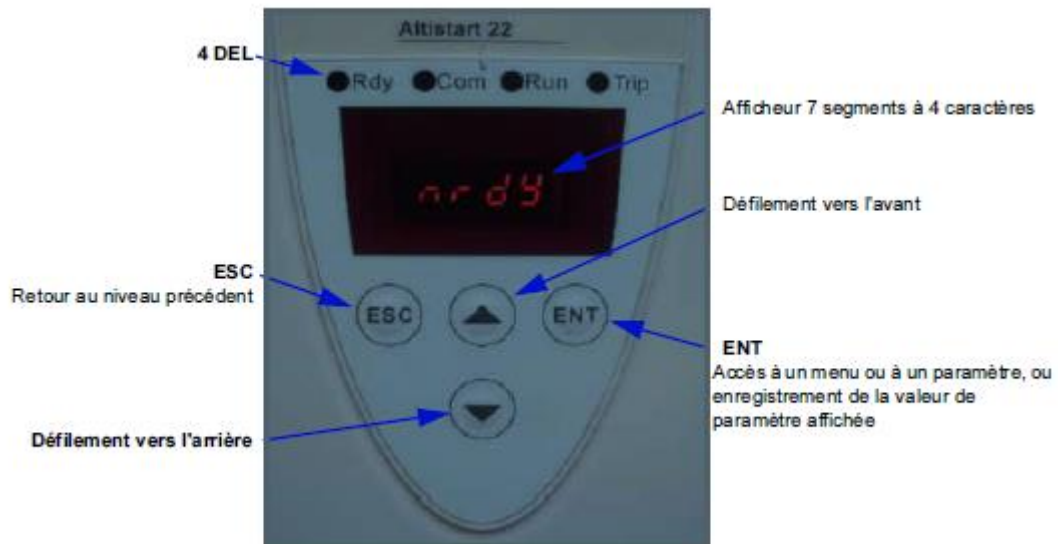


Figure B-3 commande en mode trois fils.

## B.4. Présentation de l'afficheur et des touches fonctions :



### Combinaisons spéciales de touches

Certaines combinaisons de touches sont utilisées comme raccourcis :

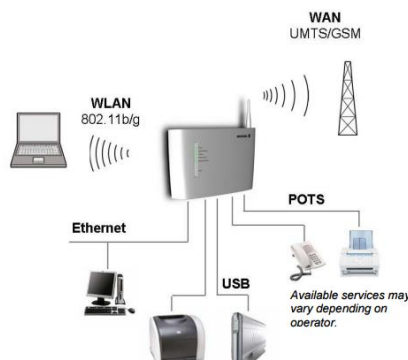
Combinaison de touches	Description
ESC + ▲	Affichage du menu <i>U t I L</i> (Utilitaire)
ENT + ▲ + ▼	Effacement du message de défaut et réinitialisation du démarreur progressif
ESC + ▲ + ▼	Démarreur progressif non verrouillé (voir le paramètre <i>L o d</i> )

## Annexe C : Routeur ERICSSON

### Le routeur Ericsson W 25 :

L'Ericsson W25 est un routeur maison avec accès Internet sans fil. Les principaux avantages du produit sont les suivants:

- Services haut débit sans fil à haute vitesse fiable L'Ericsson W25 utilise l'accès radio WCDMA / HSDPA pour fournir une vitesse élevée de données au réseau local.
- Large gamme de services vocaux L'Ericsson W25 comprend des fonctions vocales accessibles par rapport à la norme Interfaces téléphoniques analogiques. Services réseau tels que prépayés Abonnements, Identification de l'appelant (1er appel uniquement), CLIP, Appel en attente, Suppression d'appel, Le renvoi d'appel et les appels conférence multipartite sont pris en charge.
- Fax L'Ericsson W25 prend en charge les fonctions de télécopie sur IP (T.38) en utilisant un paquet Connexion commutée.
- Réseautage avancé L'Ericsson W25 prend en charge une configuration de réseau local avancée. Il Fournit un routage IP, une commutation Ethernet, un service DHCP et un NAT.
- Partage de fichiers et d'imprimantes Le partage de fichiers et de fichiers réseau est pris en charge via une connexion USB.
- Accès sans fil local avec sécurité de haut niveau L'Ericsson W25 est un point d'accès WLAN pour les locaux sans fil Réseau prenant en charge WEP, WPA et WPA2 avec des clés pré-partagées pour Sécurité WLAN.



*Figure C-1 Connexions réseau.*

### *Les Indicateurs :*

L'Ericsson W25 est équipé de six indicateurs d'état situés sur le devant. Une description générale de chaque indicateur est fournie dans le tableau suivant (de haut en bas):

<b>Texte</b>	<b>Statut</b>	<b>Description</b>
power	Vert Rouge Non éclairé	Alimenté par réseau. Alimenté par batterie. L'alimentation est éteinte.
Mobile Network	Vert Rouge clignotant Non éclairé	Connexion à un réseau UMTS. Connexion à un réseau GSM. Recherche d'une connexion. Pas de connexion au réseau mobile.
internet	Vert Non éclairé	Connexion Internet établie. Pas de connexion Internet.
Wireless LAN	Vert Non éclairé	Le LAN sans fil est actif. Le réseau local sans fil est inactif.
Message Waiting	clignotant Non éclairé	Nouveau message (s) de messagerie vocale reçu. Pas de nouveaux messages vocaux.
Alarme	Rouge clignotant	erreur. Pas d'erreur.

*Tableau C- 1 description des indicateurs de l'ERICSSON.*

### **Connecteurs :**

Les connecteurs sur l'Ericsson W25 sont positionnés comme indiqué dans la figure suivante:

<b>Connecteurs</b>	<b>Description</b>
10-28 VDC.	Alimentation pour connecter l'adaptateur.
Bouton de réinitialisation	utilisé pour restaurer la configuration Ericsson W25 aux Paramètres d'usine.
Connecteurs USB	pour connecter un (des) périphérique (s) USB au Ericsson W25 (pour le partage de fichiers et d'imprimantes).



Téléphone (à gauche)	Connecteur téléphonique, pour connecter un téléphone à l'Ericsson W25. Trois téléphones maximum peuvent être connectés. Afin d'éviter les interruptions des fonctions téléphoniques.
Téléphone / Fax (à droite)	Connecteur téléphonique / fax, pour connecter un télécopieur à L'Ericsson W25. Le fax nécessite le service de télécopie IP T.38 Souscription ou offre offerte par le fournisseur de services.

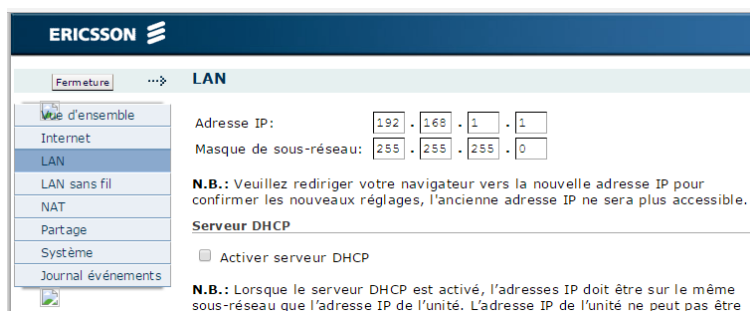
*Tableau C-2 description des connecteurs de l'ERICSSON.*

## Configuration du routeur :

1- Démarrez un navigateur sur un PC connecté à l'Ericsson W25 et tapez Http://192.168.1.1 dans le champ Adresse (URL). Après l'introduction du nom utilisateur (user) et le mot de passe (user) la page d'accueil du matériel s'affiche.



2- Afin de changer l'adresse IP du matériel on clique sur LAN pour et on introduit une nouvelle adresse IP, ensuite Cliquez sur (appliquer).



### 3- Pour configurer la table de routage on clique sur NAT Ajouter une redirection

**ERICSSON**

Fermeture ...> **NAT**

Vue d'ensemble  
Internet  
LAN  
LAN sans fil  
**NAT**  
Partage  
Système  
Journal événements

La traduction d'adresse réseau "Network Address Translation" (NAT) permet au réseau LAN d'utiliser plusieurs adresses IP pour le trafic interne et une adresse IP publique pour le trafic externe.

Activer UPnP IGD

Protocoles utilisés avec la passerelle d'application ALG (Application Level Gateway):

FTP  
 TFTP

Appliquer

---

**REDIRECTION DE PORT**

Utilisez la redirection de port pour permettre à une application Internet spécifique d'accéder à un système de réseau local.

Ajouter une redirection

Port NAT	IP serveur	Port serveur	Protocole	Supprimer	Éditer
501	192.168.1.10	501	tcp	Supprimer	Éditer
503	192.168.1.10	503	tcp	Supprimer	Éditer
504	192.168.1.10	504	tcp	Supprimer	Éditer
505	192.168.1.10	505	tcp	Supprimer	Éditer
506	192.168.1.10	506	tcp	Supprimer	Éditer
507	192.168.1.10	507	tcp	Supprimer	Éditer
80	192.168.1.10	80	tcp	Supprimer	Éditer
700	192.168.1.10	700	tcp	Supprimer	Éditer

© Copyright 2008 Ericsson AB. Tous droits réservés.

4- On insère le protocole de communication, port NAT, IP serveur, port serveur et on clique sur Appliquer.

**ERICSSON**

Fermeture ...> **Ajouter redirection de port**

Protocole:

Port NAT:

IP serveur:

Port serveur:

Appliquer Annuler

5- Pour configurer les paramètres de la puce 3G on clique sur Network -> Modem, et on introduit les données fournies par l'opérateur réseau.

id=6893d6e9e428a07202b04d61b58d7222f34084208;page=4&lang=fr

**AUTHENTIFICATION**

PIN: La carte SIM ne requiert pas d'authentification PIN

APN auto

APN (3G):

APN (2G):

Nom d'utilisateur PPP:

Mot de passe PPP:

Protocole d'authentification  PAP  CHAP

Appliquer

---

**SÉLECTION DE BANDE CELLULAIRE**

Région

Paramètres 3G:

850  
 1900  
 2100

Paramètres 2G:

850  
 900  
 1800  
 1900

Appliquer

© Copyright 2008 Ericsson AB. Tous droits réservés.

## Annexe D : TIA Portal

### D.1. Les blocs d'organisation dans le TIA Portal pour les S7-1200/1500 :

e- **Blocs d'organisation pour le traitement cyclique du programme (S7-1200/1500)**

Ensemble des blocs programmés utilisateur dans lequel on a programmé des instructions ou appelé d'autres blocs et qui doivent être traités cycliquement. L'OB1 est le bloc par défaut pour réaliser l'exécution cyclique du programme utilisateur. Classe d'évènement : Program cycle, Name: Main (OB1)

f- **Blocs d'organisation pour le démarrage (S7-1200/1500)**

Le système d'exploitation appelle une seule fois les OB de démarrage lors du passage de l'état "ARRET" à l'état "MARCHE".

Classe d'évènement : Startup (OB100)

g- **Blocs d'organisation pour les alarmes temporisées (S7-1200/1500)**

Après l'écoulement d'un temps défini, un OB d'alarme temporisée interrompt le traitement du programme cyclique.

Classe d'évènement : Time delay interrupt (OB20)

h- **Blocs d'organisation pour les alarmes cycliques (S7-1200/1500)**

Les OB d'alarmes cycliques interrompent le traitement du programme cyclique à intervalles de temps définis.

Classe d'évènement : Cyclic interrupt (OB30)

i- **Blocs d'organisation pour les alarmes de processus HSC (S7-1200/1500)**

Les OB d'alarme de processus interrompent le programme cyclique lors de l'apparition d'un évènement matériel. Les alarmes de processus ne sont pas exclusives aux "alarmes de processus HSC" mais également pour les alarmes de processus des voies TOR.

Classe d'évènement : Hardware interrupt (OB40)

j- **Bloc d'organisation pour l'alarme de diagnostic (S7-1200/1500)**

L'OB d'alarme de diagnostic interrompt le programme cyclique, lorsqu'un module apte à l'émission d'un diagnostic et validé pour un tel diagnostic a détecté une erreur.

Classe d'évènement : Diagnostic error interrupt (OB82)

k- **Bloc d'organisation pour l'erreur de temps (S7-1200/1500)**

Si le temps de cycle maximal est dépassé, l'OB d'erreur de temps interrompt le programme cyclique.

Classe d'évènement : Time error interrupt (OB80)

**l- Bloc d'organisation pour le débrogage et l'enfichage (S7-1500)**

Le système d'exploitation de la CPU appelle l'OB lorsqu'un module ou sous-module configuré et non désactivé de la périphérie décentralisée est débrogé ou enfiché.

Classe d'évènement : Pull or plug of modules (OB83)

**m- Bloc d'organisation pour l'erreur de programmation (S7-1500)**

Le système d'exploitation de la CPU appelle l'OB d'erreur de programmation lorsqu'une erreur de programmation survient lors du traitement d'une instruction du programme utilisateur.

Classe d'évènement : Programming error (OB121)

**n- Bloc d'organisation pour l'erreur de châssis d'un système PROFINET IO (S7-1500)**

En cas de défaillance d'un système maître DP, d'un esclave ou d'une partie d'un sous-module, le système d'exploitation de la CPU appelle l'OB.

Classe d'évènement : Rack or station failure (OB86)

**o- Bloc d'organisation pour l'erreur d'accès à la périphérie (S7-1500)**

Le système d'exploitation de la CPU appelle l'OB d'erreur d'accès à la périphérie lorsqu'une erreur d'accès direct aux données de la périphérie survient lors du traitement d'une instruction du programme utilisateur.

Classe d'évènement : IO access error (OB122)

**p- Blocs d'organisation pour les alarmes horaires (S7-1500)**

Les OB peuvent être appelés périodiquement ou une fois à un moment précis.

Classe d'évènement : Time of day (OB10 bis OB17)

**q- Bloc d'organisation pour MC-Interpolator (S7-1500)**

Quand on crée un objet technologique (Motion), le bloc d'organisation "MC-Interpolator" pour le traitement des objets technologiques (Motion) est appelé.

Classe d'évènement : MC-Interpolator (OB92)

**r- Bloc d'organisation pour MC-Servo (S7-1500)**

Quand vous créez un objet technologique (Motion), le bloc d'organisation "MC-Servo" pour le traitement des objets technologiques (Motion) est appelé.

Classe d'évènement : MC-Servo (OB91)

**s- Bloc d'organisation pour l'alarme d'isochronisme (S7-1500)**

Vous pouvez démarrer des sections de programme de manière isochrone par rapport à la cadence DP ou la cadence d'émission PN.

Classe d'évènement : Synchronous (OB61)

t- **Bloc d'organisation pour l'alarme d'état (S7-1500)**

Le système d'exploitation de la CPU appelle l'OB lorsqu'il reçoit une alarme d'état.

Classe d'évènement : Status (OB55)

u- **Bloc d'organisation pour l'alarme de mise à jour (S7-1500)**

Le système d'exploitation de la CPU appelle l'OB lorsqu'il reçoit une alarme de mise à jour.

Classe d'évènement : Update (OB56)

v- **Bloc d'organisation pour alarme spécifique profil ou fabricant (S7-1500)**

Le système d'exploitation de la CPU appelle l'OB lorsqu'il reçoit une alarme spécifique profil .

Classe d'évènement : Profile (OB57)

## **D.2. Temporisation dans le TIA Portal :**

5) Temporisation à l'enclenchement : **Ton.**

6) Temporisation au déclenchement: **Toff.**

7) Génération d'une impulsion: **Tpulse**

### **1.1.Description des différentes temporisations :**

- **Ton** : La sortie de cette temporisation passe à « 1 » après une durée définie pour autant que son entrée soit toujours à « 1 ». Aussitôt que l'entrée revient à « 0 », la sortie revient à « 0 ». L'équivalent de cette temporisation est le relais temporisé à l'enclenchement.
- **Toff** : La sortie de cette temporisation passe immédiatement à « 1 » aussitôt que son entrée est activée. La sortie revient à « 0 » après une durée définie, à partir du moment où l'entrée n'est plus activée. L'équivalent de cette temporisation est le relais temporisé à la chute.
- **Tpulse** : Cette temporisation génère une pulse de durée fixe à partir d'un front positif sur son entrée. Elle n'a pas d'équivalent en logique câblée classique.

### **1.2.Format du temps :**

Pour toute les temporisations CEI, le format est le suivant : T#.....s, La durée est affectée à l'entrée « PT » (Pulse Time).

## **D.3. Les Compteurs CEI dans le TIA Portal:**

L'automatisation des machines nécessite de compter le nombre de pièces mises dans un emballage, le nombre de cycles, le nombre d'événements particuliers, etc. L'opération logique qui assure cette fonction est appelée Compteur. Pour être en conformité avec la norme CEI on utilise les compteurs respectant cette norme. Le logiciel TIA met à disposition trois types de compteurs.

- Compteur (CTU : Counter UP)
- Décompteur (CTD : Counter DOWN)
- Compteur/Décompteur (CTUD : Counter UP/DOWN)

On peut utiliser un grand nombre de compteurs. Cependant il faut à chaque fois leur assigner un DB différent.

- **Compteur CTU :**

-**CU** : Entrée de comptage, lorsque l'état logique de l'entrée CU passe de l'état 0 à l'état 1 (front montant), le compteur est incrémenté de 1.

-**R** : Entrée reset, lorsque l'état logique de l'entrée R passe de l'état 0 à l'état 1, la valeur du compteur est forcée à 0 quelle que soit l'état de l'entrée CU.

-**PV** : Réglage de la valeur du compteur.

-**CV** : Mot de memento pour l'affichage de la valeur du compteur ou l'utilisation de la valeur du compteur ailleurs dans le programme.

-**Q**: est à l'état logique 1 si la valeur de comptage actuelle (CV) est égale ou supérieure à la valeur allouée à l'entrée PV. ( $CV \geq PV$ )

- **Compteur CTD :**

Description du compteur CTD :

- **CD** : Entrée de décomptage, lorsque l'état logique de l'entrée CD passe de l'état 0 à l'état 1 (front montant), le compteur est décrémenté de 1.

**LD** : Entrée « chargé », lorsque l'état logique de l'entrée LD passe de l'état 0 à l'état 1, la valeur du compteur (CV) est forcée à la valeur réglée à l'entrée PV quelle que soit l'état de l'entrée CD.

-**PV** : Réglage de la valeur du compteur.

-**CV** : Mot de memento pour l'affichage de la valeur du compteur ou l'utilisation de la valeur du compteur ailleurs dans le programme. Attention de ne pas réutiliser les bits de mémoire M101.0 à M102.7 pour autre chose dans le programme.

-**Q** : La sortie Q, est à l'état logique 1 si la valeur de comptage actuelle (CV) est égale ou inférieure à 0. ( $CV \leq 0$ )

- **Compteur décompteur CTUD :**

- **CU** : Entrée de comptage, lorsque l'état logique de l'entrée CU passe de l'état 0 à l'état 1 (front montant), le compteur est incrémenté de 1.

- **CD** : Entrée de décomptage, lorsque l'état logique de l'entrée CD passe de l'état 0 à l'état 1 (front montant), le compteur est décrémenté de 1.

- **R** : Entrée reset, lorsque l'état logique de l'entrée R passe de l'état 0 à l'état 1, la valeur du compteur est forcée à 0 quelle que soit l'état de l'entrée CU.

- **LD** : Entrée « chargé », lorsque l'état logique de l'entrée LD passe de l'état 0 à l'état 1, la valeur du compteur (CV) est forcée à la valeur réglée à l'entrée PV quelle que soit l'état de l'entrée CD.

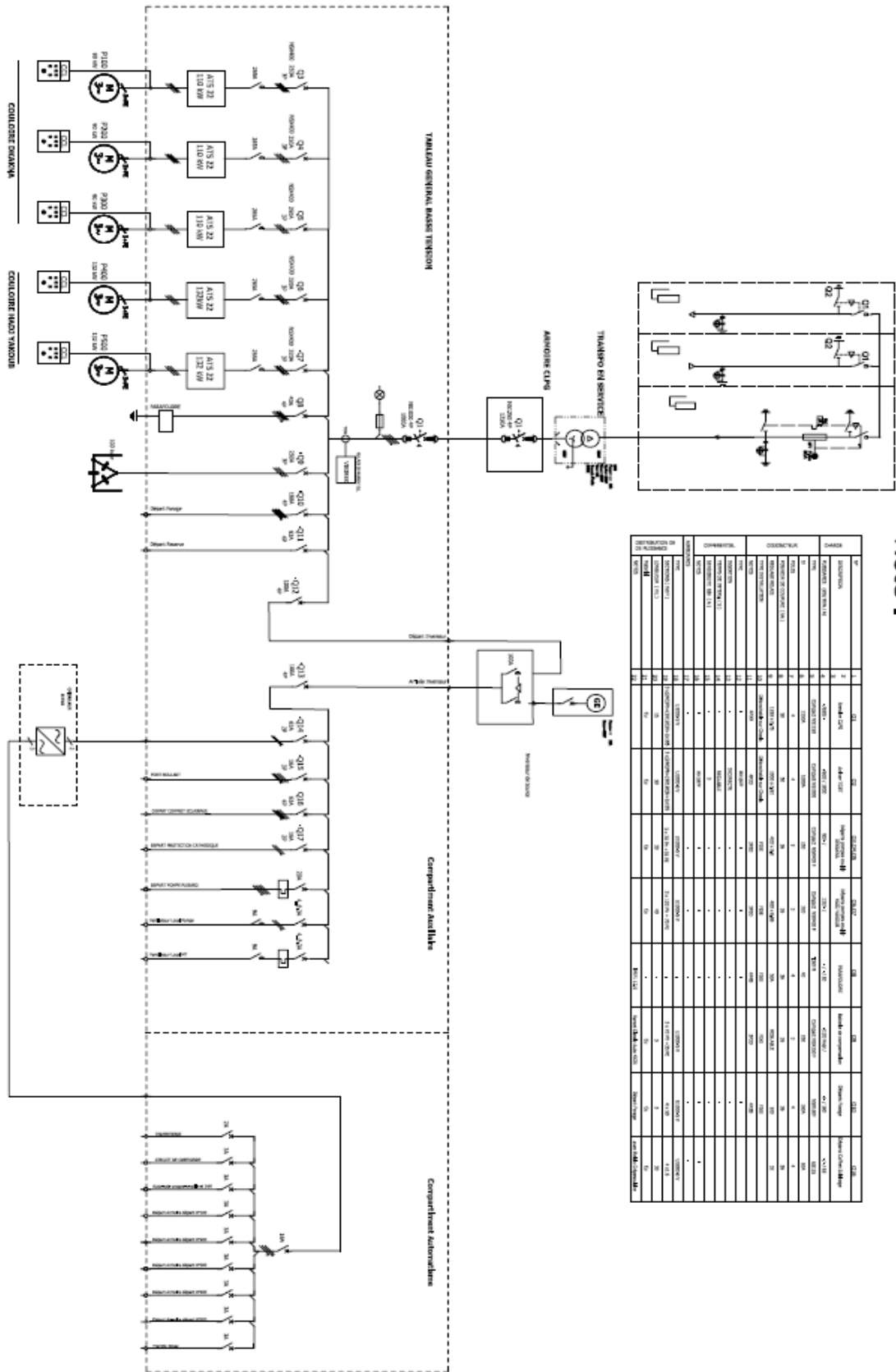
- **PV** : Réglage de la valeur du compteur. (Dans l'exemple ci-dessus 10)

- **QD** : La sortie Q, est à l'état logique 1 si la valeur de comptage actuelle (CV) est égale ou inférieure à 0. ( $CV \leq 0$ )

- **CV** : Mot de memento pour l'affichage de la valeur du compteur ou l'utilisation de la valeur du compteur ailleurs dans le programme. Attention de ne pas réutiliser les bits de mémoire M101.0 à M102.7 pour autre chose dans le programme.

- **QU** : La sortie Q, est à l'état logique 1 si la valeur de comptage actuelle (CV) est égale ou supérieure à la valeur allouée à l'entrée PV. ( $CV \geq PV$ )

# Annexe E : Schéma unifilaire de la station SP5



NO	DESCRIPTION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	COOLERS SWANKA	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31	Q32	Q33	Q34	Q35	Q36	Q37	Q38	Q39	Q40	Q41	Q42	Q43	Q44	Q45	Q46	Q47	Q48	Q49	Q50	Q51	Q52	Q53	Q54	Q55	Q56	Q57	Q58	Q59	Q60	Q61	Q62	Q63	Q64	Q65	Q66	Q67	Q68	Q69	Q70	Q71	Q72	Q73	Q74	Q75	Q76	Q77	Q78	Q79	Q80	Q81	Q82	Q83	Q84	Q85	Q86	Q87	Q88	Q89	Q90	Q91	Q92	Q93	Q94	Q95	Q96	Q97	Q98	Q99	Q100	
2	COOLERS HAO YAKOWA	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31	Q32	Q33	Q34	Q35	Q36	Q37	Q38	Q39	Q40	Q41	Q42	Q43	Q44	Q45	Q46	Q47	Q48	Q49	Q50	Q51	Q52	Q53	Q54	Q55	Q56	Q57	Q58	Q59	Q60	Q61	Q62	Q63	Q64	Q65	Q66	Q67	Q68	Q69	Q70	Q71	Q72	Q73	Q74	Q75	Q76	Q77	Q78	Q79	Q80	Q81	Q82	Q83	Q84	Q85	Q86	Q87	Q88	Q89	Q90	Q91	Q92	Q93	Q94	Q95	Q96	Q97	Q98	Q99	Q100	