

**République Algérienne Démocratique et Populaire**  
وزارة التعليم العالي والبحث العلمي  
**Ministère de l'Enseignement Supérieur Et de La Recherche Scientifique**

**Faculté des Sciences et  
Technologies**

جامعة غرداية

كلية العلوم والتكنولوجيا

**Département Sciences et  
Technologies**



قسم العلوم والتكنولوجيا

**Université de  
Ghardaïa**

**Mémoire en vue de l'obtention du diplôme de  
Master académique en Automatique  
Spécialité : Automatique**

**THEME**

**Gestion Et Supervision D'une Station De  
Pompage à Base D'automate SIEMENS**

**Présenté par**

- MOSBAH Faysal
- SEBGAG Taib Abdelmadjid

**Membres du jury**

**Grade**

KIFOUCHE Abdessalam

**Grade MAA**

**Examineur**

BOUKHARI Hamed

**Grade MCB**

**Examineur**

Fihakhir Amine Mahdi

**Grade MCB**

**Encadreur**

**Juin 2021**

# *Dédicace*

*Je dédie ce modeste travail*

*À ma mère, mon père, qui m'a encouragé à aller de l'avant*

*À mes chères sœurs et à mes chers frères*

*À tous mes proches et ceux qui mes sont chers*

*À tous les membres de la famille MOSBAH*

*À toutes les personnes que je porte dans mon cœur*

*À tous mes amis sans exception*

*Aux étudiants de la spécialité Automatique PROMO 2021*

*À mes enseignants durant mon cursus scolaire, du primaire à*

*L'université*

*À mon Binôme et sa famille*

*Merci à toutes les personnes qui m'ont aidé de près ou de loin.*

**FAYSAL**

# *Dédicace*

*Je dédie ce modeste travail particulièrement*

*À ma douce maman*

*À mon père, pour tous ce qu'il a faits pour moi*

*À toute ma famille en particulier*

*À mes frères, mes sœurs*

*À mon binôme et sa famille*

*À tout ce qui m'encourage toujours de près et de loin.*

*TAYEB*

# *Remerciements*

*De par le nom de Dieu tout miséricordieux et tout compatissant. Par-dessus tout, nous tenons à remercier Dieu tout-puissant de nous avoir armé de courage et de patience pour y parvenir au bout de ce modeste travail.*

*Nous tenons à exprimer nos vifs remerciements à notre promoteur Monsieur FIHAKHIR AMINE MAHDI pour nous avoir encadrés durant notre projet de fin d'études et nous conseiller tout le long de notre travail.*

*Nous tenons aussi à remercier chaleureusement les membres de jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre projet.*

*Enfin, nous exprimons notre profonde gratitude à notre famille, ainsi qu'à tous ceux qui nous ont aidés de loin ou de près à faire ce travail. Nous mentionnons en particulier M. BOUZID Ahmed Amin, ingénieur PLC, pour son encadrement et ces conseils.*

*FAYSAL & TAYEB*

# Résumé

Le travail présent dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables « SIEMENS ».

La station consiste un réservoir de stockage, des groupes électropompes, des vannes motorisées et un réservoir « anti-bélier » pour protéger les pompes.

Notre mission est la reprogrammation de la gestion des stations de pompage hydraulique avec le logiciel « TIA PORTAL » et la supervision avec « WinCC HMI ». Utilisation de l'automate « S7-1200 ».

La station de pompage étudié est située à KOUBA Alger .

**MOTS CLES :** Station de pompage, API, Programmation, TIA portal, WinCC, PLC, Siemens, S7-1200

## ملخص

العمل المنجز في هذه المذكرة يتمحور أساسا حول إعادة برمجة إدارة محطة لضخ المياه وذلك باستخدام برمجة « SIMIENS » و باستخدام برنامج «TIA Portal». وتصميم شاشة إشراف من خلالها نستطيع مراقبة جميع أجهزة المحطة باستخدام برنامج "«WinCC IHM»".

تتكون المحطة من خزان تخزين ومضخات كهربائية وصمامات آلية وخزان "هوائي مائي" لحماية المضخات.

نوع آلي البرمجة المستخدم هو «S7-1200». محطة الضخ المدروسة تقع في بلدية القبة بالجزائر.

**كلمات مفتاحية:** محطة الضخ، مسير صناعي مبرمج سيمنس، برنامج تيا بورتال، وحدة تحكم منطقية قابلة للبرمجة، آلي البرمجة S7-1200 .

# Table des matières

<i>Dédicace</i> .....	i
<i>Dédicace</i> .....	ii
<i>Remerciements</i> .....	iii
Résumé.....	ii
Table des matières.....	iii
Liste des figures.....	vii
Liste des tableaux.....	ix
Liste des abréviations.....	1
Introduction générale.....	2
Chapitre 1 : Les Automates Programmables Industriels.....	3
1.1 Introduction.....	4
1.2 Historique.....	4
1.3 Définition de l'API.....	4
1.4 Architecture des automates programmables industriels.....	5
1.5 Structure interne des automates programmables.....	6
1.5.1 Une unité de traitement ou processeur.....	6
1.5.2 Une mémoire programme.....	7
1.5.3 Une mémoire de données.....	7
1.5.4 Une interface d'entrée.....	7
1.5.5 Une interface de sortie.....	7
1.5.6 Un module d'alimentation.....	7
1.6 Les automates Siemens.....	8
1.7 Introduction à l'automate S7-1200.....	8

1.8	Présentation de la CPU 1214C .....	9
1.8.1	Caractéristiques techniques de la CPU .....	10
1.8.2	Modes de fonctionnement de la CPU .....	10
1.9	Conclusion .....	12
Chapitre 2 : Les Logiciels Programmables .....		13
2.1	Introduction .....	14
2.2	Description du logiciel TIA Portal .....	14
2.2.1	Les avantages du logiciel TIA Portal .....	14
2.3	SIMATIC STEP 7 .....	15
2.3.1	Utilisation de STEP 7 .....	15
2.3.2	Vue du portail et vue du projet .....	16
2.3.3	Adressage des E/S .....	17
2.3.4	Mnémoniques .....	18
2.3.5	Mémentos .....	18
2.3.6	Types de blocs de programme .....	19
2.3.7	Les langages des programmations .....	21
2.4	La supervision .....	22
2.4.1	Introduction à SIMATIC HMI .....	22
2.4.2	SIMATIC WinCC (TIA Portal) .....	24
2.5	Conclusion .....	24
Chapitre 3 : Description De La Station De Pompage .....		25
3.1	Introduction .....	26
3.2	Notre objectif .....	26
3.3	Les équipements utilisés .....	27
3.3.1	Équipement MT (Moyen Tension) .....	27
3.3.2	Équipement électrique BT (Basse tension) .....	28

3.3.3	Le Pupitre de commande.....	31
3.3.4	Le groupe électropompe.....	32
3.3.5	La vanne .....	32
3.3.6	Le démarreur progressif SIRIUS.....	33
3.3.7	Variateur de vitesse SINAMICS V20 .....	34
3.3.8	Système anti-bélier .....	35
3.3.9	Les capteurs .....	36
3.3.10	Débitmètre.....	37
3.4	Algorithme de procès.....	37
3.5	Conclusion .....	41
Chapitre 4 : Programmation Et Supervision .....		42
4.1	Introduction .....	43
4.2	Programme du système anti bélier .....	43
4.2.1	Cahier de charge du Programme anti bélier.....	43
4.2.2	Le programme de l'anti bélier .....	44
4.2.3	La simulation de l'anti bélier .....	46
4.3	Le Programme gestion de la vanne .....	47
4.3.1	Cahier de charge du Programme Gestion de la vanne .....	47
4.3.2	Le programme de la gestion de la vanne.....	48
4.3.3	La simulation du Programme Gestion de la vanne.....	51
4.4	Le Programme autorisation du démarrage du GEP .....	52
4.4.1	Cahier de charge du Programme Autorisation.....	52
4.4.2	La programmation de l'Autorisation.....	54
4.5	Le Programme du Défaut .....	58
4.5.1	Cahier de charge du Défaut.....	58
4.5.2	La programmation du défaut.....	60



4.6	Le Programme Groupe Electropompe .....	63
4.6.1	Cahier de charge du GEP .....	64
4.6.2	Le programme du GEP .....	65
4.6.3	La simulation du Programme GEP .....	67
4.7	Conclusion .....	68
	Conclusion générale .....	69
	Bibliographies.....	70

# Liste des figures

<b>Figure 1-2</b> : Architecture d'un API.....	5
<b>Figure 1-3</b> : Structure interne d'un automate programmable .....	6
<b>Figure 1-1</b> : Automate Siemens.....	8
<b>Figure 1-4</b> : Automate programmable S7-1200.....	9
<b>Figure 1-5</b> : Fenêtre du contrôleur SIMATIC PLCSIM .....	11
<b>Figure 1-6</b> : Modes de fonctionnement de la CPU .....	11
<b>Figure 2-1</b> : Vue du portail.....	17
<b>Figure 2-2</b> : Adressage des E/S .....	18
<b>Figure 2-3</b> : Types de blocs de programme .....	19
<b>Figure 2-4</b> : Bloc de configuration .....	21
<b>Figure 2-5</b> : Tableau présente trois Langages de programmation des API.....	22
<b>Figure 2-6</b> : Interface Homme Machine.....	23
<b>Figure 2-7</b> : Vue du IHM WinCC dans TIA portal .....	24
<b>Figure 3-1</b> : Tableau Electrique.....	28
<b>Figure 3-2</b> : Vue de face avant du tableau basse tension .....	29
<b>Figure 3-3</b> : Vue de face avant de la HMI .....	31
<b>Figure 3-4</b> : Vue de générale des groupes électropompes .....	32
<b>Figure 3-5</b> : Vanne motorisée .....	33
<b>Figure 3-6</b> : Le courant d'un moteur asynchrone triphasé réduit lors du démarrage par un démarreur progressif SIRIUS .....	34
<b>Figure 3-7</b> : Le démarreur progressif SIRIUS 3RW4024.....	34
<b>Figure 3-8</b> : Variateur de vitesse SINAMICS V20 .....	35
<b>Figure 3-9</b> : Système anti-bélier .....	35
<b>Figure 3-10</b> : Capteur de niveau .....	36

<b>Figure 3-11</b> : Capteur de température .....	36
<b>Figure 3-12</b> : Capteur de vibration .....	37
<b>Figure 3-13</b> : Débitmètre.....	37
<b>Figure 4-1</b> : Table variable du Programme « anti bélier ».....	44
<b>Figure 4-2</b> : Table variable du Programme « Gestion de la vanne ».....	47
<b>Figure 4-3</b> : Table variable du Programme « Autorisation ».....	53
<b>Figure 4-4</b> : Table variable du Programme « Défaut » .....	59
<b>Figure 4-5</b> : Table variable du Programme « GEP».....	65

# Liste des tableaux

<b>Tableau 3-1</b> : Caractéristiques techniques de la CPU 1214C DC/DC/Relay .....	10
<b>Tableau 2-1</b> : Types de données Bool, Byte, Word et DWord.....	15
<b>Tableau 2-2</b> : Types de données entiers.....	15
<b>Tableau 2-3</b> : Types de données "date et heure" .....	16
<b>Tableau 2-4</b> : Types de données réels à virgule flottante .....	16

# Liste des abréviations

<b>AC</b>	alternative current (courant alternatif)
<b>API</b>	Automate Programmable industriel
<b>DC</b>	direct current (courant continu)
<b>CPU</b>	Central Processing Unit
<b>EEPROM</b>	Electrically Erasable Programmable Read Only Memory
<b>GEP</b>	groupe électropompe
<b>IHM</b>	Interface Homme/Machine
<b>LD</b>	langage à contacts LADDER
<b>RAM</b>	Random Access Memory
<b>ROM</b>	Read Only Memory
<b>SCL</b>	langage structure
<b>TOR</b>	tout ou rien
<b>PLCSIM</b>	Le logiciel de simulation d'automate siemens
<b>SIMATIC</b>	est une série de contrôleurs logiques programmables et de systèmes d'automatisation, développés par Siemens
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol
<b>LED</b>	Light-Emitting Diode
<b>E/S</b>	Entrées/ Sorties
<b>TIA PORTAL</b>	Totally Integrated Automation Portal

# Introduction générale

Les progrès réalisés dans l'électronique et la baisse des coûts des composants électroniques ont amené les responsables des entreprises à recourir à l'automatisation.

Aujourd'hui, l'automate programmable industriel « API » est le composant d'automatisation le plus largement utilisé. On les retrouve dans presque la plupart des secteurs de l'industrie car ils répondent aux besoins d'adaptation et de flexibilité d'un grand nombre de processus.

Le domaine de l'hydraulique est également témoin, entre autres, de cette révolution de l'automatisation. Des entreprises algériennes (ANBT, SEEAL, Hydro-Aménagement, ...etc.) spécialisées dans le stockage, la purification et la distribution de l'eau cherchent à mettre en œuvre cette solution d'automatisation dans les stations de pompage et de traitement. Siemens est une entreprise compétitive qui fournit ce type de service industriel.

Dans le but d'amélioration des applications (programmes) de gestion des stations de pompage on s'est intéressé à la commande et la programmation d'un par un automate programmable siemens (SIMATIC S7-1200) à l'aide d'un logiciel de programmation TIA PORTAL V16.

Notre travail est divisé en quatre chapitres :

Le premier chapitre est consacré aux généralités sur les automates programmables d'une manière générale, puis d'une façon détaillée sur l'automate S7-1200 de siemens.

Dans le deuxième chapitre nous présentons logiciel TIA portal et la description du logiciel, de supervision et une description générale de langage LADDER qui vont être utilisées à la réalisation de notre projet à dernière chapitre.

Dans le troisième chapitre, nous avons présenté une description générale de la station de pompage et des équipements d'automatisation qui y sont utilisés.

Le quatrième chapitre présente la programmation et la supervision de différentes parties de la station par le logiciel TIA Portal, selon le cahier des charges détaillé.

Enfin, notre travail est terminé par une conclusion générale

# **Chapitre 1:**

# **Les Automates Programmables**

# **Industriels**

## 1.1 Introduction

Le développement rapide des industries conduit à une augmentation continue du besoin d'automatisation, c'est pourquoi l'API a remplacé les armoires de relais en raison de sa flexibilité de mise en œuvre, Ce chapitre sera consacré à la présentation des automates programmables de Siemens.

## 1.2 Historique

Les automates programmables industriels sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande. Les coûts de l'électronique permettant alors de remplacer avantageusement les technologies actuelles[1].

## 1.3 Définition de l'API

Selon la norme française EN 61131-1, un automate programmable est un système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en œuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties Tout ou Rien ou analogiques divers types de machines ou de processus.

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

- Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées/sortie industrielles.
- Il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, microcoupures de la tension d'alimentation, parasites, etc.).
- Enfin, sa programmation à partir des langages spécialement développés pour le traitement des fonctions d'automatismes facilitent son exploitation et sa mise en œuvre [2].



## 1.4 Architecture des automates programmables industriels

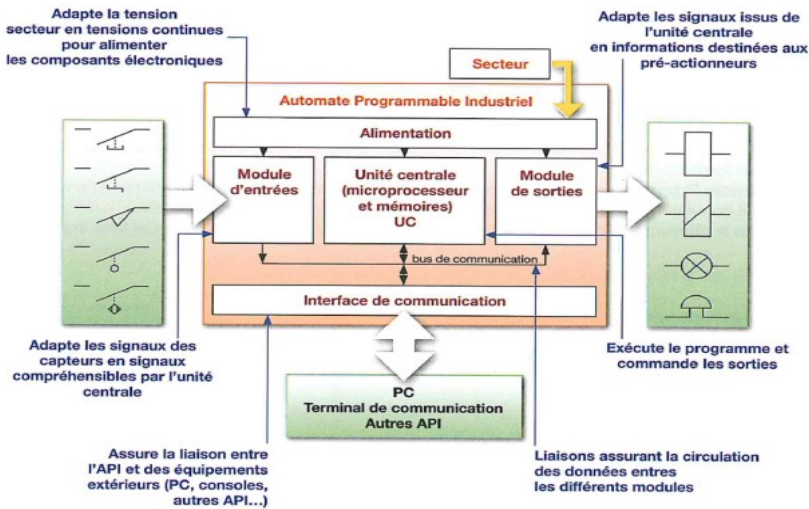


Figure 1-1 : Architecture d'un API [3]

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante :

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM et EEPROM nécessaire pour stocker les programmes, les données, et les paramètres de configuration du système.
- Un module d'alimentation qui, à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues +/- 5V, +/-12V ou +/-15V.
- Un ou plusieurs modules d'entrées 'Tout ou Rien' ou analogiques pour l'acquisition des informations provenant de la partie opérative (procédé à conduire).
- Un ou plusieurs modules de sorties 'Tout ou Rien' (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties [4].

## 1.5 Structure interne des automates programmables

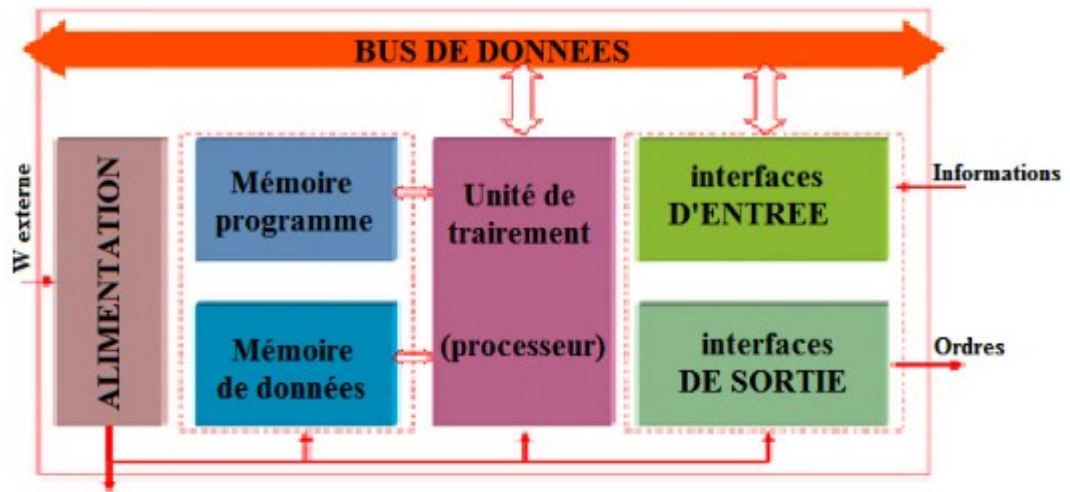


Figure 1-2 : Structure interne d'un automate programmable [5]

Les API comportent six parties principales :

- Une unité de traitement ou processeur.
- Une mémoire programme.
- Une mémoire de données.
- Une interface d'entrée.
- Une interface de sortie.
- Un module d'alimentation.

Ces six parties sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces six secteurs de l'API). Ces six parties réunies forment un ensemble compact appelé automate [5].

### 1.5.1 Une unité de traitement ou processeur

Le processeur gère l'ensemble des échanges informationnels en assurant :

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

### 1.5.2 Une mémoire programme

La mémoire programme de type RAM contient les instructions à exécuter par le processeur afin de déterminer les ordres à envoyer aux pré actionneurs reliés à l'interface de sortie en fonction des informations recueillies par les capteurs reliés à l'interface d'entrée.

### 1.5.3 Une mémoire de données

La mémoire de donnée permet le stockage de :

- L'image de l'entrée reliée à l'interface d'entrée.
- L'état des sorties élaborées utilisées par le processeur.
- Les valeurs internes utilisées par le programme (résultats de calculs, états intermédiaires).
- Les états forcés ou non des E/S.

### 1.5.4 Une interface d'entrée

L'interface d'entrée permet la connexion à l'API d'une multitude de capteurs pouvant être:

- TOR (logiques ou Tout Ou Rien),
- Numériques,
- Analogiques.

Ces différentes entrées sont mises en forme par l'interface d'entrée avant d'être stockées dans la mémoire de données.

### 1.5.5 Une interface de sortie

L'interface de sortie permet la connexion à l'API d'une multitude de préactionneurs pouvant être :

- TOR (logiques ou Tout Ou Rien).
- Numériques.
- Analogiques.

### 1.5.6 Un module d'alimentation

Le module d'alimentation transforme l'énergie externe provenant du réseau en la mettant en forme afin de fournir aux différents modules de l'API les niveaux de tension nécessaires à leur bon fonctionnement.

Plusieurs niveaux de tension peuvent être utilisés par les circuits internes (3v, 5v, 12v, 24v...). Il sera dimensionné en fonction des consommations des différentes parties.

## 1.6 Les automates Siemens



Figure 1-3 : Automate Siemens

Si nous devrions citer tous les modèles des automates Siemens, nous passerions plusieurs heures à le faire tellement ils sont nombreux. Cependant, on peut catégoriser les automates Siemens par gamme. Le portefeuille des automates Siemens est libellé sous le nom SIMATIC. Cela englobe non seulement les petits automates servant à réaliser des tâches logiques simples aux automates destinés aux systèmes plus complexes. On peut ainsi classer les automates Siemens suivant les gammes Logo qui sont plutôt des modules logiques, les gammes S7-200 qui se programment avec le logiciel microWin, les gammes S7-1200 qui sont des automates très compacts et qui seront les futurs successeurs des S7-200, les gammes S7-300 pour des applications de grande taille, les gammes S7-400 pour dans la plupart du temps commander les industries de process à haut taux de disponibilité, les gammes S7-1500 qui sont les dernières générations d'automates de la marque Siemens. Outre ces gammes d'automates, Siemens dispose d'autres contrôleurs comme les PC industriels (gamme Microbox) ou les automates logiciels comme WinAC [6].

## 1.7 Introduction à l'automate S7-1200

L'automate SIMATIC S7-1200 est un mini-contrôleur modulaire utilisé pour les petites performances. Il existe un éventail complet de modules pour une adaptation optimisée à la tâche d'automatisation.

- Le contrôleur S7 est composé d'une CPU qui est équipée d'entrées et de sorties de signaux numériques et analogiques.

- Des modules additionnels d'entrées/sorties (modules IO) peuvent être installés si les entrées et sorties intégrées ne sont pas suffisantes pour l'application désirée.
- Si besoin est, des modules de communication RS232 ou RS485 sont ajoutés.
- Une interface TCP/IP intégrée est obligatoire pour toutes les CPU.
- Avec le programme S7, l'API surveille et contrôle une machine ou un process. Les modules IO sont interrogés dans le programme S7 au moyen d'adresses d'entrées (%I) et référencés au moyen d'adresses de sorties (%Q).
- Le système est programmé avec le logiciel STEP 7 Basic VX.X (TIA Portal VX.X) [7].

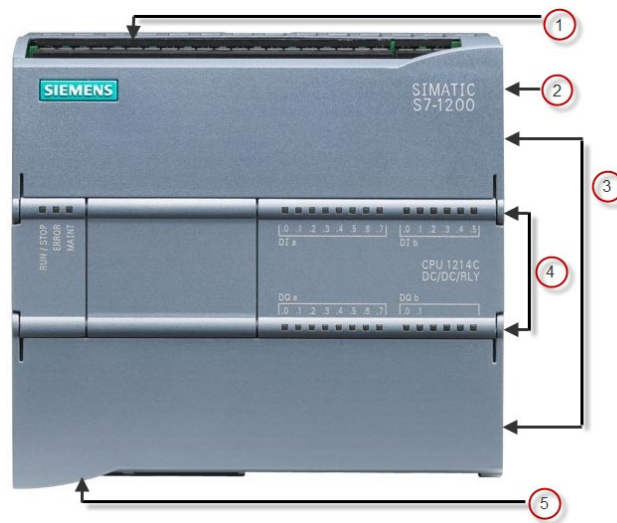


Figure 1-4 : Automate programmable S7-1200 [8]

- 1) Prise d'alimentation
- 2) Logement pour carte mémoire sous le volet supérieur
- 3) Connecteurs amovibles pour le câblage utilisateur (derrière les volets)
- 4) DEL d'état pour les E/S intégrées
- 5) Connecteur PROFINET (sur la face inférieure de la CPU) [8].

## 1.8 Présentation de la CPU 1214C

La CPU 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 contrôleur puissant [9].

### 1.8.1 Caractéristiques techniques de la CPU

**Tableau 1-1** : Caractéristiques techniques de la CPU 1214C DC/DC/Relay [10]

<b>Informations Générales</b>	
Désignation du type de produit	CPU 1214C DC/DC/Relay
Version du firmware	V4.4
<b>Tension d'alimentation</b>	
Valeur nominale (CC)	24V CC
Courant d'entrée	
Consommation (valeur nominale)	500 mA ; uniquement CPU
<b>Courant de sortie</b>	
Pour bus interne (5 V CC), max.	1 600 mA ; max. 5 V CC pour SM et CM
<b>Mémoire</b>	
Mémoire de travail	100 Kbyte
Mémoire de chargement	4 Mbyte
Puissance dissipé	12KW
<b>Configuration matérielle</b>	
Entrées Analogiques	02
Entrées TOR	14 ; intègre
Sorties TOR	10: relais
Sorties analogiques	0
<b>Programmation</b>	
Langage de programmation	CONT, LIST, LOG
<b>Dimensions</b>	
Largeur	110 mm
Hauteur	100 mm
Profondeur	75 mm
<b>Poids</b>	<b>435 g</b>

### 1.8.2 Modes de fonctionnement de la CPU

La CPU a les modes de fonctionnement suivants :

- En mode « **STOP** », la CPU n'exécute pas le programme, et vous pouvez
- Charger un projet.
- En mode « **STARTUP** », la CPU entame une procédure de démarrage.
- En mode « **RUN** », le programme est exécuté de façon cyclique. Les projets ne peuvent pas être chargés dans une CPU en mode **RUN**.

La CPU n'a pas de commutateur physique pour changer de mode de fonctionnement. Le mode **STOP** ou **RUN** se change en utilisant le bouton sur le panneau de commande du logiciel

Tia Portal v16. De plus, le panneau de commande est muni d'un bouton **MRES** pour faire une réinitialisation générale de la mémoire et il affiche l'état actuel des LED de la CPU [11].



Figure 1-5 : Fenêtre du contrôleur SIMATIC PLCSIM

La couleur de la LED des états **RUN/STOP** sur la face avant de la CPU indique le mode de fonctionnement actuel.

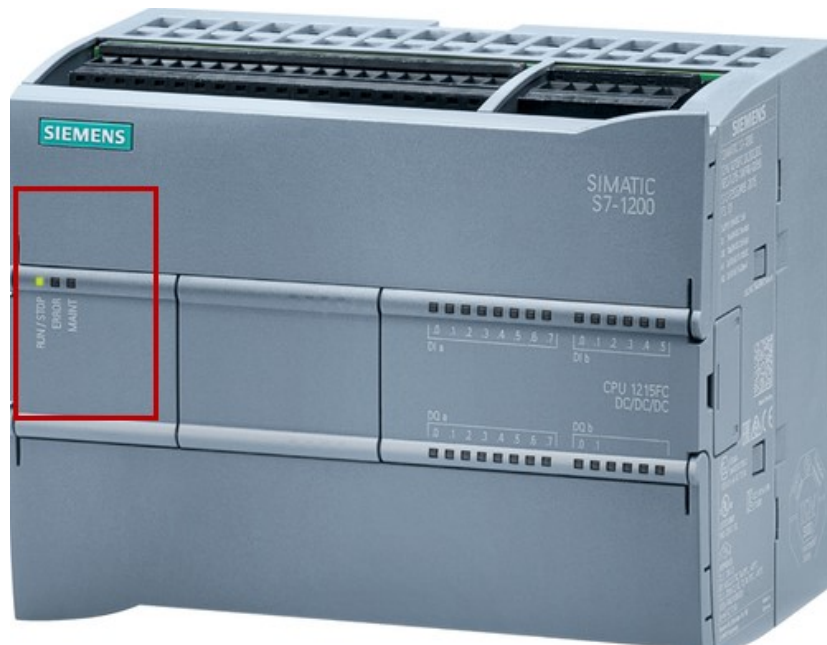


Figure 1-6 : Modes de fonctionnement de la CPU [11]

- Une lumière **JAUNE** indique le mode **STOP**.
- Une lumière **VERTE** indique le mode **RUN**.
- Une lumière **CLIGNOTANTE** indique le mode **STARTUP**.

En outre il y a les LED « **ERROR** » et « **MAINT** » qui indiquent respectivement si une erreur est survenue et si une maintenance est requise [11].

## 1.9 Conclusion

Dans ce chapitre nous avons présenté les API avec sa structure interne. L'étude des automates nous aide à mieux classer les produits d'automatisation qu'on aura à manipuler, et aussi à bien choisir l'automate qui convient à notre application. C'est le Siemens S7-1200.



# **Chapitre 2:**

# **Les Logiciels Programmables**

## 2.1 Introduction

La plateforme de développement TIA Portal de Siemens permet de faire un gain important en temps lors du développement de systèmes d'automatisation. C'est une plateforme tout en un comportant le logiciel STEP 7 pour la programmation d'automates et WinCC pour les interfaces homme-machine IHM [12].

Dans ce chapitre on va donner un aperçu général sur TIA Portal ainsi que le langage de programmation LADDER.

## 2.2 Description du logiciel TIA Portal

La plate-forme "Totally Integrated Automation Portal" est le nouvel environnement de travail de Siemens pour mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant SIMATIC STEP 7 et SIMATIC WinCC. Elle permet de gagner beaucoup de temps dans le développement de systèmes d'automatisation. C'est une plate-forme intégrée qui nous permet de peut simuler intuitivement notre projet avant de le publier [13, 14].



### 2.2.1 Les avantages du logiciel TIA Portal

- Programmation intuitive et rapide : avec des éditeurs de programmation nouvellement développés SCL, CONT, LOG, LIST et GRAPH.
- Efficacité accrue grâce aux innovations linguistiques de STEP 7 : programmation symbolique uniforme, Calculate Box, ajout de blocs durant le fonctionnement, et bien plus encore.
- Performance augmentée grâce à des fonctions intégrées : simulation avec PLCSIM, télémaintenance avec TeleService et diagnostic système cohérent.
- Technologie flexible : Fonctionnalité motion control évolutive et efficace pour les automates S7-1500 et S7-1200.
- Sécurité accrue avec Security Integrated : Protection du savoir-faire, protection contre la copie, protection d'accès et protection contre la falsification.
- Environnement de configuration commun avec pupitres IHM et entraînements dans l'environnement d'ingénierie TIA Portal [13, 14].

## 2.3 SIMATIC STEP 7

SIMATIC STEP 7, intégré à TIA Portal v16, est le logiciel de configuration, programmation, vérification et diagnostic de tous les automates SIMATIC. Doté d'un grand nombre de fonctions conviviales, SIMATIC STEP 7 garantit une efficacité nettement supérieure pour toutes les tâches d'automatisation, qu'il s'agisse de la programmation, de la simulation, de la mise en service ou de la maintenance [15].

### 2.3.1 Utilisation de STEP 7

#### 2.3.1.1 Types de données utilisées en STEP 7

Il existe plusieurs types de données distinctes dans le STEP 7, ces variables sont déclarées préalablement avant chaque début de programme. Le tableau suivant réunit ces différentes variables :

**a) Types de données pour bit et séquence de bits :**

**Tableau 2-1** : Types de données Bool, Byte, Word et DWord [10]

Type de données	Taille en bits	Type de nombre
BOOL	1	Booléen
		Binaire
		Octal
		Hexadécimal
Byte	8	Binaire
		Entier non signé
		Octal
		Hexadécimal
Word	16	Binaire
		Entier non signé
		Octal
		Hexadécimal
DWord	32	Binaire
		Entier non signé
		Octal
		Hexadécimal

**b) Types de données entiers (U = non signé, S = court, D= double) :**

**Tableau 2-2** : Types de données entiers [10]

Type de données	Taille en bits	Plage de nombres
USInt	8	0 à 255
Sint	8	-128 à 127

UInt	16	0 à 65 535
Int	16	-32 768 à 32 767
UDInt	32	0 à 4 294 967 295
DInt	32	-2 147 483 648 à 2 147 483 647

c) Types de données "date et heure" :

Tableau 2-3 : Types de données "date et heure" [10]

Type de données	Taille en bits	Plage
TIME	32 bits	T#-24d_20h_31m_23s_648ms à T#24d_20h_31m_23s_647ms Sauvegardé en tant que : -2 147 483 648 ms à +2 147 483 647 ms
DATE	16 bits	D#1990-1-1 à D#2168-12-31
TIME_OF_DAY	32 bits	TOD#0:0:0.0 à TOD#23:59:59.999
DTL (date et heure long)	12 octets	Min. : DTL#1970-01-01-00:00:00.0 Max. : DTL#2554-12-31- 23:59:59.999 999 999

d) Types de données réels à virgule flottante (L=long) :

Tableau 2-4 : Types de données réels à virgule flottante [10]

Type de données	Taille en bits	Plage de nombres
Real	32	-3,402823e+38 à -1,175 495e-38, ±0, +1,175 495e-38 à +3,402823e+38
LReal	64	-1,7976931348623158e+308 à -2,2250738585072014e-308, ±0, +2,2250738585072014e-308 à +1,7976931348623158e+308

### 2.3.2 Vue du portail et vue du projet

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

1. **La vue du portail** : est une vue orientée sur les tâches du projet.
2. **La vue du projet** : est une vue des composants du projet et des zones de travail et éditeurs correspondants [13, 14].

#### 2.3.2.1 La vue du portail

La vue du portail offre une vue orientée sur les tâches des outils. Vous pouvez y décider rapidement ce que vous souhaitez faire et appeler l'outil requis pour la tâche correspondante. Si nécessaire, un basculement automatique dans la vue du projet a lieu pour la tâche sélectionnée.

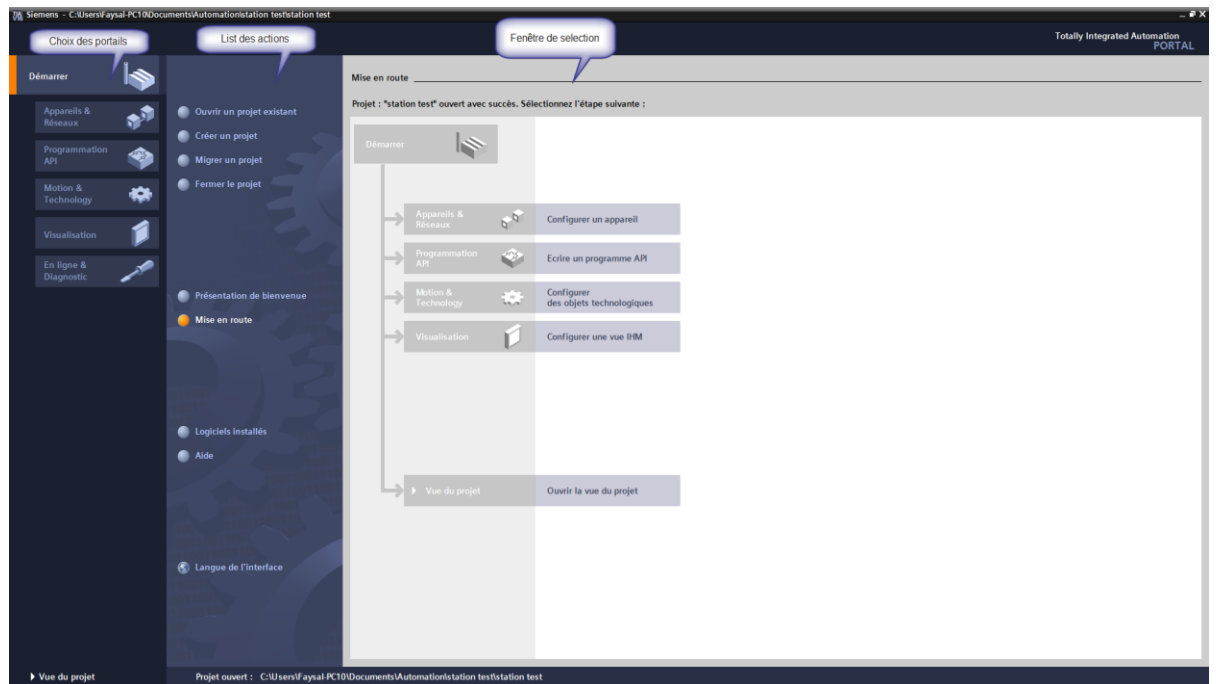


Figure 2-1 : Vue du portail

### 2.3.2.2 Vue du projet

La vue du projet correspond à une vue structurée de l'ensemble des composants du projet.

**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 à bibliothèques des composants, bloc de programme à 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.

### 2.3.3 Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matérielle, il faut aller dans « *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 [15].

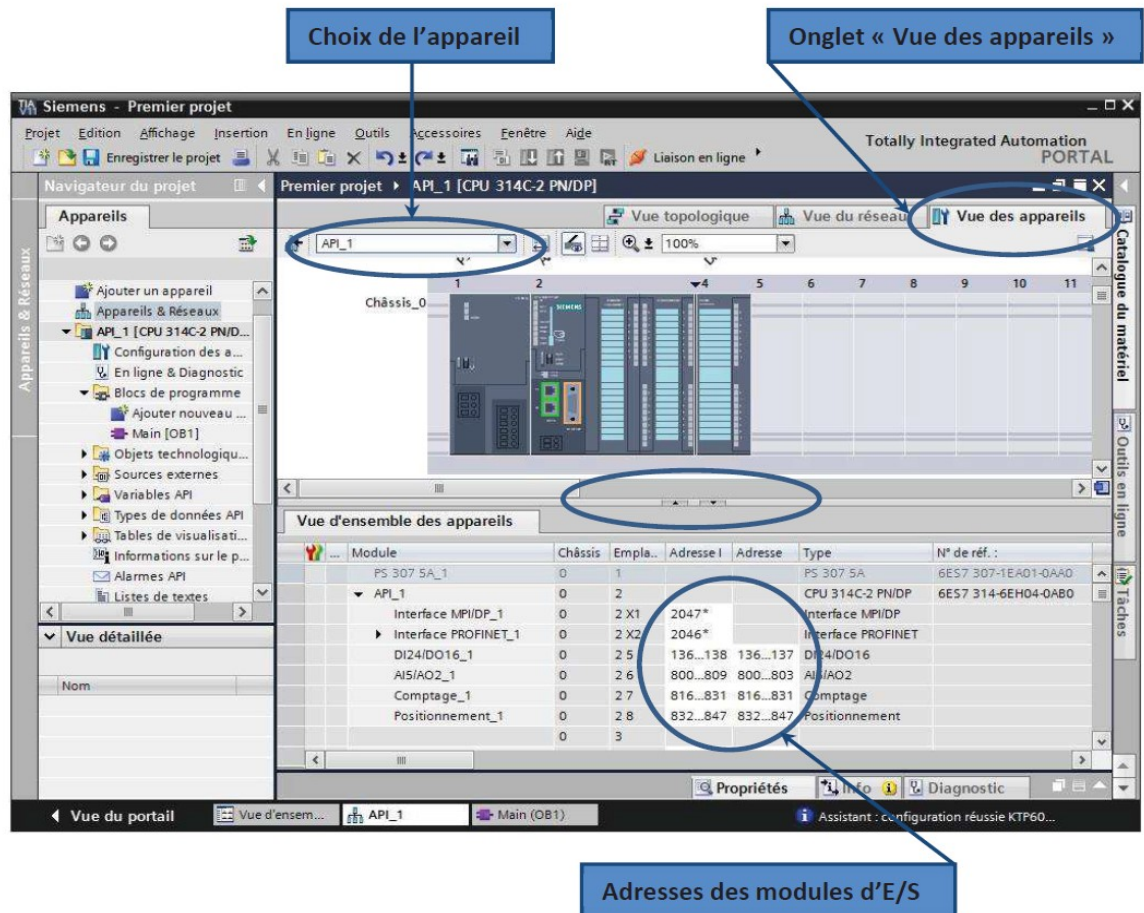


Figure 2-2 : Adressage des E/S

### 2.3.4 Mnémoniques

Une mnémonique (Nom symbolique) nous permet d'utiliser des désignations parlantes à la place d'adresses absolues. En combinant l'usage de mnémoniques courtes et de commentaires explicites, on répond à la fois aux besoins d'une programmation concise et d'une programmation bien documentée.

Les mnémoniques améliorent considérablement la lisibilité et la clarté d'un programme [2].

### 2.3.5 Mémentos

Les mémentos sont des éléments électroniques bistables qui servent à mémoriser les états logiques.

Les mémentos sont utilisés pour les opérations internes de l'automate pour lesquelles l'émission d'un signal n'est pas nécessaire [2].

### 2.3.6 Types de blocs de programme

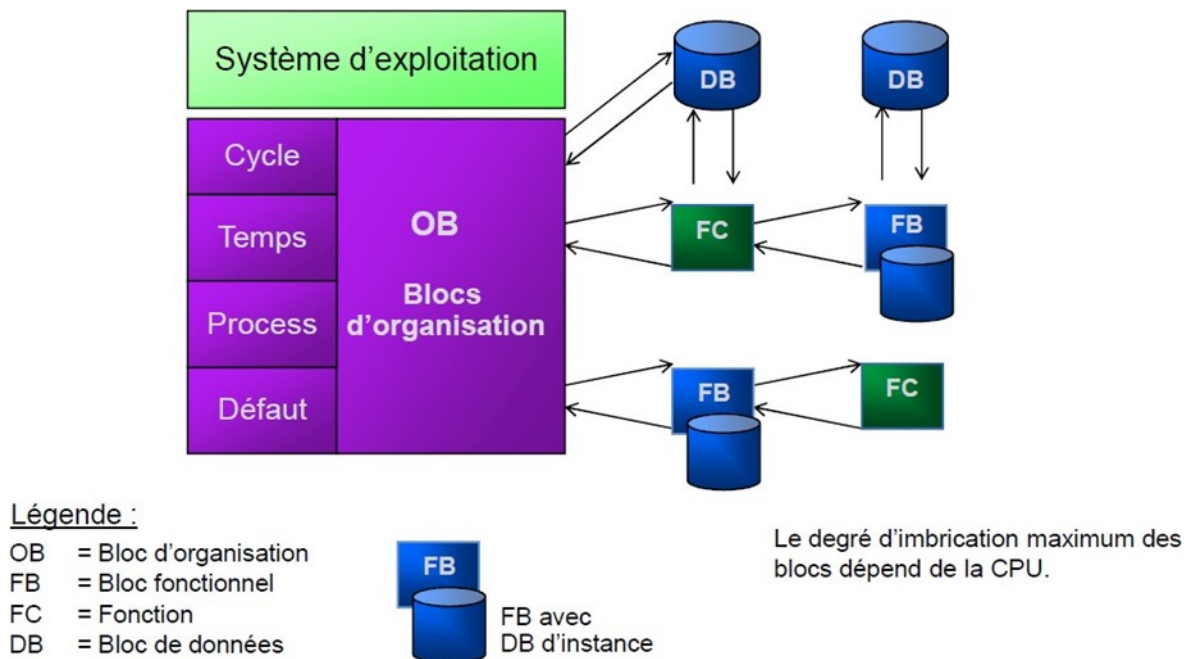


Figure 2-3 : Types de blocs de programme [16]

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, FC et DB.

#### 2.3.6.1 Bloc d'Organisation (OB)



Ils constituent l'interface entre le système d'exploitation et le programme utilisateur.

Les OB sont programmables par l'utilisateur, ce qui permet de déterminer le comportement de la CPU.

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

- Comportement au démarrage.
- Exécution cyclique du programme.
- Exécution du programme déclenchée par des alarmes (cyclique, processus, diagnostic, ...).
- Traitement des erreurs.

Pour que le traitement du programme démarre, le projet doit posséder au moins un OB cyclique.[16]

### 2.3.6.2 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 opérands globaux.

Elles sont utilisées pour la programmation de fonctions utilisées plusieurs fois. On simplifie de ce fait la programmation [16].

### 2.3.6.3 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 [16].

### 2.3.6.4 Bloc de données (DB)



Les DB sont employés afin de tenir à disposition de l'espace mémoire pour les variables de données. Il y a deux catégories de blocs de données. Les DB globaux où tous les OB, FB et FC peuvent lire des données enregistrées et écrire eux-mêmes des données dans le DB. Les instances DB sont attribuées à un FB défini [16].

La figure ci-dessous montre les blocs de configuration.



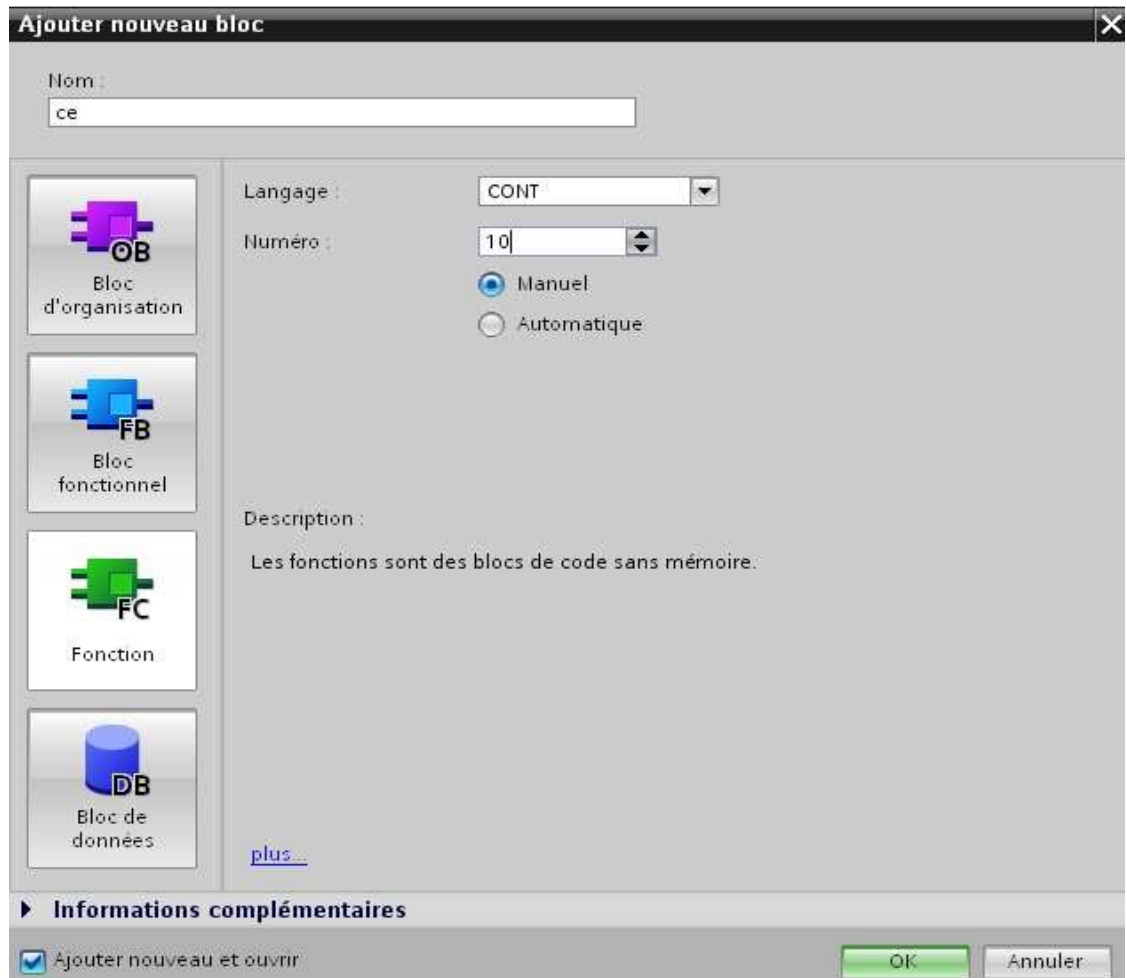


Figure 2-4 : Bloc de configuration

### 2.3.7 Les langages des programmations

Chaque automate possède son propre langage. Les constructeurs proposent tous une interface logicielle répondant à la norme CEI 11313. Cette norme définit cinq langages de programmation utilisables, qui sont [17] :

- LD (Ladder Diagram)
- IL (Instruction List)
- FBD (Function Block Diagram)
- ST (Structured Text)
- SFC (Sequential Function Charts)

Le tableau suivant présente quelques fonctions par trois langages :

Désignation	Langage LD	Langage IL	Langage FBD
ET Logique		U	
ET NON		UN	
OU Logique		O	
OU NON		ON	
Affectation résultat		=	
ET d'une expression		U(	
OU d'une expression		O(	
Exécute l'instruction différée		)	
Definer repère (Etiquette)		: <repère>	
Saut		SPA	

Figure 2-5 : Tableau présente trois Langages de programmation des API [18]

### 2.3.7.1 Le langage Ladder Diagram (LD)

Le langage Ladder Diagram peut être considéré comme une représentation graphique d'équations booléennes combinant des contacts (arguments d'entrée) avec des bobines (résultats de sortie). Le langage LD permet la description de tests et de modifications de données booléennes à l'aide de symboles graphiques placés dans un diagramme. Les symboles graphiques LD sont organisés comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont connectés à gauche et à droite à des barres d'alimentation verticales [19].

## 2.4 La supervision

### 2.4.1 Introduction à SIMATIC HMI

Lorsque la complexité des processus augmente et que les machines et les installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a

besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme Machine (IHM).

Un système IHM constitue l'interface entre l'homme (opérateur) et le processus (machine/installation). Le contrôle proprement dit du processus est assuré par le système d'automatisation. Il existe par conséquent une interface entre l'opérateur et WinCC flexible (sur le pupitre opérateur) et une interface entre WinCC flexible et le système d'automatisation [20].

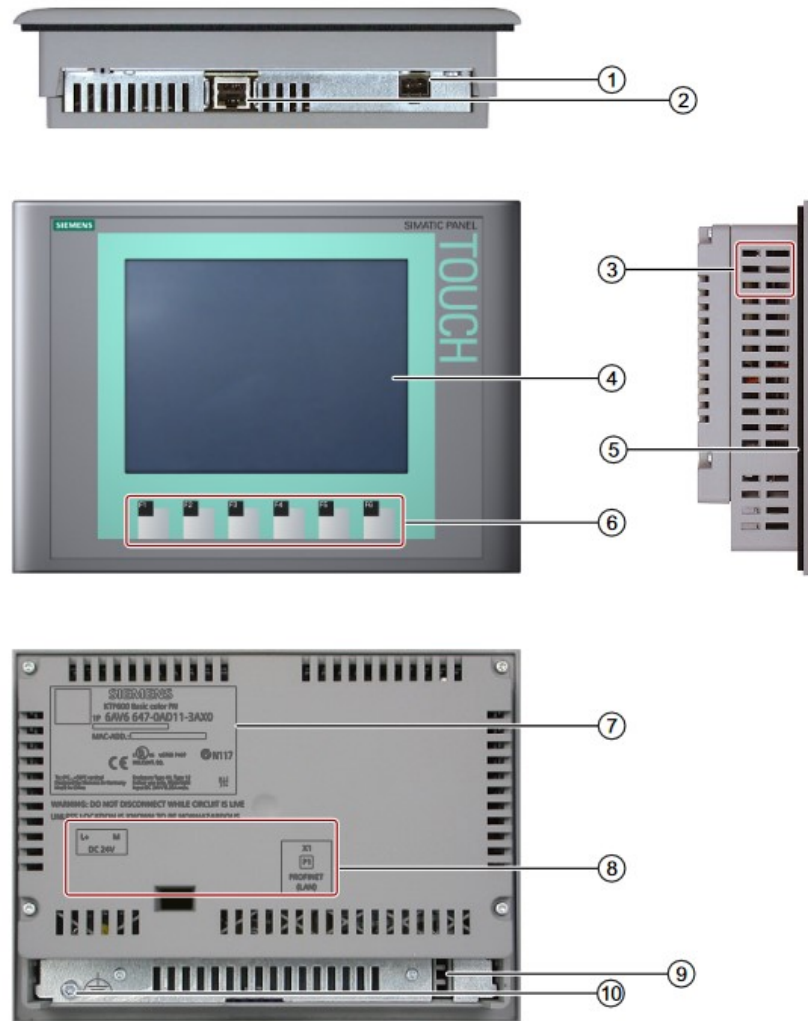


Figure 2-6 : Interface Homme Machine [21]

- |                                     |                                     |
|-------------------------------------|-------------------------------------|
| 1. Connecteur d'alimentation        | 6. Touches de fonction              |
| 2. Interface PROFINET               | 7. Plaque signalétique              |
| 3. Encoches pour griffe de fixation | 8. Désignation de l'interface       |
| 4. Afficheur/écran tactile          | 9. Glissière pour bande de repérage |
| 5. Joint de montage                 | 10. Prise de terre fonctionnelle    |

## 2.4.2 SIMATIC WinCC (TIA Portal)

Famille de systèmes de configuration avec WinCC Basic, Comfort, Advanced et Professional pour les pupitres opérateur SIMATIC, ainsi que pour les systèmes de supervision WinCC Runtime Advanced et WinCC Runtime Professional fonctionnant sur PC.

WinCC (TIA Portal) est basé sur le nouveau cadre d'ingénierie central Totally Integrated Automation Portal (TIA Portal), qui offre à l'utilisateur une solution uniforme, efficace et intuitive à toutes les tâches d'automatisation.

WinCC (TIA Portal) offre également une ingénierie uniforme, du Panneau de base jusqu'aux applications SCADA.

Avec les produits STEP 7 (TIA Portal), WinCC (TIA Portal) constitue la solution optimale pour une ingénierie intégrée et efficace [12].

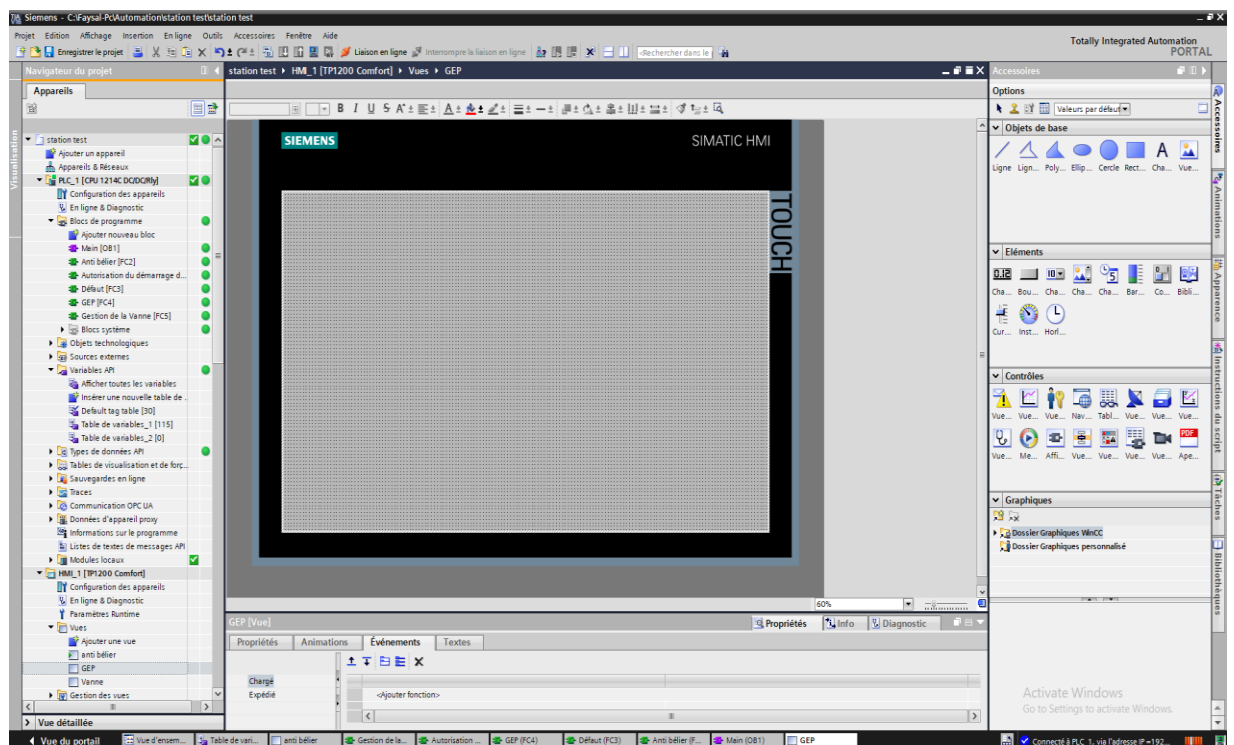


Figure 2-7 : Vue du IHM WinCC dans TIA portal

## 2.5 Conclusion

Dans ce chapitre on a donné une vue générale sur le logiciel Siemens TIA Portal coté programmation et supervision. La gamme des logiciels TIA Portal offre une facilité d'utilisation et de compréhension, Intégrant la totalité des langages et les outils nécessaires pour une programmation complète et souple.

# **Chapitre 3:**

## **Description De La Station De Pompage**

### 3.1 Introduction

En général, on essaie de faire véhiculer les eaux gravitaires ment, 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 et de faire l'automatisation d'une telle station. Dans ce chapitre on va donner une description générale de la station concernée.

### 3.2 Notre objectif

Le but c'est d'automatisé un système de pompage d'eau potable de la Banlieue Est d'Alger qui se situé à Kouba (poste 107). La station est conçue pour un réservoir de stockage d'aspiration de pompe intermédiaire. La pompe fonctionne à 1300 m /h respectivement 2600 m /h avec une pression de refoulement de 58 m. Sur le collecteur de refoulement principal des pompes, un réservoir anti-bélier protège les pompes en cas d'une éventuel vague de pression.

Les vannes de décharge des pompes sont motorisées, et les vannes d'aspiration sont manuelles. Les moteurs des pompes sont munis de démarreur progressif et de variateur de vitesse. Un débitmètre, installé sur la ligne d'aspiration qui indique et enregistre le débit en temps réel quand une ou plusieurs pompes sont en service.

De la ligne d'alimentation de Garidi, il y a deux lignes (l'une principale et l'autre de secours) qui alimentent la station de pompage en eau. Une ligne DN 1000 séparée par une vanne alimente le réservoir (R100) et l'autre ligne DN 700 séparée avec une vanne qui peut alimenter directement le collecteur d'aspiration de la pompe (système By Pass).

Le niveau du réservoir R100 intermédiaire est contrôlé par un indicateur de niveau à ultrasons LC 01 et des capteurs de l'état du niveau (niveau Bas : LLL, niveau Moyen : LLA, niveau Haut : HL). Sur l'écran de la salle de contrôle, l'opérateur voit le niveau du réservoir en temps réel qui est sera schématisé par une jauge du niveau du réservoir R100, indiquant le contenu de 0 à 100%. De surcroît, un flotteur arrête les pompes au niveau très bas (indiqué par le capteur à ultrason) et informe l'opérateur par une alarme sonore et visuel sur l'écran de supervision quand le niveau haut de (HL) est aussi atteint.

Le débit d'eau vers le réservoir est contrôlé par une vanne altimétrique. Un pilote de niveau haut et un pilote de niveau bas sont ajustés pour une ouverture réglable. Le remplissage du

réservoir est effectué à partir du fond. Le réservoir est équipé d'un trop-plein et d'une vanne de drainage. La sortie du réservoir DN 1000 alimente le collecteur d'aspiration de la pompe.

En cas d'une panne du réseau électrique dans la station de pompage, le groupe électrogène diesel se met en marche pour assurer le fonctionnement d'une pompe afin de garantir la continuité de service en ressources en eau.

***Donc l'objectif principal de cette réalisation est :***

- Apporter une optimisation de la station de pompage, qui est absorbé en fonction des besoins en temps réel pour un fonctionnement des pompes qui est affiché sur le pupitre de commande.
- Le fonctionnement de la station prévoit un fonctionnement en mode dégradé en cas de panne sur les éléments d'automatisme ou d'une éventuel pompe (à remplacer ou en cas de maintenance) pour assurer une continuité de service.

### **3.3 Les équipements utilisés**

#### **3.3.1 Equipement MT (Moyen Tension)**

##### **3.3.1.1 Le câble MT**

Le câble MT fait la liaison entre la cellule (déjà existante sur le site) et le transformateur. Sa fonction proprement dit est le transport d'énergie électrique d'un endroit vers un autre. Ce câble sera placé dans la cellule existante jusqu'au transformateur qui lui sera à proximité (c'est-à-dire dans le même local, très peu distant l'un de l'autre).

##### **3.3.1.2 La Cellule de protection Auxiliaire Transformateur**

La protection des transformateurs de puissance permettra de préserver notre installation MT, contre tout risque électrique du type (court-circuit, surintensité, etc.). Ces cellules seront placées à proximité du transformateur au côté des cellules existantes. Le nombre de cellules correspond au nombre de source, c'est-à-dire qu'il y deux arrivées (SN1 et SN2).

##### **3.3.1.3 Le transformateur**

Le transformateur a pour rôle essentiel d'abaisser ou l'augmenter la tension, ici dans notre installation le transformateur sera d'abaisser la tension donc généralement on dit que c'est un transformateur abaisseur. Il aura du côté primaire (entré du transfo) une tension de 10KV et du coté secondaire (en sorti du transfo) il aura comme valeur une tension de 400V.

Il sera placé au plus proche des cellules SM6 QM et sera équipé de relais Buchholz et de thermomètre à 2 seuils.

Nombre de transformateur : 2 transformateurs (il y aura un transformateur par arrivé réseau, d'où l'installation dispose de 2 arrivées SN1 et SN2 donc 2 transformateurs pour assurer l'alimentation en énergie électrique du site).

### **3.3.2 Equipement Electrique BT (Basse tension)**

#### **3.3.2.1 Tableau Electrique**

Le tableau électrique recouvre les divers organes de commande et de puissance. Elle sera placée dans le local électrique de la station. Il y aura placé sur ses armoires des étiquettes afin de pouvoir les identifiées (arrivées, départ groupe électro-pompe 1, 2 et 3, réserve, auxiliaires, compensation). 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 3-1** : Tableau Electrique





Figure 3-2 : Vue de face avant du tableau basse tension

### 3.3.2.2 Arrivée générale

- Un inverseur de source avec verrouillage électrique et mécanique, il sera équipé de:
  - Deux (02) disjoncteurs de protection Masterpact 1000A 4Pole chacun, ils seront débroschables sur châssis avec des contacts auxiliaires OF, des contacts signal défauts SD, des bobines d'ouverture MX et des bobines de fermeture électrique, ils seront équipés des protections différentielles.
  - Un (01) disjoncteur de protection Masterpact 1000A 4Pole, il sera débroschable sur châssis avec un contact auxiliaire OF, un contacts signal défaut SD, une bobine d'ouverture MX et une bobine de fermeture électrique, il sera équipé d'une protection différentielle.
- Des jeux de barre verticaux et horizontaux supportant 2500A, 4 Pôles, ils seront montés sur des isolateurs. Dimension : 80 x 20 mm
- Trois (03) Centrales de mesure PM710:
  - Sous comptage/allocation des coûts.
  - Surveillance à distance de l'installation avec un port Profibus.
  - Surveillance des harmoniques (THD)
  - Des entrées et des sorties logiques.
  - Trois (03) transformateurs d'intensité.
- Trois (03) relais de phase pour la protection contre l'inversion des phases, déséquilibre des hases, manque phase, les sous-tension et surtension.
- Un jeu de barre de terre.

### 3.3.2.3 Armoire Départ 1 et 2

Elles sont aménagées de la même façon, cette à dire qu'elles contiennent la partie Puissance ainsi que les Démarreurs progressif (SIRIUS). Les départs groupes électropompes 315 kW chacun avec les démarreurs progressifs électroniques SIRIUS 3RW40, les inductances de ligne et les contacteurs de by-pass pour limiter les dégagements calorifiques.

### 3.3.2.4 Armoire Départ 3

Cette armoire se compose de la partie Puissance pour le départ 3, où il figure le départ groupe électropompe 355 kW avec le variateur de vitesse SINAMICS V20.

Protection et sécurités du variateur et du moteur :

Protection thermique :

- Contre les échauffements excessifs.
- Protection thermique intégrée dans le variateur par calcul permanent du  $I^2t$ .
- Protection par sonde PTC.

Protection contre :

- Les courts-circuits entre les phases motrices.
- Les coupures des phases motrices.
- Les surintensités entre les phases de sortie.
- Les surtensions.
- Les sous-tensions.
- Rotor du moteur bloqué.
- Les défauts d'isolement terre.
- Les défauts au niveau des entrées analogique et logique.
- Un dépassement de la vitesse limite.
- Défaut de communication.

Sécurité :

- De surtension et de sous-tension du réseau.
- D'absence de phase réseau, en triphasé.

### 3.3.2.5 Cellule Réserve

Elle sera vide, c'est-à-dire qu'elle est prévue pour une future extension du site.

### 3.3.2.6 Cellule Auxiliaire

Cette armoire se compose de la partie départs auxiliaires, tel que :

- Eléments chauffants et ventilateurs pour les groupes électropompes
- Départs puissance vers les vannes motorisées
- Départs vers les coffrets auxiliaires, etc.

#### Armoires batteries de compensation :

Pour chaque départ groupe électropompe, une armoire batterie de condensateur est proposée, elle sera constituée de condensateur, de contacteurs spécifiques pour commander les condensateurs.

### 3.3.3 Le Pupitre de commande

Sur le pupitre de commande, on y trouve sur l'extérieur des voyants (marche/arrêt/défaut, etc.) et des boutons (Marche/arrêt/arrêt d'urgence, etc.) ainsi que les commutateurs (Mode locale/Mode distant/Modes Manuel dégradé, etc.) ainsi que l'écran de supervision qui affichera l'évolution du Procès (Marche/arrêt/Défauts etc.).



**Figure 3-3** : Vue de face avant de la HMI

A l'intérieur du pupitre de commande, il y aura l'automate ainsi que son alimentation et le bornier on l'on retrouve les Entrées/Sorties et un répartiteur pour une éventuelle connexion à l'automate, ce répartiteur dispose d'un mode de protocole qui est le Profibus avec des

emplacements disponibles pour un futur raccord avec d'autre appareillage communicant (Utilisant le Profibus).

### **3.3.4 Le groupe électropompe**

Notre station de pompage contient 3 groupes électropompes, chaque groupe se compose des éléments nécessaires au pompage de l'eau (pompe, vanne, démarreur progressif, variateur de vitesse).

#### **La pompe**

La pompe utilisée **OMEGA 250-480 A** est une pompe centrifuge se caractérise par :

Débit 1300 m<sup>3</sup>/h, puissance 237,3KW, Rendement 86,6%



**Figure 3-4** : Vue de générale des groupes électropompes

### **3.3.5 La vanne**

La vanne joue un rôle très important dans la station, en effet elle garantit un démarrage et un arrêt sécurisé de la pompe (la pompe démarre et s'arrête toujours avec une vanne fermée : cela

la protège contre le refoulement de l'eau à des vitesses très importantes pouvant détruire toute la station).



Figure 3-5 : Vanne motorisée

### 3.3.6 Le démarreur progressif SIRIUS

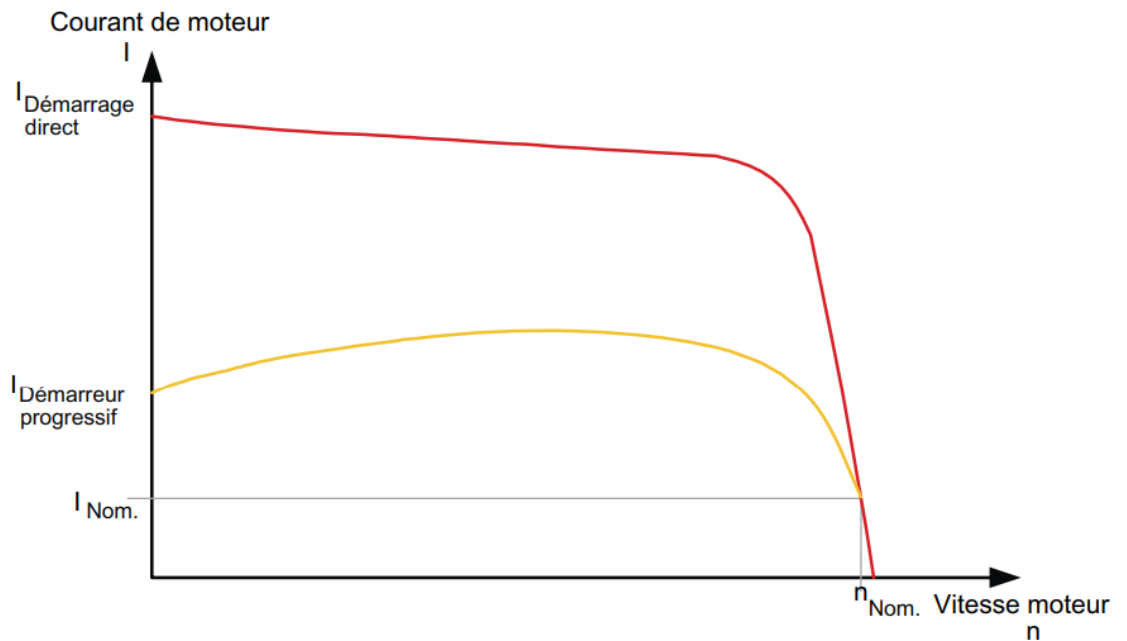
Les démarreurs progressifs SIRIUS 3RW40 comprennent deux thyristors à circuit antiparallèle sur deux de leurs trois phases. Il s'agit pour chacune de ces deux phases d'un thyristor pour la partie d'onde positive et d'un thyristor pour la partie d'onde négative (cf. l'image "Commande par angle de phase et schéma de configuration d'un démarreur progressif à commande biphasée avec contacts de bypass intégrés"). Le courant dans la troisième phase non commandée correspond à la somme des courants des phases commandées.

L'angle de phase permet d'amener la valeur efficace de la tension du moteur en l'espace d'un temps de démarrage sélectionnable depuis une tension de départ définie jusqu'à la tension assignée du moteur.

Le courant du moteur est proportionnel à la tension appliquée au moteur. Le courant de démarrage est ainsi réduit en fonction de la tension appliquée au moteur.

Le couple de rotation évolue selon le carré de la tension appliquée au moteur. Le couple de démarrage est ainsi réduit selon le carré de la tension appliquée au moteur [22].

Les graphiques suivants représentent l'allure du courant et du couple de démarrage d'un moteur asynchrone triphasé en liaison avec un démarreur progressif :



**Figure 3-6 :** Le courant d'un moteur asynchrone triphasé réduit lors du démarrage par un démarreur progressif SIRIUS [22]



**Figure 3-7 :** Le démarreur progressif SIRIUS 3RW4024

### 3.3.7 Variateur de vitesse SINAMICS V20

Le variateur de vitesse triphasé Basic SINAMICS V20 de Siemens avec tension nominale de 380 / 480V AC est adapté aux applications industrielles standard grâce à sa polyvalence, facilité d'utilisation, mise en service courte ainsi que l'efficacité des coûts. Le variateur de fréquence SINAMICS V20 est adapté pour les entraînements avec une puissance nominale jusqu'à 1,5 kW. Les applications potentielles comprennent, par exemple, les pompes, les compresseurs, les ventilateurs, et les convoyeurs [23].



Figure 3-8 : Variateur de vitesse SINAMICS V20

### 3.3.8 Système anti-bélier

ANTI-BELIER : Prévention ou absorption des coups de bélier, élimination des chocs hydrauliques qui se produit lorsque l'écoulement est interrompu plus rapidement que la décélération possible par les pertes de charge dues au frottement du liquide sur les parois. Ceci peut être causé soit par la fermeture trop brusque d'une vanne, soit par l'arrêt d'une pompe. Dans ce dernier cas, un vide se crée derrière la colonne de liquide en mouvement et le choc hydraulique est subi par la pompe elle-même quand l'écoulement s'inverse pour remplir le vide.

Dans notre station on utilise un système anti-bélier : un ballon dont le niveau est indiqué par 5 détecteurs (poires de niveau) contrôlés par :

- Un compresseur qui compresse l'air dans l'anti-bélier afin diminuer le niveau d'eau en cas de surpression d'eau.
- Une vanne qui sert à purger l'eau afin d'augmenter le niveau d'eau dans le ballon en cas de dépression d'eau.



Figure 3-9 : Système anti-bélier

### 3.3.9 Les capteurs

#### 3.3.9.1 Les capteurs de niveau (poire de niveau)

Les capteurs de type TOR sont destinés pour indiquer les niveaux comme suite :

- 3 poires de niveau (niveau Bas : LLL, niveau Moyen : LLA, niveau Haut : HL) dans la bêche (Réservoir source du bas), pour la gestion des pompes et pour éviter la marche a sec de la pompe.
- 5 poires de niveau (niveau très Bas : LLA, niveau Bas : LL, niveau Moyen : NL, niveau Haut : HL Niveau très Haut : HHL) dans le ballon de système anti béliet pour protéger les pompes en cas d'une éventuel vague de pression.



Figure 3-10 : Capteur de niveau

#### 3.3.9.2 Les capteurs de température

Dans notre station on utilise pour chaque pompe 3 capteurs de température pour les bobines et 2 capteurs de température pour les paliers. Ces capteurs sont utilisés pour signaler des alarmes en cas de dépassement des seuils.



Figure 3-11 : Capteur de température

#### 3.3.9.3 Les capteurs de vibration

Ces capteurs sont utilisés pour mesurer le taux de vibration des bobines et des paliers pour signaler des alarmes en cas de dépassement des seuils limites.





**Figure 3-12** : Capteur de vibration

### 3.3.10 Débitmètre

Il mesure le débit et le transmet vers l'automate. Ainsi on peut régler la vitesse nécessaire pour éviter la marche à sec et cela dans le cas où le niveau du réservoir soit très bas.



**Figure 3-13** : Débitmètre

## 3.4 Algorithme de procès

L'Algorithme du procès est géré par l'automate, cependant l'opérateur peut intervenir durant le Procès pour changer sa configuration. Pour modifier l'état du Procès l'opérateur dispose soit des boutons et commutateurs placés sur le pupitre soit sur le superviseur (Simatic HMI KTP400 Basic), il y aura une interface pour éventuellement changer et observer l'état du Procès.

L'Automate S7-1200 aura pour rôle essentiel de gérer les actionneurs tel que les groupes électropompe, les vannes motorisées, Il assurera la sécurité du procès afin d'éviter toutes fausse manipulation de la part de l'opérateur, mais aussi il permettra la communication entre le variateur de vitesse (SINAMICS V20), les démarreurs progressives (SIRIUS 3RW40) et le superviseur (HMI KTP400).

L'automate va tester les démarreurs progressifs et le variateur en les initialisent pour réinitialiser leur état et pour vérifier s'il y aurait une possibilité d'une anomalie.

Par mesure de sécurité pour les moteurs, le démarrage des pompes se fait à vide (c'est-à-dire sans charge, ici la charge est l'eau à travers la canalisation) pour éviter toutes éventuelles de surchauffe des pompes. Le démarrage d'une pompe se passe pendant un court instant c'est le régime transitoire qui n'est rien d'autre que : le temps que met la dynamique du système a passé d'une valeur initiale (ici pompe à l'arrêt) à une valeur finale (la pompe est arrivée à sa vitesse nominale).

Une fois que la pompe est atteint sur régime permanent (c'est-à-dire que l'appel de courant à disparut et le courant c'est stabiliser à sa valeur nominale), l'automate actionnera alors l'ouverture des vannes.

L'indicateur de niveau (LS 01) et le contrôleur de niveau à Ultrason (LC 01) donneront l'image du contenu du Réservoir R100.

- Si le capteur indique un niveau très bas (LLL) alors les pompes seront toutes actionnées
- Si le capteur indique un niveau moyen (LLA) alors on activera soit une ou deux pompe(s)
- Si le capteur indique un niveau haut (HL) alors on activera une seule pompe.

Les indicateurs de niveaux seront affichés sur le superviseur sous la forme d'une jauge en temps réel ainsi que l'état de l'activation des pompes.

Par mesures de sécurité l'automate gèrera le fonctionnement des pompes, si une pompe décèle un défaut de vibration (VBS) ou de température du bobinage (TW) ou la température du pallier moteur (TB) ou autre défaillance technique alors l'automate arrêtera la pompe en question et le superviseur affichera le nature de la défaillance de la (ou des) pompe(s) en question.

### ***Le superviseur***

Le superviseur est un écran tactile de contrôle de l'ensemble de l'application du procès, l'application est programmable avec des logiciels de type SCADA (SIMATIC WinCC V16). On peut à partir de cet écran de superviseur observé les fonctions en temps réel suivantes qui seront disponibles :

- Etat du Procès
- Etat de la distribution électrique
- Valeurs des mesures électriques provenant des tableaux de distribution MT

- Valeurs des mesures analogiques de niveau débit pression etc....
- Choix et visualisation du mode de fonctionnement
- Signal d'anomalie sonore et visuelle
- Consignation et impression des états et des défauts
- Edition de courbes historiques
- Edition de rapports journaliers, mensuels et annuels

### ***Gestion des taches du système***

La tâche de POMPAGE se décompose en plusieurs autres tâches :

- Tâche de chargement par le Réservoir R100 avec le contrôle de ses niveaux
- Tâche gestion des Groupes de pompage (SP107-A, SP107-B et SP107-C).
- Tâche gestion des Sécurités (Compresseur C101 et Anti-Bélier V101, etc.).
- Tâches gestion des Défauts et Supervision Process.

Les différentes tâches sont gérées par l'automate, on y trouve des signaux

- Des entrées en TOR (Tout ou Rien)
- Des entrées Ana (Analogique)
- Des sorties TOR
- Des sorties Ana

L'automate, via la carte Profibus communique avec les Démarreurs et le Variateur.

### ***Gestion des Anomalies***

#### Défaut, Alarmes, définition:

On appelle défaut, un événement dont l'origine est une anomalie physique. On définit l'alarme comme une représentation visuelle du défaut. L'alarme est l'information issue du défaut. Elle est transmise à l'opérateur par l'intermédiaire du terminal IHM.

On distingue plusieurs types de défauts :

- **Les défauts câblés** (ou défauts électriques). Ces défauts sont acquis directement par câblage depuis les armoires électriques. Ils correspondent à des dysfonctionnements liés à la sécurité des hommes : arrêts d'urgences etc. et la protection des machines : les défauts des départs actionneurs (thermiques). Ces défauts sont surveillés par le système en permanence. Une alarme de ce défaut est générée.
- **Les défauts générés par le programme**, ils peuvent être de plusieurs natures :

- Ceux liés à une anomalie physique d'un ou plusieurs constituants de l'équipement : exemple discordance entre la commande et le retour d'état du contacteur, rupture de fils d'une mesure...
- Les défauts système. Ils correspondent à des anomalies physiques liées aux organes du système (par exemple : défaut de carte).

### **Procédure d'acquiescement des défauts**

Lorsqu'un défaut apparaît, une alarme est obligatoirement associée à celui-ci. L'opération d'acquiescement est réalisée par l'opérateur sur le terminal IHM en deux étapes :

- La prise en compte effective de l'alarme sur la page d'alarme associée au défaut en appuyant sur une touche spécifique.
- L'acquiescement proprement dit du défaut en sélectionnant la page d'exploitation de l'équipement associé au défaut et en validant l'acquiescement (cet acquiescement n'est effectif que si l'événement qui est à l'origine du défaut a disparu).

### **Défaut électrique :**

Ce défaut câblé regroupe les dysfonctionnements liés à la chaîne de commande du moteur : thermique, limiteur de couple, etc. Il est généré quel que soit le mode de marche de l'actionneur.

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 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 d'un actionneur.

Lorsque l'automate donne un ordre à un actionneur (exemple : marche ou arrêt), 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), 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.

### **3.5 Conclusion**

Ce chapitre décrit l'ensemble des fonctions qui sont représentées dans la station de KOUBA 107. Une telle description est nécessaire pour comprendre le fonctionnement de la station et par la suite pouvoir créer les sections du programme de gestion de la station.

# **Chapitre 4 :**

# **Programmation Et Supervision**

## 4.1 Introduction

Dans ce chapitre, on créera cinq Programmes principaux :

- Le Programme Anti bélièr.
- Le Programme Gestion de la vanne.
- Le Programme Autorisation du démarrage de la pompe.
- Le Programme Défaut : qui résume les différents défauts de la station.
- Le Programme GEP : pour la gestion du groupe électropompe.

Tous les programmes sont écrits en langage CONT (LADDER) intégré dans le logiciel TIA PORTAL suite à son avantage de souplesse et de facilité de compréhension par les autres programmeurs. La supervision des trois parties : l'anti bélièr, les pompes et les vannes de la station de pompage a été aussi développées dans le WINCC qui est intégré au TIA PORTAL.

## 4.2 Programme du système anti bélièr

### 4.2.1 Cahier de charge du Programme anti bélièr

Le ballon anti bélièr comprend pour la partie automatisation : 5 capteurs de niveau, une vanne et un compresseur, du même le bloc contient les signaux des 5 capteurs en entrée, et doit envoyer de plus des deux signaux d'alarme de très haut et très bas niveau, un signal d'activation de la vanne et du compresseur, et dans le cas normal un autre signal d'autorisation est activé.

Les 5 entrées :

- HHL : niveau très haut de l'anti bélièr.
- HL : niveau haut.
- NL : niveau normal ou moyen.
- LL : niveau bas.
- LLA : niveau très bas.

Les 5 sorties :

- sec\_niveau\_haut : sécurité niveau très haut (lancement d'une alarme lorsque le niveau est très haut)
- sec\_niveau\_bas : sécurité niveau très bas (lancement d'une alarme lorsque le niveau est très bas)
- Auto : autorisation du démarrage du groupe électropompe.

- Comp : démarrage du compresseur.
- cmd\_ouv\_vanne : commande ouverture de la vanne de l'anti béliér.

HL	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
HHL	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NL	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
LL	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
LLA	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
comp	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sec_niveau_haut	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sec_niveau_bas	Bool	%Q1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Auto	Bool	%Q1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
cmd_ouv_vanne	Bool	%Q1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

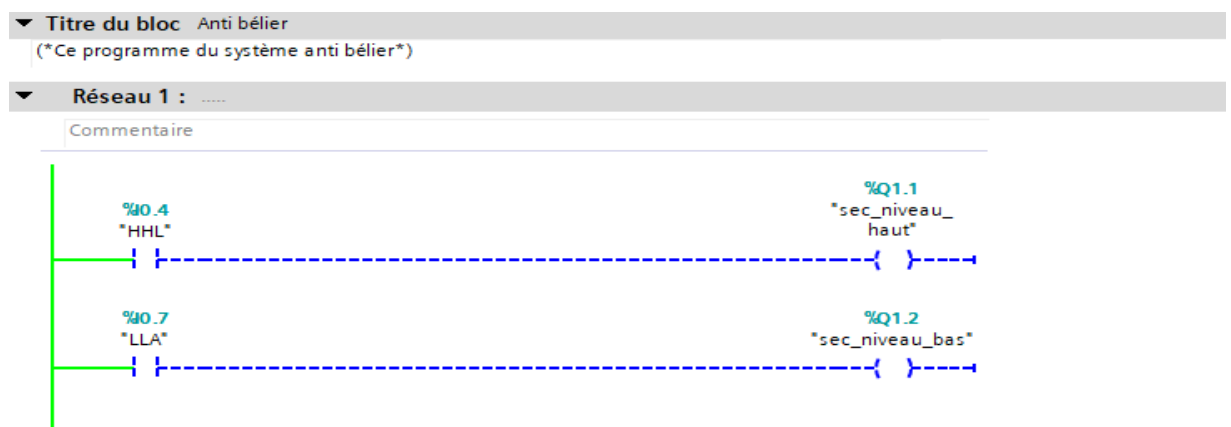
Figure 4-1 : Table variable du Programme « anti béliér »

**Le Programme anti béliér doit assurer :**

- Activation des sorties sec\_niveau\_haut et sec\_niveau\_bas dans le cas où les signaux HHL et LLA sont actives.
- Garder le niveau de l'eau dans l'anti béliér entre HL et LL, ainsi :
  - Si le niveau est HL on démarre le compresseur, ce dernier exerce une pression sur l'air dans l'anti béliér et par la suite pouvoir baisser le niveau d'eau jusqu'à atteindre le niveau NL.
  - Si le niveau est LL on démarre la vanne, ceci permettra l'évacuation de l'air dans l'anti béliér et par la suite pouvoir augmenter le niveau d'eau jusqu'à atteindre le niveau NL.
- Le signal autorisation est actif lorsque le niveau d'eau soit compris entre HL et LL.

### 4.2.2 Le programme de l'anti béliér

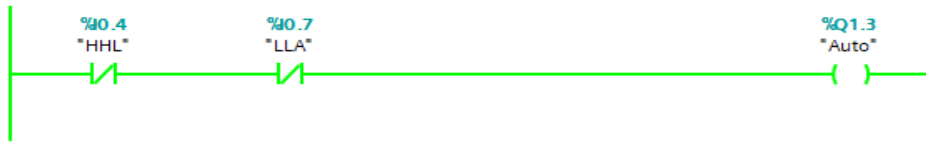
La section du programme est écrite en langage LADDER (CONT)





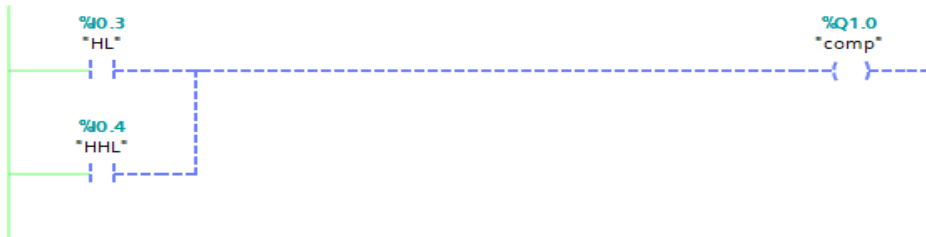
▼ Réseau 2 : .....

(\*On autorise le démarrage du groupe électropompe si les niveau d'eau dans l'anti béliet est bon : ni HHL ni LLA \*)



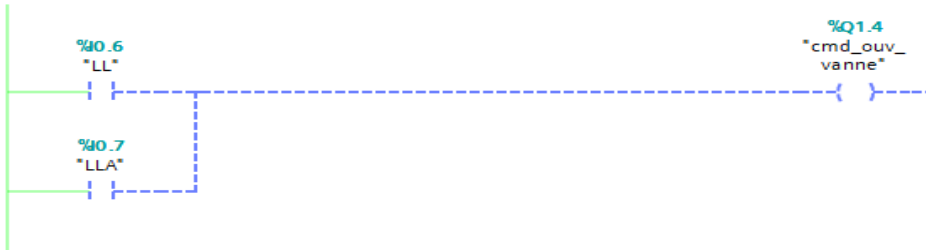
▼ Réseau 3 : .....

(\*démarrage du compresseur\*)



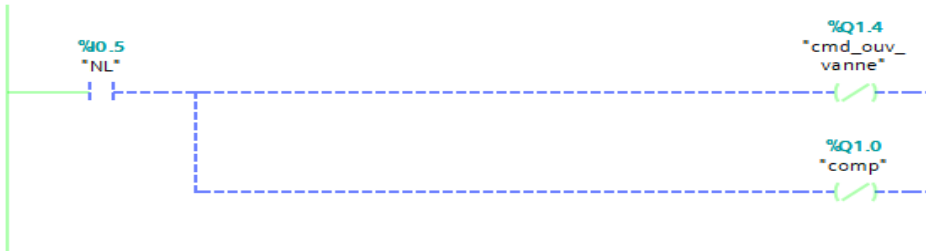
▼ Réseau 4 : .....

(\*ouverture de la vanne \*)



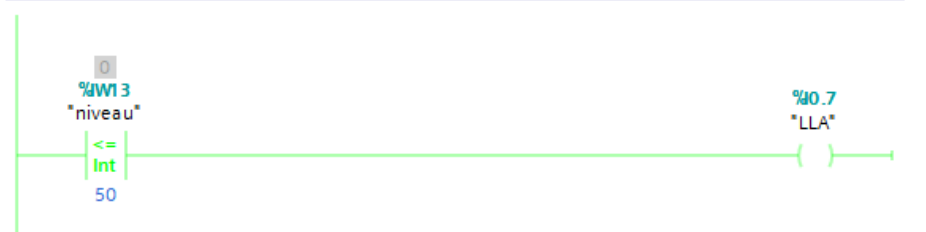
▼ Réseau 5 : .....

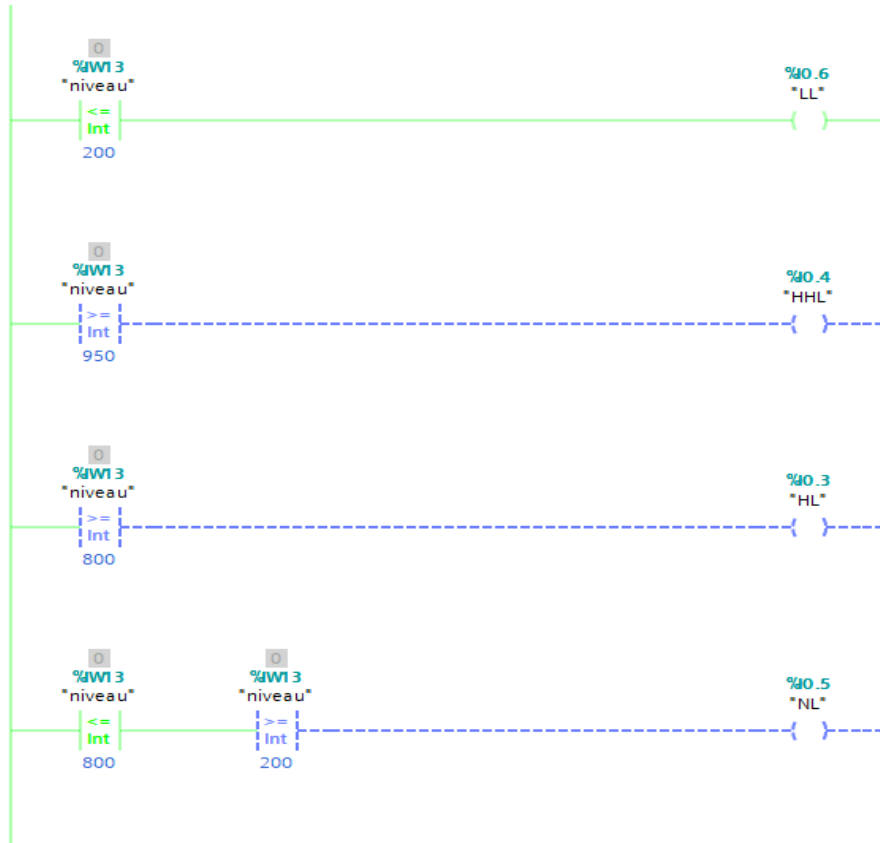
(\*Si le niveau est normal : alors on arrête le compresseur et on ferme la vanne\*)



▼ Réseau 6 : Les niveaux

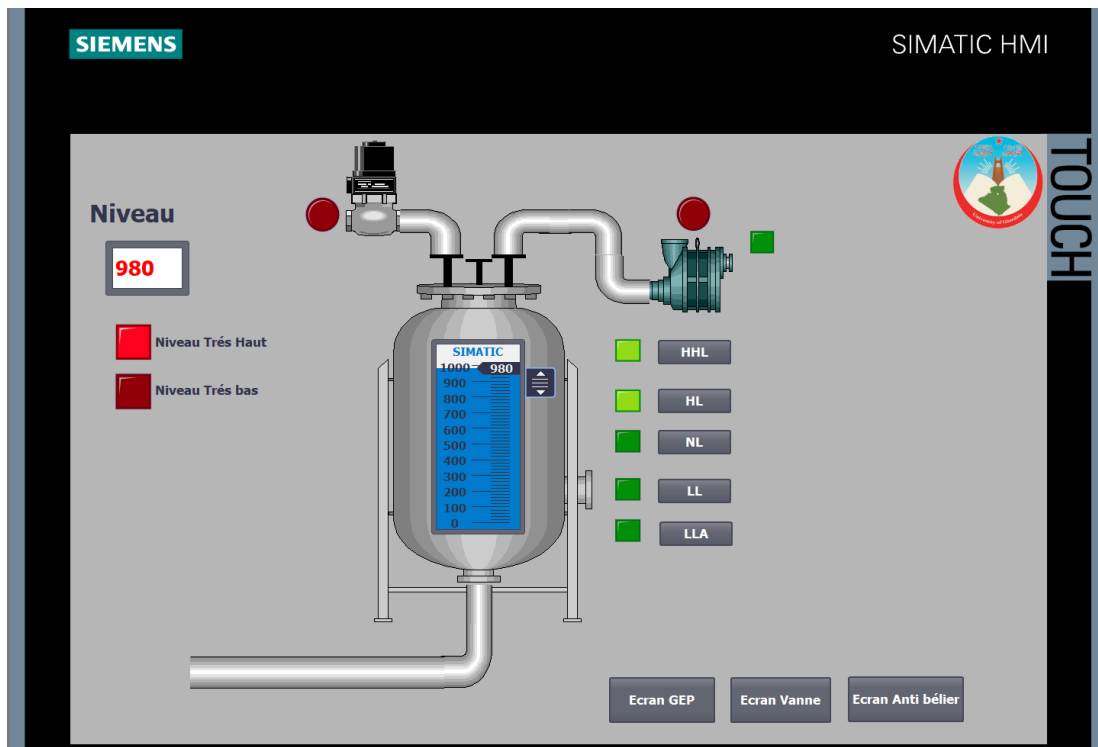
Commentaire





### 4.2.3 La simulation de l'anti bélier

La figure ci-dessous illustre la simulation à l'aide d'un logiciel « WinCC IHM » intégré dans « TIA PORTAL ».



### 4.3 Le Programme gestion de la vanne

#### 4.3.1 Cahier de charge du Programme Gestion de la vanne

La vanne motorisée reçoit les signaux d'ouverture et de fermeture et peut envoyer les deux signaux indiquant son état (complètement ouverte ou complètement fermée). Le Programme reçoit 6 signaux et doit générer 5 signaux :

Les 6 entrées :

- ouvrir : Commande ouvrir la vanne (signal envoyé par l'utilisateur ou un autre programme)
- fermer : Commande fermer la vanne (signal envoyé par l'utilisateur ou un autre programme)
- fco : Fin de course vanne ouverte (signal générer par la vanne)
- fcf : Fin de course vanne fermée (signal générer par la vanne)
- default : Défaut de la vanne
- aquit : Acquiescement d'un défaut

Les 5 sorties :

- cmd\_ou : Commande d'ouverture de la vanne (signal envoyé vers la vanne)
- cmd\_fe : Commande de fermeture de la vanne (signal envoyé vers la vanne)
- dis\_o : Discordance ouverture
- dis\_f : Discordance fermeture
- incoh : Incohérence

	ouvrir	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	fermer	Bool	%I1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	fco	Bool	%I1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	fcf	Bool	%I1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	default	Bool	%I1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	aquit	Bool	%I1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	cmd_ou	Bool	%Q1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	cmd_fe	Bool	%Q1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	dis_o	Bool	%Q1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	dis_f	Bool	%Q2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	incoh	Bool	%Q2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 4-2 : Table variable du Programme « Gestion de la vanne »

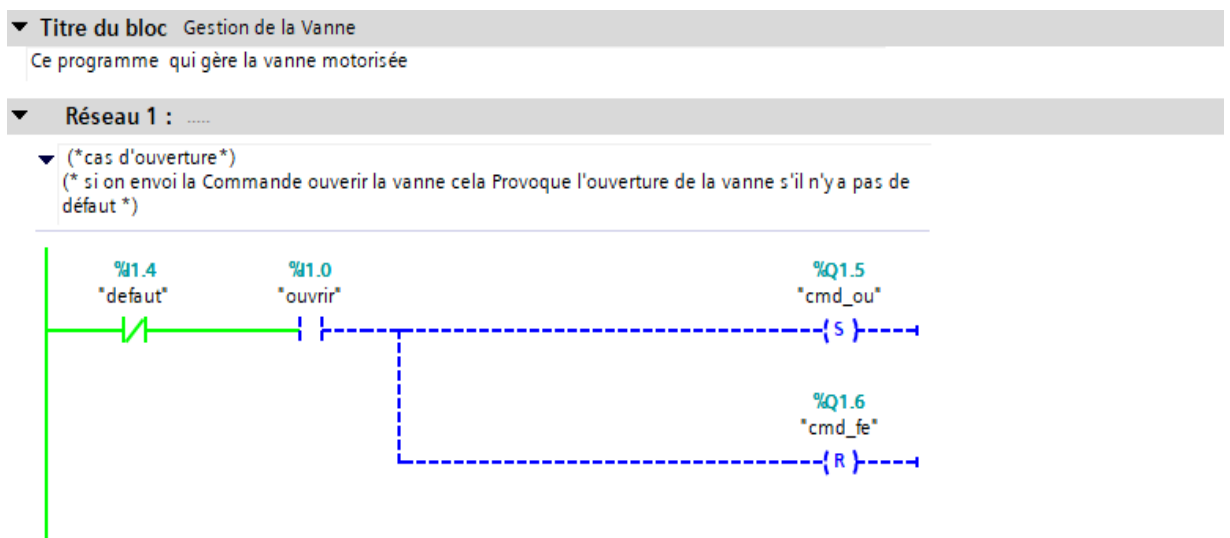
Le Programme vanne doit assurer :

- La génération des signaux d'ouverture et de fermeture de la vanne en cas d'absence de défaut.

- La génération des signaux discordance d'ouverture et de fermeture :
  - La discordance d'ouverture est générée 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 due à son inertie). Pour des raisons de sécurité ce temps sera une variable locale et non pas d'entrée pour le bloc.
  - De même pour la discordance fermeture.
- La génération du signal incohérence lorsque on reçoit les deux signaux fcf et fco en même temps pendant une durée réglée par le programmeur (variable locale) et cela de peur que la présence de ces deux signaux n'est qu'un parasite.
- L'acquiescement des défauts : discordance d'ouverture et de fermeture et incohérence. L'acquiescement de ces défauts n'est appliqué qu'à l'absence du défaut et après un appuie sur le bouton aquit :
  - Pour la discordance d'ouverture, l'absence de défaut signifie que la vanne est pu être ouverte.
  - Pour la discordance de fermeture, l'absence de défaut signifie que la vanne est pu être fermée.
  - Pour l'incohérence, l'absence de défaut signifie que les deux signaux fcf et fco ne sont plus actifs en même temps vanne est pu être ouverte.

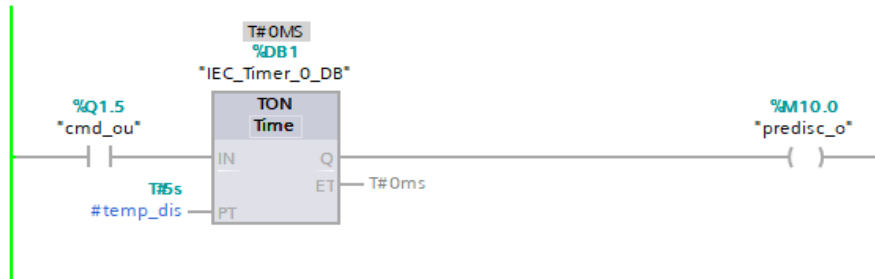
### 4.3.2 Le programme de la gestion de la vanne

La section du programme est écrite en langage LADDER (CONT)



▼ Réseau 2 : .....

- ▼ (\* lancer une temporisation "temps maximum nécessaire pour l'ouverture de la vanne" \*)  
(\*predisc pour dire seulement que la temporisation est terminée \*)



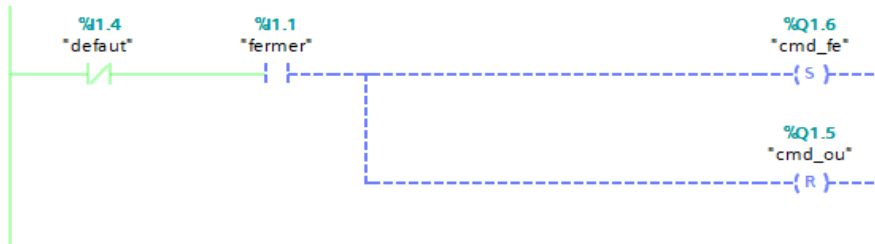
▼ Réseau 3 : .....

- ▼ (\*si la temporisation est terminée et qu'on n'a pas reçu le signal fco "fin d'ouverture de la vanne")  
(\* alors on envoi un signal d'erreur discordance \*)



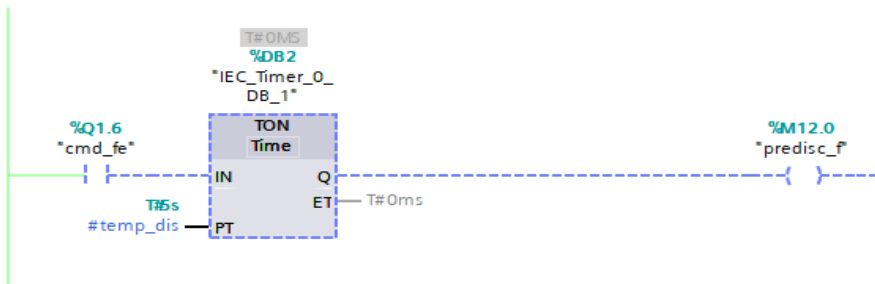
▼ Réseau 4 : .....

- ▼ (\*cas de fermeture c'est les Mêmes tapes \*)



▼ Réseau 5 : .....

Commentaire



▼ Réseau 6 : .....

Commentaire



▼ Réseau 7 : .....

▼ Les commandes d'ouverture et de fermeture de la vanne (\*s'arrêtent après recevoir des signaux fco et fcf respectivement\*)



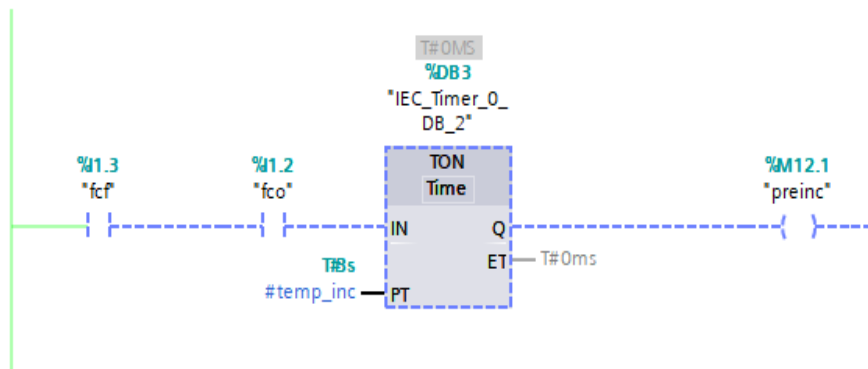
▼ Réseau 8 : .....

Commentaire



▼ Réseau 9 : .....

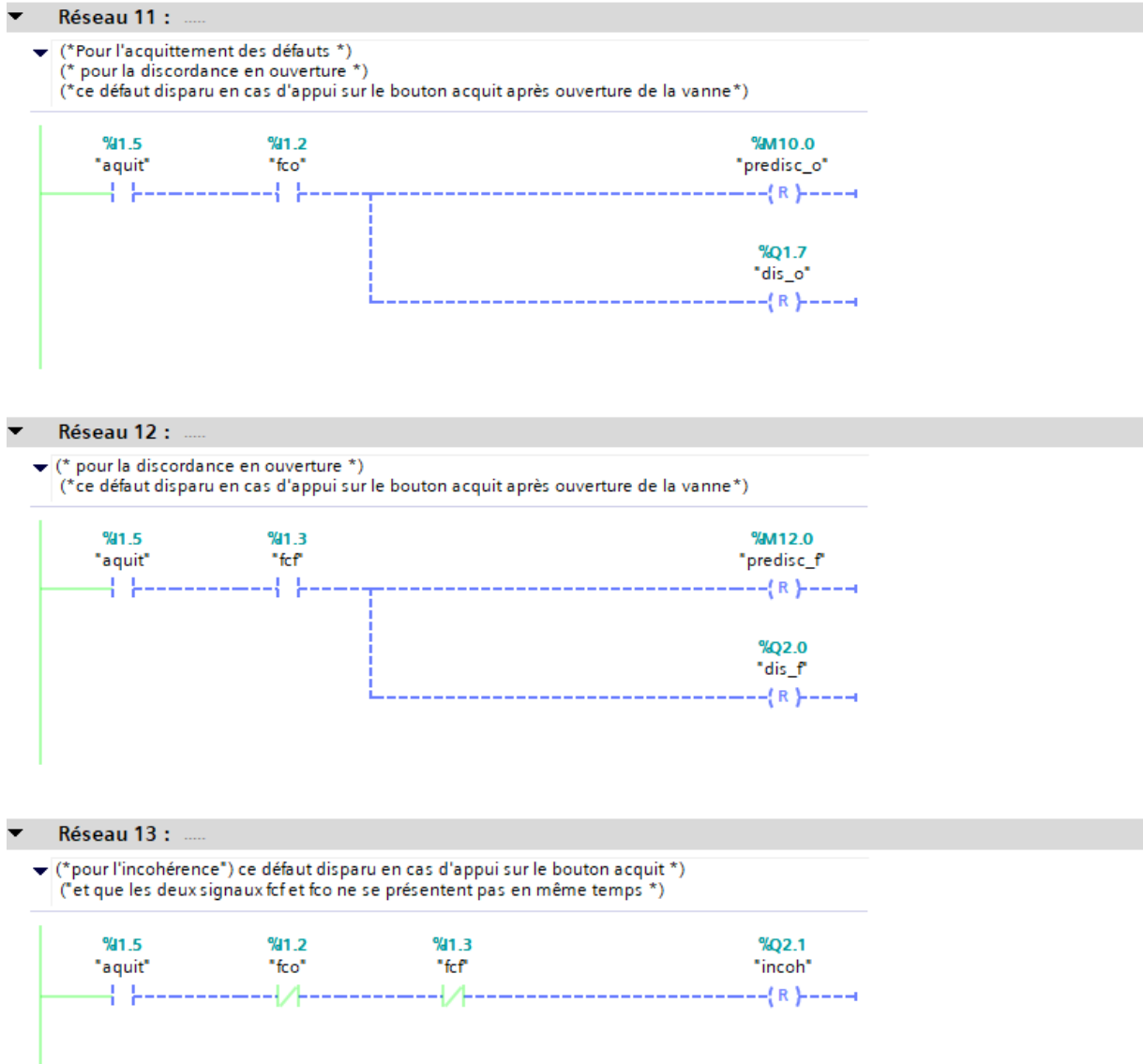
▼ (\*Test de l'incohérence\*)  
 (\*dès que les deux signaux fcf et fco arrivent en même temps en commence à calculer le temps écoulé (preinc) avec temps=1s \*)



▼ Réseau 10 : .....

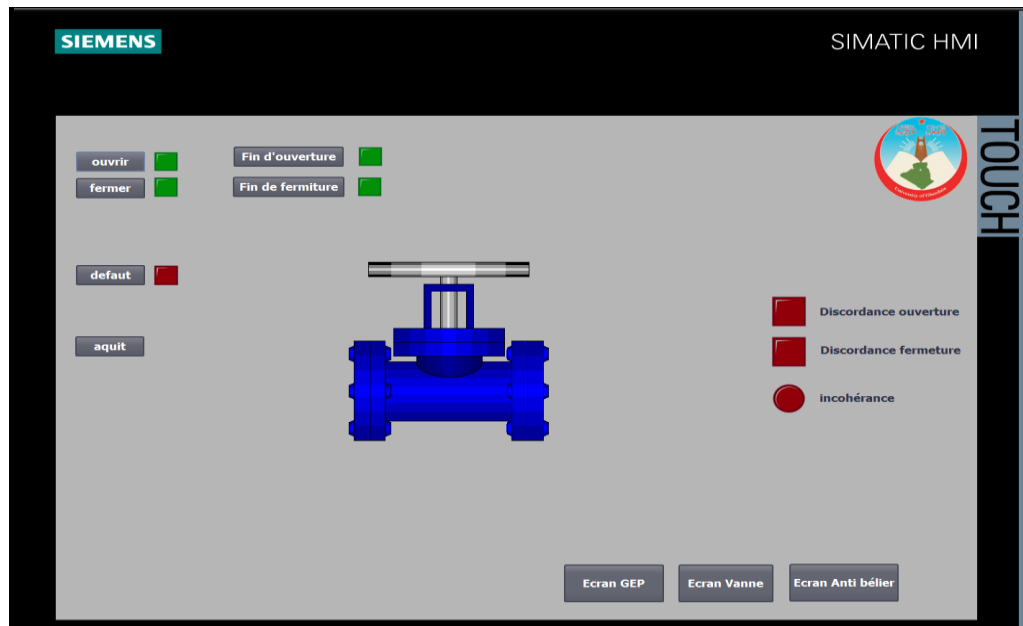
(\*si le temps ecoulé = temps\_inc alors on signale l'incohérence\*)





### 4.3.3 La simulation du Programme Gestion de la vanne

La figure ci-dessous illustre la simulation à l'aide d'un logiciel « WinCC IHM » intégrant dans « TIA PORTAL ».



#### 4.4 Le Programme autorisation du démarrage du GEP

Avant de démarrer le GEP il est nécessaire de vérifier plusieurs facteurs qui entrent en jeu :

- Le facteur électrique (Transformateurs et groupe électrogène) permettant de savoir le nombre de GEP autorisés à se mettre en marche,
- Nombre de GEP en marche,
- Nombre de GEP en phase de démarrage,
- Nombre de GEP en phase d'arrêt

##### 4.4.1 Cahier de charge du Programme Autorisation

Le rôle principal de ce Programme est l'envoi d'un signal d'autorisation au GEP après la vérification de toutes les conditions nécessaires pour le démarrage.

Dans notre Programme nous avons traitées le cas de 6 GEP, 2 transformateurs, 1 groupe électrogène.

Les entrées :

- tansfo\_T1: le 1er Transformateur en fonctionnement.
- tansfo\_T2: le 2ème Transformateur en fonctionnement.
- Groupe\_EG : groupe électrogène en fonctionnement.
- GEP1\_en\_DEM : le GEP1 est en phase de démarrage.
- GEP2\_en\_DEM : le GEP2 est en phase de démarrage.
- GEP3\_en\_DEM : le GEP3 est en phase de démarrage.



- GEP4\_en\_DEM : le GEP4 est en phase de démarrage.
- GEP5\_en\_DEM : le GEP5 est en phase de démarrage.
- GEP6\_en\_DEM : le GEP6 est en phase de démarrage.
- GEP1\_en\_ARRET: le GEP1 est en phase d'arrêt.
- GEP2\_en\_ARRET: le GEP2 est en phase d'arrêt.
- GEP3\_en\_ARRET: le GEP3 est en phase d'arrêt.
- GEP4\_en\_ARRET: le GEP4 est en phase d'arrêt.
- GEP5\_en\_ARRET: le GEP5 est en phase d'arrêt.
- GEP6\_en\_ARRET: le GEP6 est en phase d'arrêt.
- GEP1\_en\_Marche : le GEP1 est en marche.
- GEP2\_en\_Marche : le GEP2 est en marche.
- GEP3\_en\_Marche : le GEP3 est en marche.
- GEP4\_en\_Marche : le GEP4 est en marche.
- GEP5\_en\_Marche : le GEP5 est en marche.
- GEP6\_en\_Marche : le GEP6 est en marche.

La sortie :

- Autoris: signale qui donne l'autorisation de démarrage au GEP.

	tansfo_T1	Bool	%I1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	tansfo_T2	Bool	%I1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Groupe_EG	Bool	%I2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP1_en_DEM	Bool	%I2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP2_en_DEM	Bool	%I2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP3_en_DEM	Bool	%I2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP4_en_DEM	Bool	%I2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP5_en_DEM	Bool	%I2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP6_en_DEM	Bool	%I2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP1_en_ARRET	Bool	%I2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP2_en_ARRET	Bool	%I3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP3_en_ARRET	Bool	%I3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP4_en_ARRET	Bool	%I3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP5_en_ARRET	Bool	%I3.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP6_en_ARRET	Bool	%I3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP1_en_Marche	Bool	%I3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP2_en_Marche	Bool	%I3.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP3_en_Marche	Bool	%I3.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP4_en_Marche	Bool	%I4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP5_en_Marche	Bool	%I4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	GEP6_en_Marche	Bool	%I4.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Autoris	Bool	%Q2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 4-3 : Table variable du Programme « Autorisation »

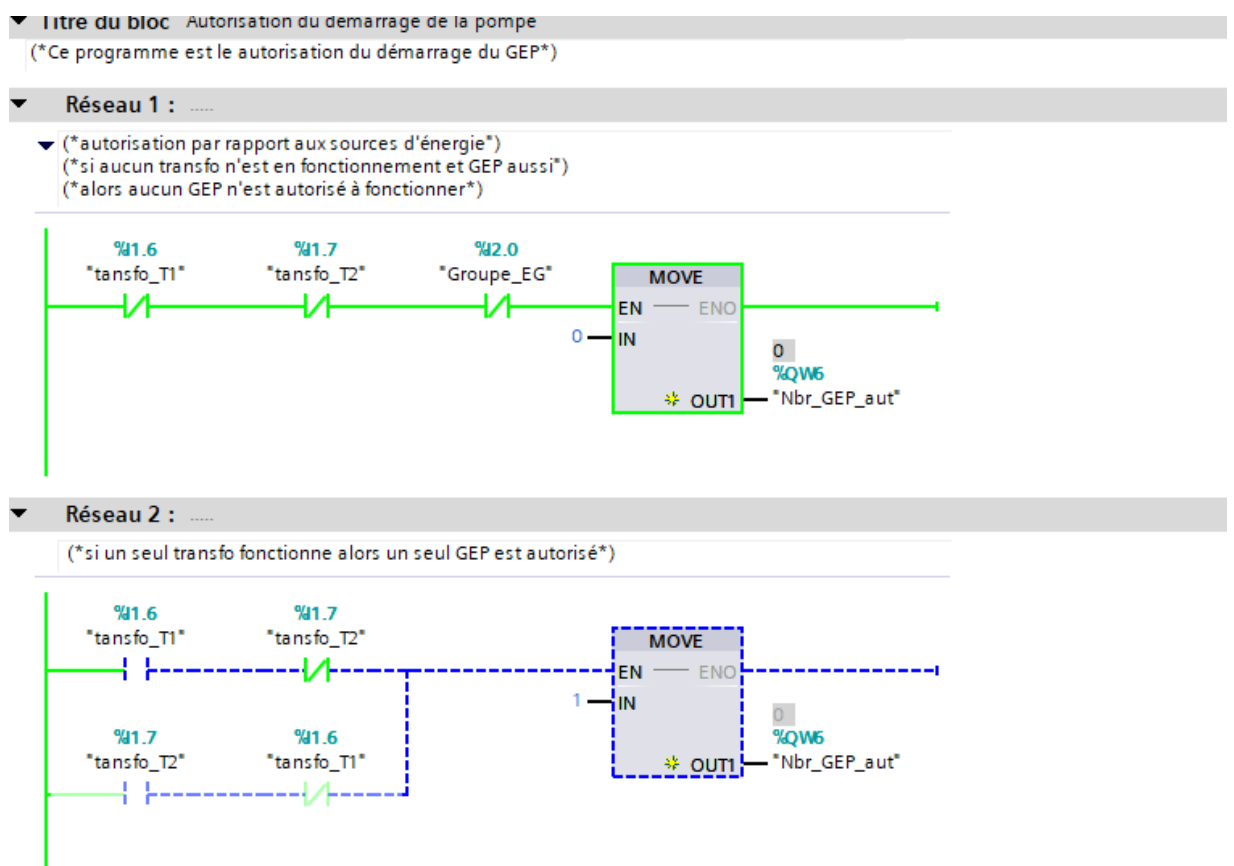
*Le Programme d'autorisation doit*

- Définir le nombre de GEP autorisée à se mettre en marche selon le nombre des équipements électriques en fonctionnement :

- Si les 2 transformateurs sont en fonctionnement alors 3 GEP sont autorisés à se mettre en marche.
- Si un des 2 transformateurs est en fonctionnement alors 1 GEP autorisé à se mettre en marche.
- Si le groupe électrogène est en fonctionnement alors 2 GEP autorisés à se mettre en marche.
- Si le nombre de GEP autorisés est supérieur au nombre de GEP en marche alors l'autorisation de marche de GEP est activé (désactivé dans le cas contraire).
- Si aucun GEP n'est en phase de démarrage ou en phase d'arrêt alors l'autorisation de démarrage de GEP activé (désactivé dans le cas contraire).

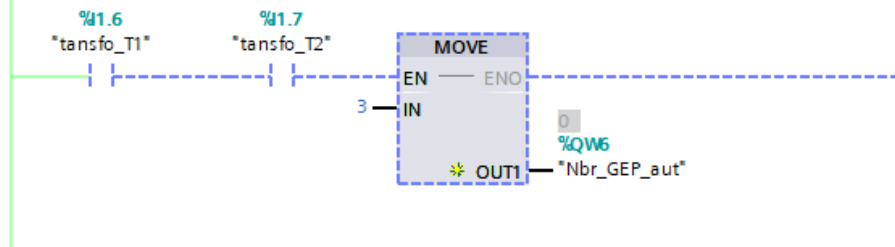
### 4.4.2 La programmation de l'Autorisation

La section du programme est écrite en langage LADDER (CONT)



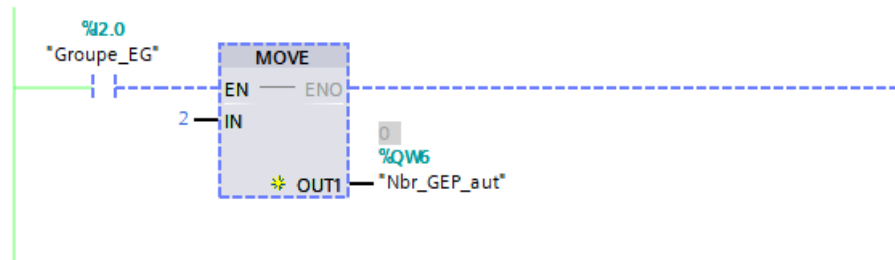
Réseau 3 : .....

(\*si les é transfos fonctionnent : 2 GEP sont autorisés\*)



Réseau 4 : .....

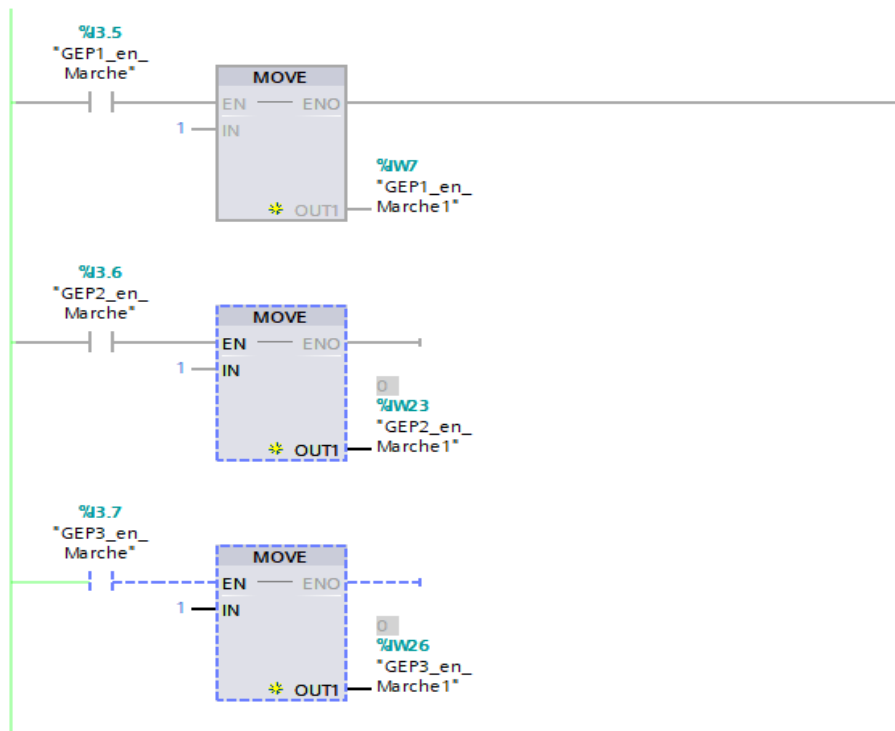
(\*s'il n'y a que le Groupe électrogène en fonctionnement : 2 GEP sont autorisés\*)

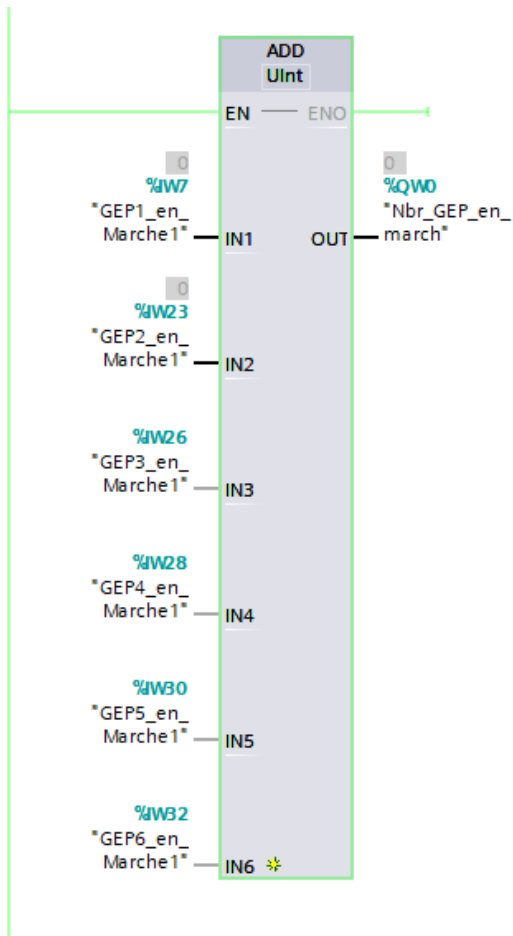
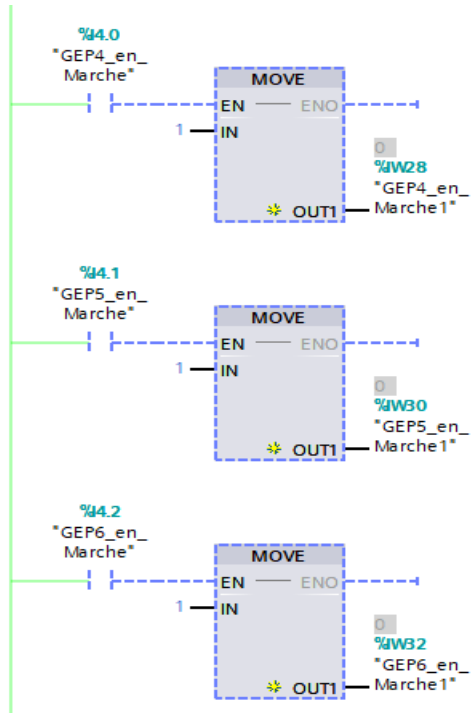


Réseau 5 : .....

Bool to Uint (MOVE)

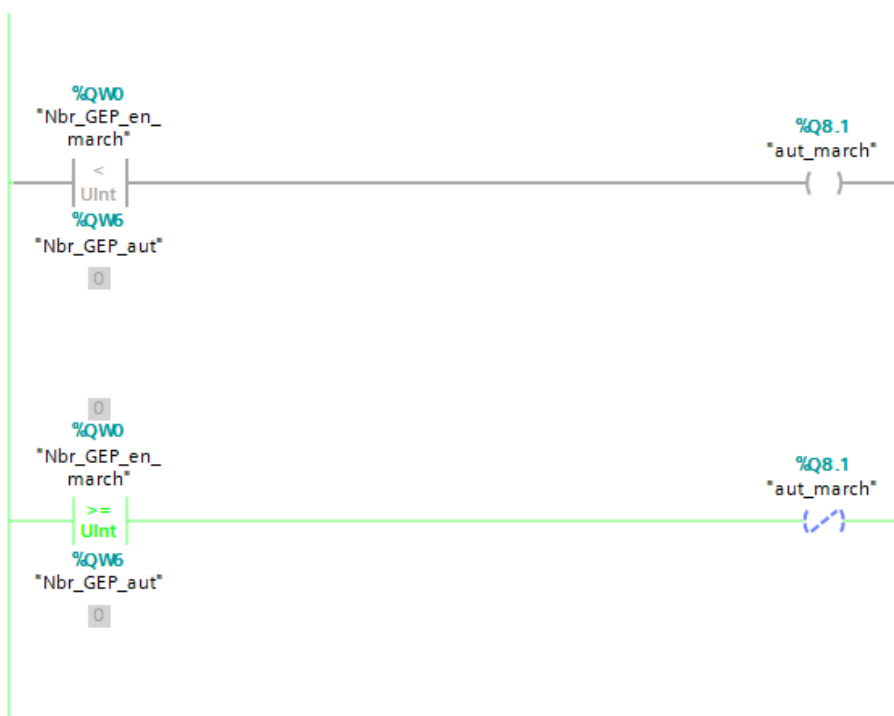
(\*nombre total des GEP en marche = la somme arithmétique de ces derniers\*) (ADD)





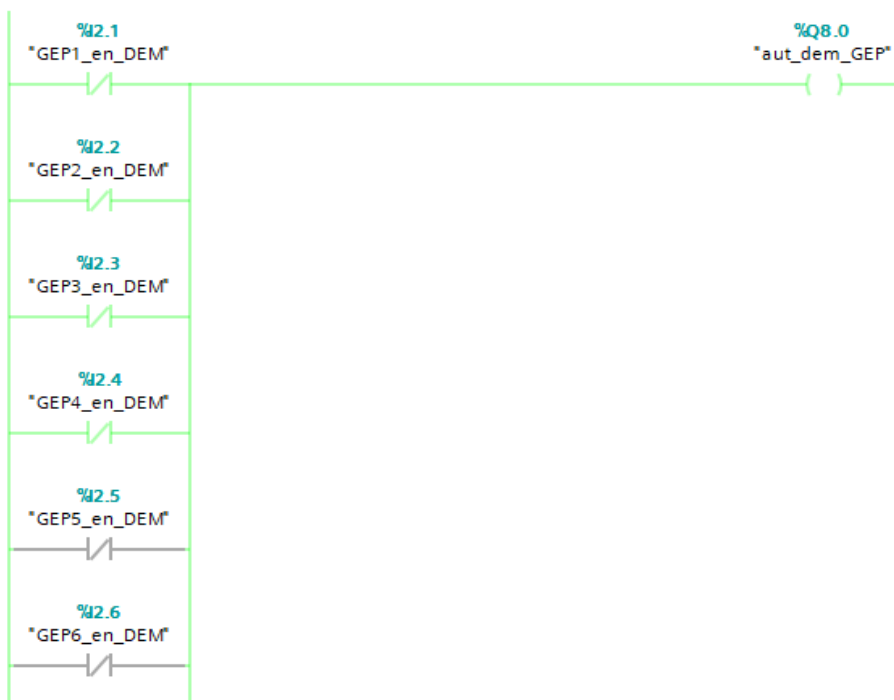
Réseau 6 : .....

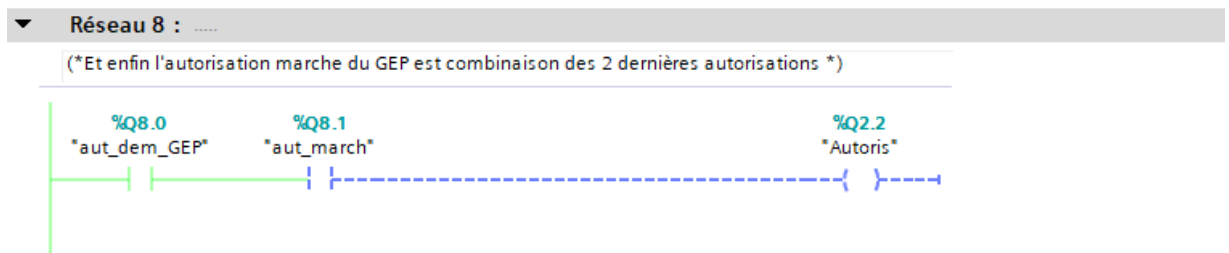
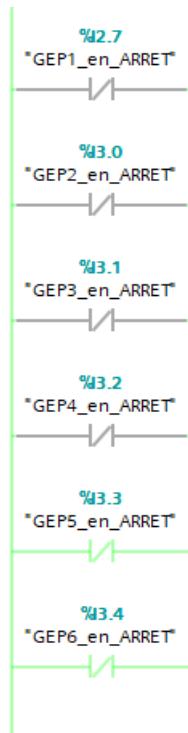
(\*on autorise le démarrage d'un autre GEP si le nombre total des GEP en marche est inferieur aux nombres des GEP autorisés à fonctionner\*)



Réseau 7 : .....

(\*On autorise un autre GEP à démarrer s'il n'y a aucun autre en phase de démarrage \*)





## 4.5 Le Programme du Défaut

### 4.5.1 Cahier de charge du Défaut

Le but de ce Programme est la signalisation de 3 défauts principaux :

- Défaut de température des bobinages du moteur de la pompe, ce dernier est équipé de trois capteurs de température.
- Défaut de température des paliers (deux capteurs de température).
- Défaut de vibration du GEP.

Ce Programme présent 16 signaux en entrée et 4 en sorties

Les 16 entrées :

- temperatureB1, 2 et 3 : Températures des bobines de la pompe.
- temperatureB\_seuil : Température seuil des bobines de la pompe.
- temperatureP1 et 2 : Températures des paliers de la pompe.

- temperatureP\_seuil : Température seuil des paliers de la pompe.
- vibration : vibration mécanique de la pompe.
- vibration\_seuil : vibration seuil de la pompe.
- default\_van : défaut de la vanne.
- disjoncteur\_van : disjoncteur vanne.
- disjoncteur\_gep : disjoncteur GEP.
- default\_isolem : défaut isolement.
- default\_demmareur : défaut démarreur progressive (SIRIUS).
- GEP\_sous\_tension : GEP sous-tension (présence d'alimentation).
- acquit : pour l'acquittement des défauts.

Les 4 sorties :

- default\_temperatureB : défaut de température des bobinages.
- default\_temperatureP : défaut de température des paliers.
- default\_vibration : pour dire que la vibration a dépassé le seuil.
- default\_D : qui résume tous les défauts.

	temperatureB1	Int	%IW5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	temperatureB2	Int	%IW9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	temperatureB3	Int	%IW10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	temperatureB_seuil	Int	%IW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	temperatureP1	Int	%IW14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	temperatureP2	Int	%IW16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	temperatureP_seuil	Int	%IW18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	vibration	Int	%IW20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	vibration_seuil	Int	%IW22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_van	Bool	%I6.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	disjoncteur_van	Bool	%I6.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	disjoncteur_gep	Bool	%I6.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_isolem	Bool	%I7.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_demmareur	Bool	%I7.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	GEP_sous_tension	Bool	%I7.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	acquit	Bool	%I7.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_temperatureB	Bool	%Q9.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_temperatureP	Bool	%Q9.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_vibration	Bool	%Q9.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	default_D	Bool	%Q9.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

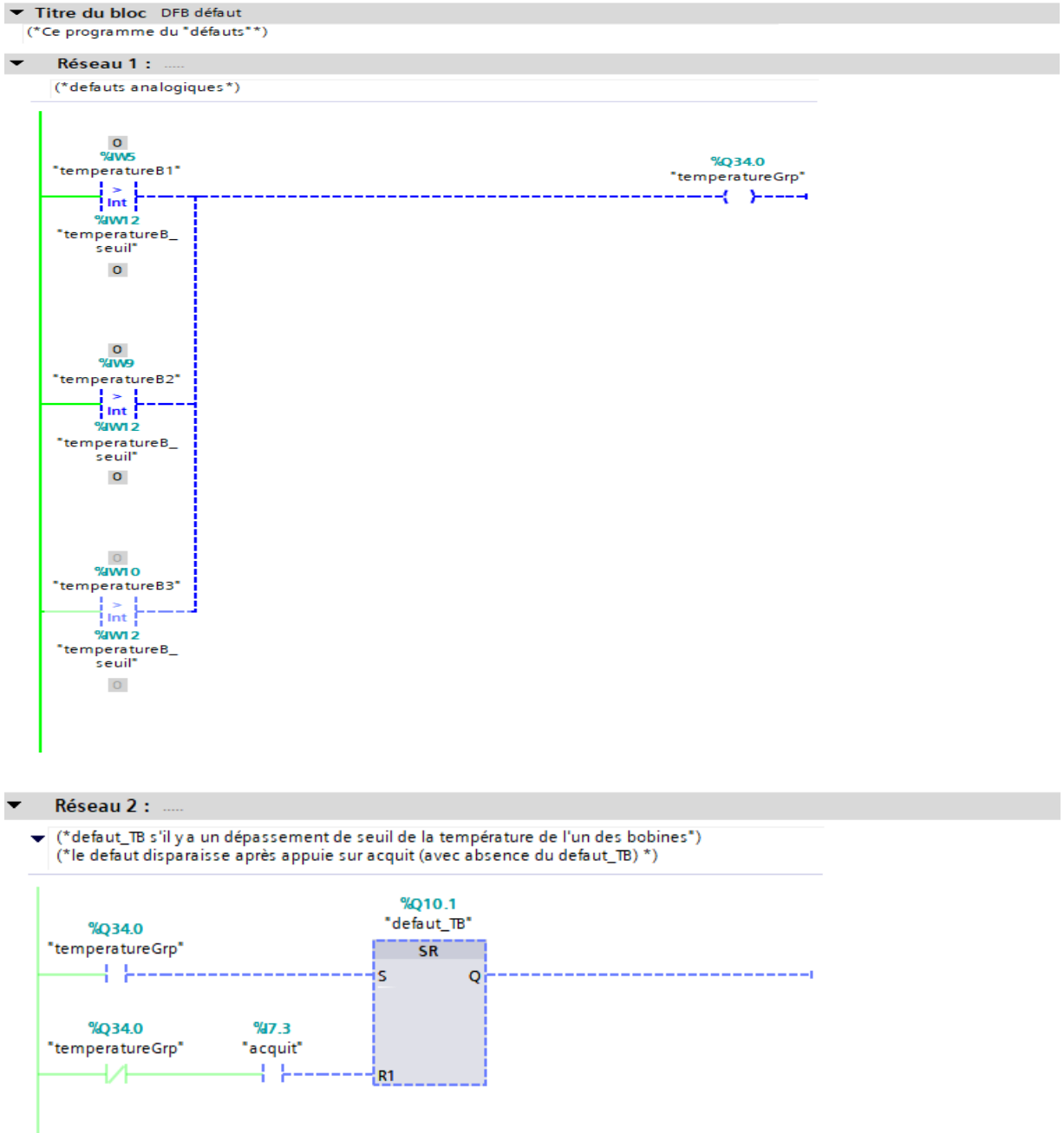
Figure 4-4 : Table variable du Programme « Défaut »

*Le Programme d'autorisation doit*

- Envoyer 4 signaux signalant l'existence d'un défaut et le préciser

### 4.5.2 La programmation du défaut

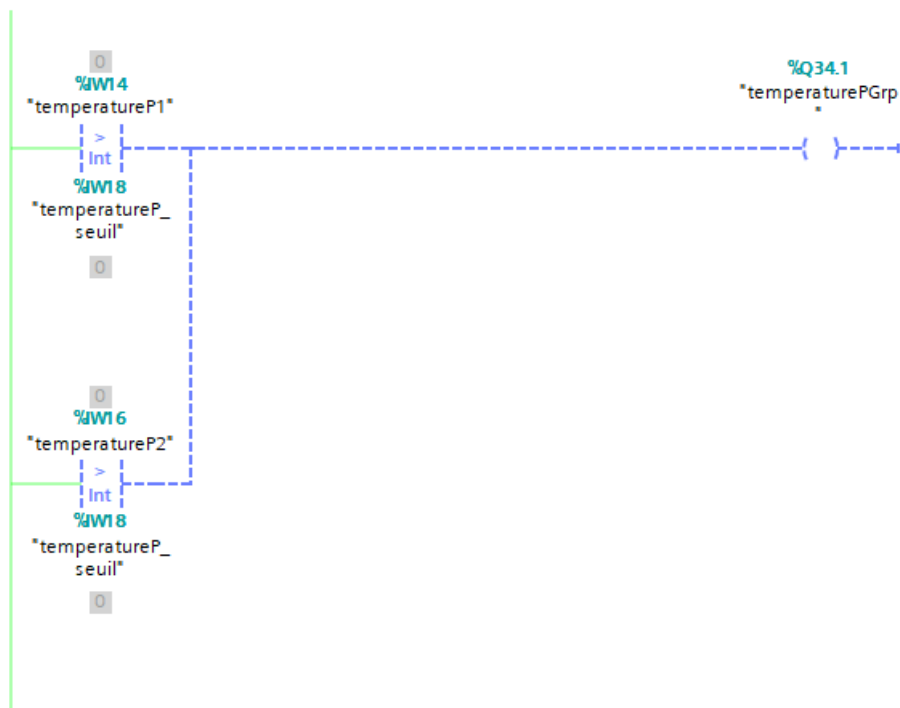
La section du programme est écrite en langage LADDER (CONT)





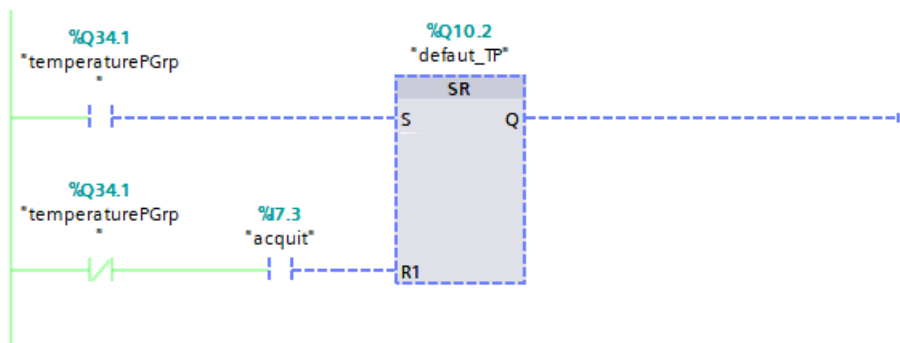
▼ Réseau 3 : .....

▼ (\*default\_TB s'il y a un dépassement de seuil de la température de l'un des paliers\*)  
 (\*le défaut disparaît après appui sur acquit (avec absence du default\_TP) \*)



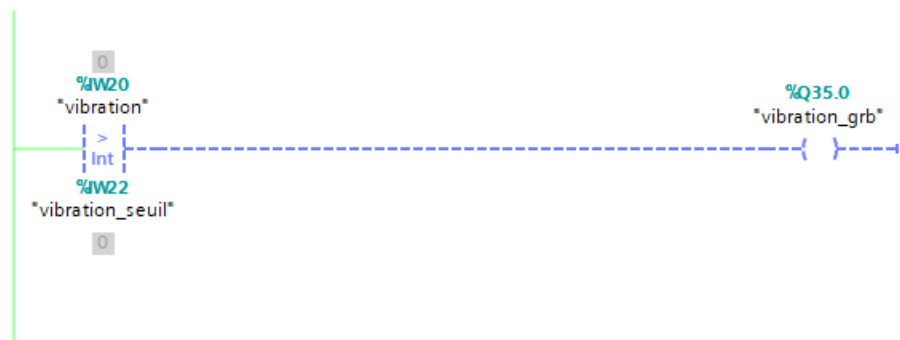
▼ Réseau 4 : .....

Commentaire



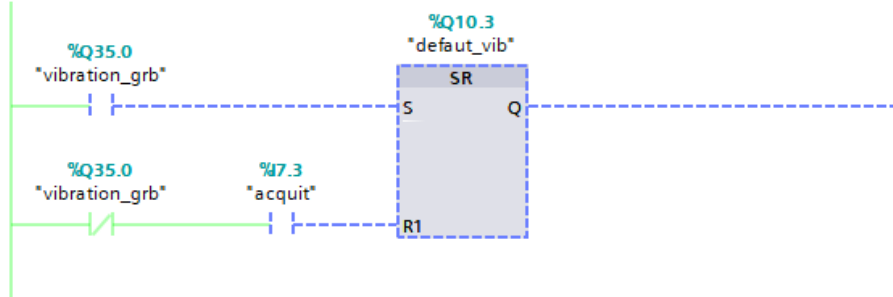
▼ Réseau 5 : .....

▼ (\*le défaut vibration en cas de dépassement de seuil de ce dernier\*)



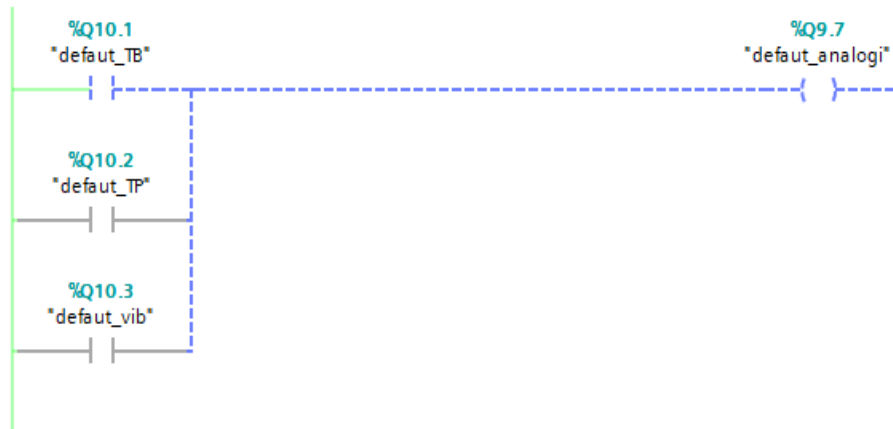
▼ Réseau 6 : .....

(\*le défaut disparaisse après appuie sur acquit (avec absence du défaut\_vib) \*)



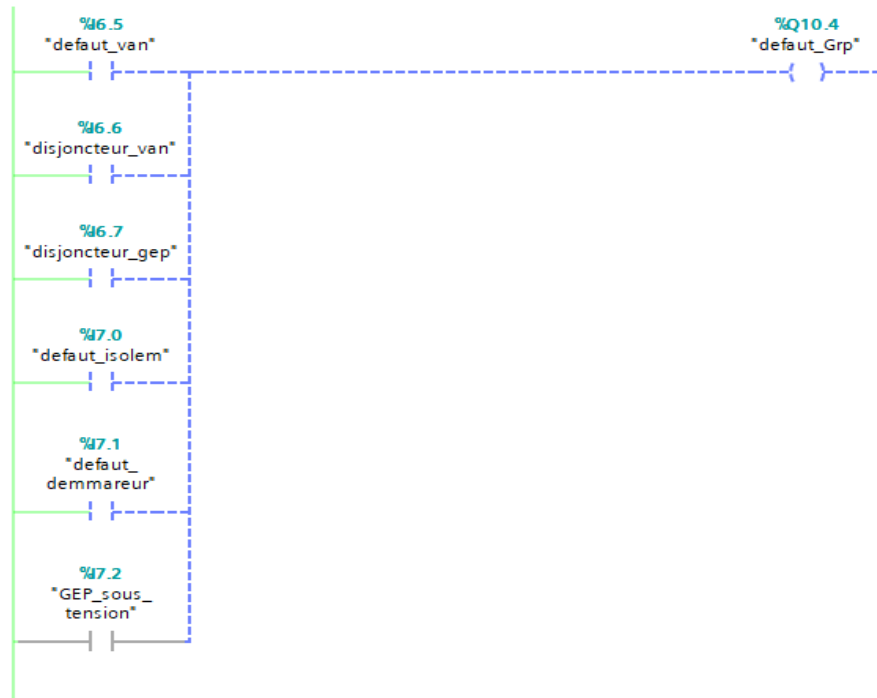
▼ Réseau 7 : .....

(\*Le défaut analogique final est la combinaison de tout les défauts analogiques\*)



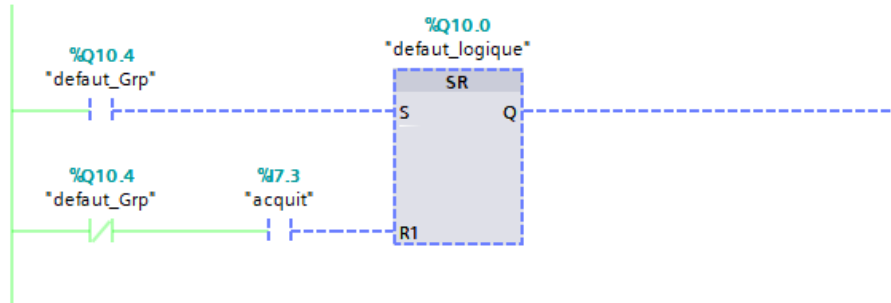
▼ Réseau 8 : .....

(\*le défaut logique est la combinaison de tout les défauts analogiques\*)



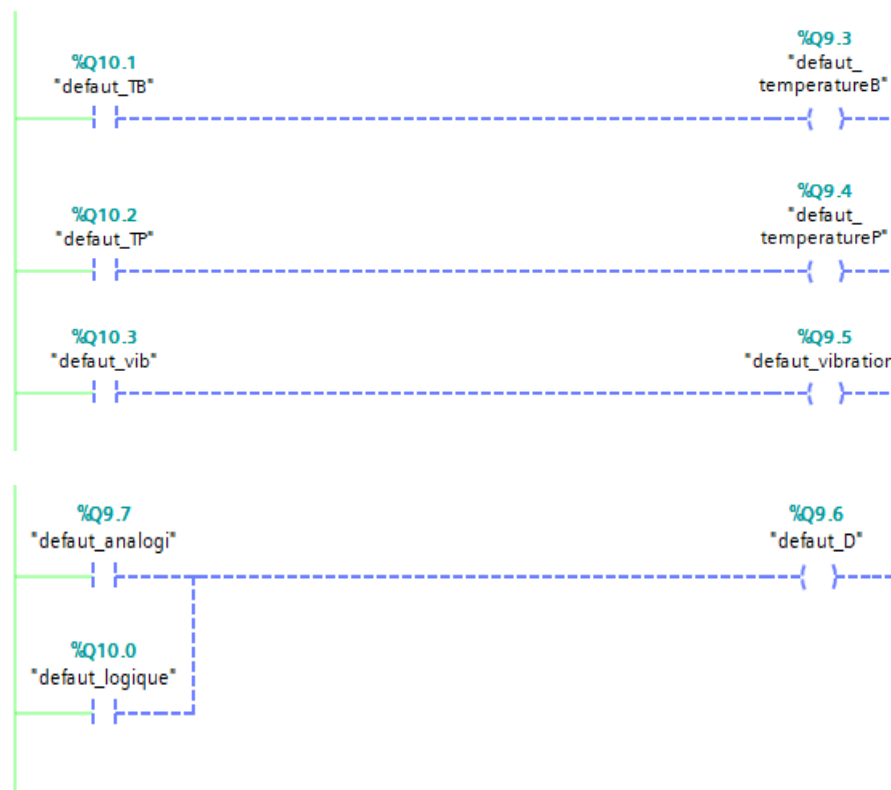
▼ Réseau 9 : .....

(\*le défaut disparaisse après appuie sur acquit (avec absence du défaut\_vib) \*)



▼ Réseau 10 : .....

(\* affectation des sorties\*)



## 4.6 Le Programme Groupe Electropompe

Le groupe électropompe est constitué de la pompe et son moteur, le GEP envoie deux signaux, un pour indiquer que son démarrage est terminé et un autre pour indiquer la fin de son arrêt .et sachant que la vanne envoie aussi 2 signaux (complètement ouverte et complètement fermée) ; le Programme de GEP assurera un bon fonctionnement de la pompe (un démarrage et un arrêt sécurisé).

### 4.6.1 Cahier de charge du GEP

Le Programme GEP est constitué de 9 entrées et de 9 sorties.

Les 9 entrées :

- marche : commande marche du GEP (signal envoyé par l'utilisateur ou un autre programme)
- arrêt : commande arrêt du GEP (envoyé par l'utilisateur ou un autre programme)
- FIN\_DEM : signal fin démarrage du GEP (envoyé par le GEP)
- FIN\_Arret : pour dire que le GEP est complètement arrêté (envoyé par le GEP)
- van\_fermee : fin de course vanne fermée
- van\_ouverte : fin de course vanne ouverte
- reset\_defaut : bouton reset défaut
- aut\_GEP : signal autorisation démarrage du GEP (envoyé par le Programme autorisation)
- defaut\_GEP : signal défaut (envoyé par le Programme défaut)

LES 9 sorties :

- ouvrir\_van : signal d'ouverture de la vanne
- fermer\_van : signal de fermeture de la vanne
- marche\_arret\_GEP : commande d'arrêt du GEP
- reset\_defaut\_GEP : commande de démarrage du GEP
- voyant\_marche\_GEP : voyant marche du GEP
- voyant\_defaut\_GEP : voyant défaut du GEP
- GEP\_en\_DEM : pour dire que le GEP est en phase de démarrage
- GEP\_en\_ARRET : pour dire que le GEP est en phase d'arrêt
- GEP\_en\_Marche : pour dire que le GEP est en fonctionnement

***Le Programme anti bélier doit assurer :***

- Un démarrage à vide et un arrêt à vide de la pompe :
  - Le démarrage à vide (sans charge) pour diminuer en max le couple de démarrage et assurer un démarrage rapide de la pompe, pour cela le démarrage se fait toujours avec vanne fermée. Après recevoir du signal FIN\_DEM on pourra ouvrir la vanne.
  - L'arrêt à vide (vanne fermée) pour éviter que la pompe absorbe le coup de bélier qui pourra la détruire mais aussi le reculement de l'eau (l'expérience a montré qu'un arrêt

avec une vanne ouverte provoque le reculement de l'eau à travers la pompe avec des grandes vitesses pouvant détruire complètement le GEP).

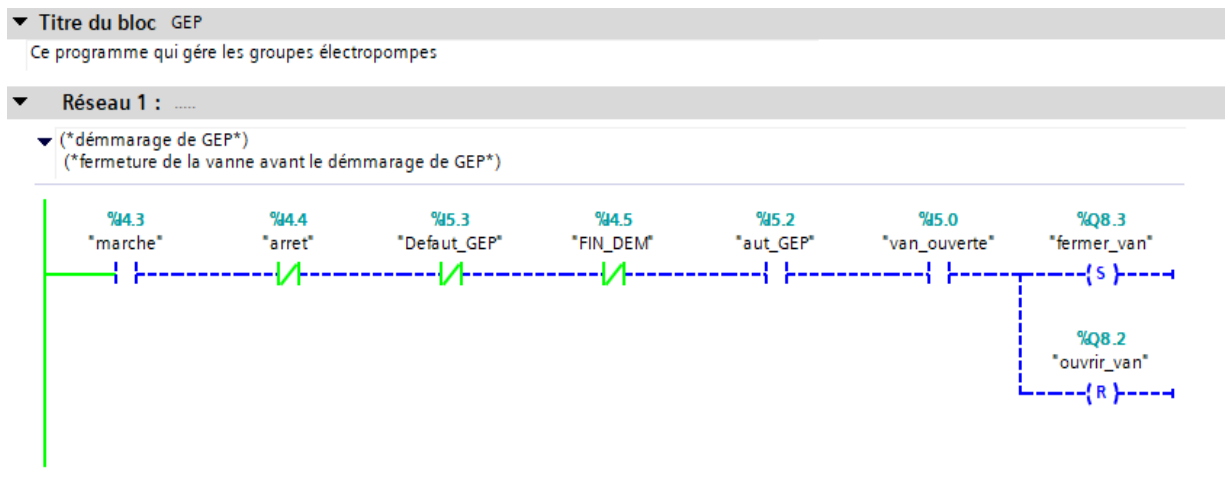
- Envoie de 3 signaux indiquant l'état du GEP : GEP\_en\_DEM, GEP\_en\_Arret, GEP\_en\_Marche.

<input type="checkbox"/>	marche	Bool	%I4.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	arret	Bool	%I4.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	FIN_DEM	Bool	%I4.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	FIN_Arret	Bool	%I4.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	van_fermee	Bool	%I4.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	van_ouverte	Bool	%I5.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	reset_defaut	Bool	%I5.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	aut_GEP	Bool	%I5.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	Defaut_GEP	Bool	%I5.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	ouvrir_van	Bool	%Q8.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	fermer_van	Bool	%Q8.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	marche_arret_GEP	Bool	%Q8.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	reset_defaut_GEP	Bool	%Q8.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	voyant_marche_GEP	Bool	%Q8.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	voyant_defaut_GEP	Bool	%Q8.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	GEP_en_DEM	Bool	%Q9.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	GEP_en_ARRET	Bool	%Q9.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	GEP_en_Marche	Bool	%Q9.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure 4-5 : Table variable du Programme « GEP »

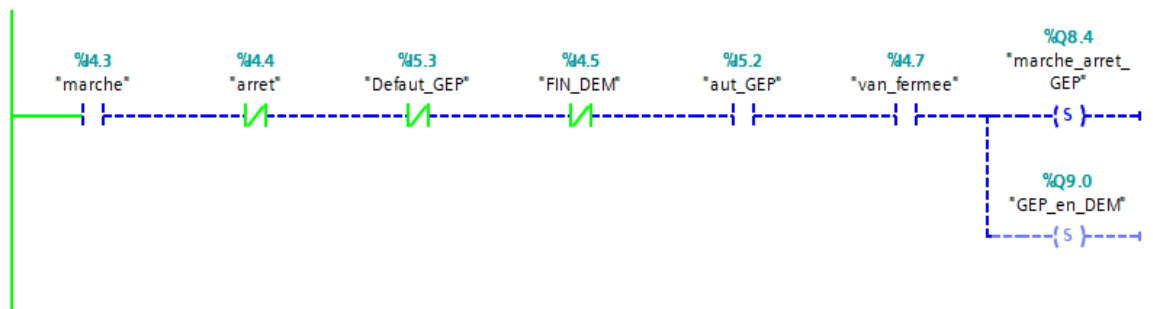
## 4.6.2 Le programme du GEP

La section du programme est écrite en langage LADDER (CONT)



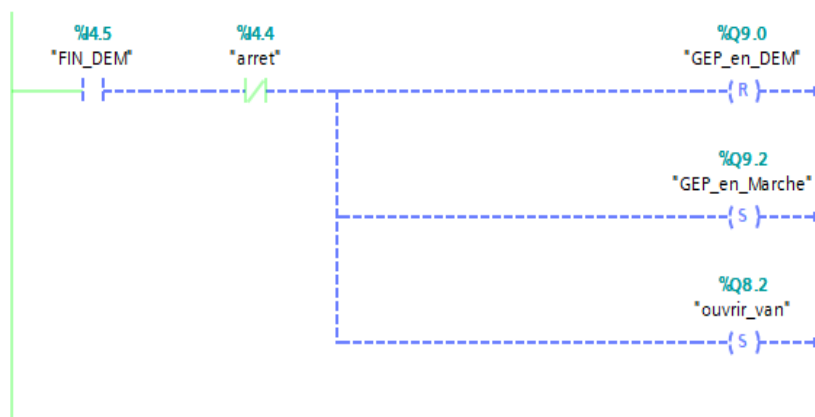
▼ Réseau 2 : .....

(\*si la vanne est fermée on démarre le GEP \*)



▼ Réseau 3 : .....

(\*ouverture de la vanne après le demarrage de GEP\*)



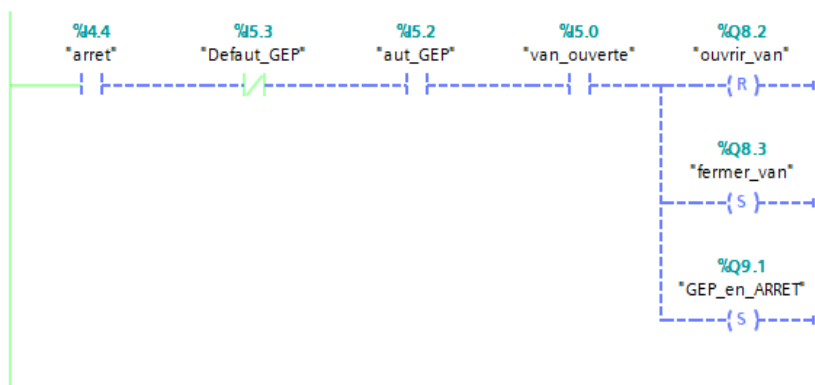
▼ Réseau 4 : .....

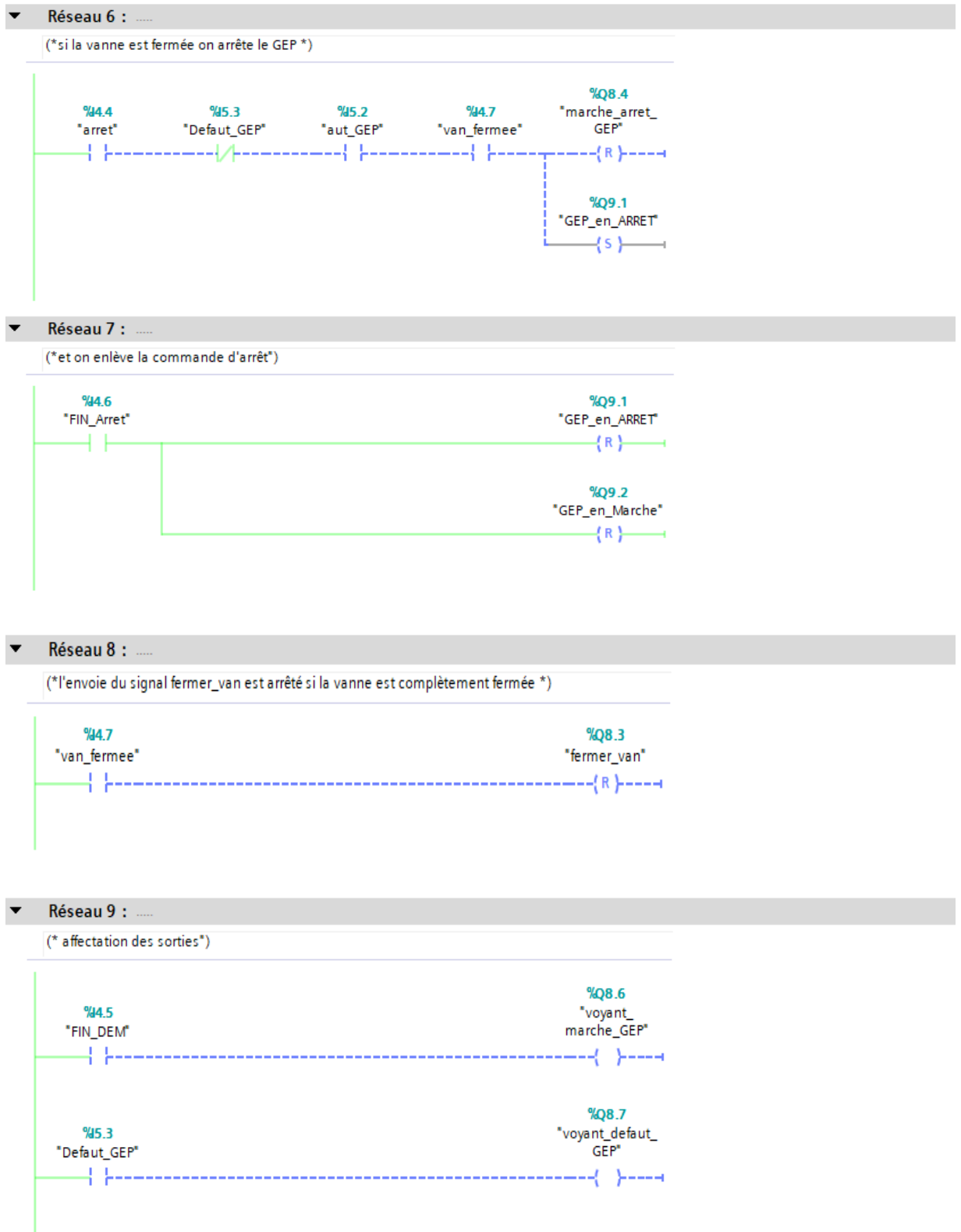
(\*l'envoi du signal ouvrir\_van est arrêté si la vanne est complètement ouverte \*)



▼ Réseau 5 : .....

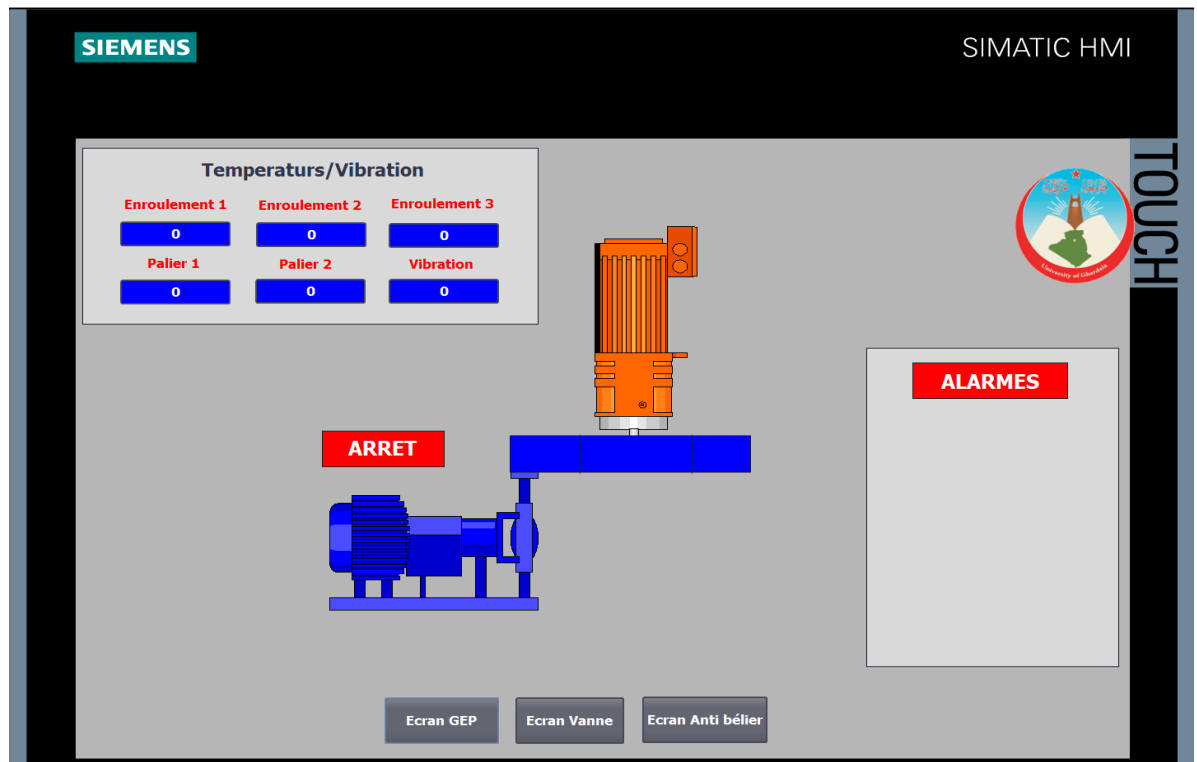
(\*fermeture de la vanne avant l'arrêt du GEP\*)





### 4.6.3 La simulation du Programme GEP

La figure ci-dessous illustre la simulation à l'aide d'un logiciel « WinCC IHM » intégrant dans « TIA PORTAL ».



## 4.7 Conclusion

Dans ce chapitre, les r sultats de notre projet sont pr sent s apr s avoir  t  programm s et simul s par le logiciel TIA PORTAL v16 d di    ce langage qui nous a permis de cr er un programme, d'afficher les entr es et sorties, et de simuler ce programme.



## Conclusion générale

Ce mémoire présente la gestion et la supervision d'une station pompage par API Siemens S7-1200 et logiciel Tia Portal v16.

Le rôle de l'ingénieur doit être l'optimisation des techniques permettant d'aboutir à l'objectif avec simplicité et performance.

Notre contribution s'est portée sur l'optimisation des programmes de la gestion de la station de pompage. Ces programmes sont présentés d'une façon simple et compréhensible. Les résultats et tests obtenus répondent bien aux cahiers des charges.

L'utilisation des interfaces homme-Machine (IHM) dans notre programme a pour objectif la visualisation de l'état du procès pendant l'exécution de la simulation et par la suite les intégrer réellement dans la pratique nous a montré la nécessité d'avoir un aspect artistique pouvant faire la différence dans l'étape de commercialisation du projet.

Ce projet a permis d'apprécier vraiment la nécessité d'avoir un aspect de communication avec d'autres spécialistes dans le domaine de notre application, que ce soit dans la mécanique des fluides, l'électrotechnique ou l'informatique industrielle une telle communication permettra à l'automaticien de bien comprendre son projet et par la suite pouvoir améliorer ces performances en faisant appel à ses propres connaissances.

# Bibliographies

- [1]. GONZAGA, A. *LE'S AUTOMATES PROGRAMMABLE*.  
[https://www.geea.org/IMG/pdf/LES\\_AUTOMATES\\_PROGRAMMABLES\\_INDUS\\_TRIELS\\_pour\\_GEEA.pdf](https://www.geea.org/IMG/pdf/LES_AUTOMATES_PROGRAMMABLES_INDUS_TRIELS_pour_GEEA.pdf).
- [2]. RAHMANI, A.e.G. and T. Nabil, *Automatisation d'une station de pendage d'eau filtrante*. 2015, Universités cameraman Ira - bannie.
- [3]. *Automatismes programmables et simulation*. 2021; V1.1: [http://lycees.ac-rouen.fr/maupassant/Melec/co/Communication/Syst\\_automat/co/10\\_Present\\_API.html](http://lycees.ac-rouen.fr/maupassant/Melec/co/Communication/Syst_automat/co/10_Present_API.html)
- [4]. A.Dali and A.M.Fihakhir, *Gestion Et Supervision D'une Station De Pompage à Base D'automate SCHNEIDER à L'aide Des DFB*. 2009, ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE.
- [5]. Youba, M.B. and M.B. Sofiane, *Commande et Contrôle d'un Compresseur d'Air Par Un Automate Programmable* 2012, Université ABDERAHMANE MIRA de Bejaia.
- [6]. automationsense. *Les automates Siemens pour les nuls*. 2016;  
<https://www.automation-sense.com/blog/automatisme/les-automates-siemens-pour-les-nuls.html>.
- [7]. Nantes, L.M.L.C. *Programmation de l'API SIMATIC S7-1200 avec TIA Portal*. 2013;  
<https://sti-monge.fr/maintenancesystemes/wp-content/uploads/2013/02/Initiation-1-TIA-Portal-MS1.pdf>.
- [8]. *SIMATIC S7 Automate programmable S7-1200* 2012;  
<https://www.competencesquebec.com/wp-content/uploads/2014/01/Automates-programmables.pdf>.
- [9]. *Automate S 7-1200 – Siemens*. 2019; <https://lab4sys.com/fr/automate-s7-1200-siemens/>.
- [10]. *Manuel système Automate programmable S 7-1200* 2014;  
[https://media.automation24.com/manual/fr/s71200\\_System\\_Manual\\_fr-FR\\_fr-FR.pdf](https://media.automation24.com/manual/fr/s71200_System_Manual_fr-FR_fr-FR.pdf).
- [11]. Siemens. *Support d'après-vente/de formation | Module 011-001 TINA Portal*. 2018;  
<https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/hw-config-s7-1200/sce-011-001-firmware-update-s7-1200-r1709-fr.pdf>.
- [12]. WAHIBA, O.L. and M. MARIEM, *Gestion de prototype « Industrial control work cell » par automate SIEMENS*. 2020, UNIVERSITE Ghardaïa.
- [13]. AG, S., «*SIMATIC STEP 7 dans le portail Totally Integrated Automation Portal*», . 2013.
- [14]. SIEMENS, *SIEMENS, manuel produit TIA PORTAL V13*. 2013.
- [15]. &, B.Y. and B.S.A. Mouhaimen, *Automatisation et télégestion d'une station de pompage*. 2019, Université SAAD DAHLAB de BLIDA.
- [16]. CTA. *Atelier C TIA Portal CTIA06 : programmation des automates*. 2016;  
<https://docplayer.fr/storage/18/841077/1621363949/kuuFx42BZ0QysUgTgZ1r1A/841077.pdf>.

- [17]. Yanis, M.I.B.e.M.S., *Étude et automatisation d'une station de pompage d'eau glacée*. 2017/2018, A/Mira de Bejaïa.
- [18]. *Présentation de quelques instructions usuelles (SIMATIC, marque de Siemens AG)*.  
[https://www.uvt.rnu.tn/resources-uvt/cours/Automates/chap3/co/Module\\_chap3\\_11.html](https://www.uvt.rnu.tn/resources-uvt/cours/Automates/chap3/co/Module_chap3_11.html).
- [19]. JOUVRAY, N., *Langages de programmation pour systèmes automatisés : norme CEI 61131-3*. 2008.
- [20]. *SIMATIC HMI WinCC flexible*  
2008[https://cache.industry.siemens.com/dl/files/010/18796010/att\\_99845/v1/Manuel\\_utilisateur\\_WinCC\\_flexible\\_fr-FR.pdf](https://cache.industry.siemens.com/dl/files/010/18796010/att_99845/v1/Manuel_utilisateur_WinCC_flexible_fr-FR.pdf).
- [21]. *SIMATIC HMI Pupitres opérateur Basic Panels 2012*  
[https://media.automation24.com/manual/fr/hmi\\_basic\\_panels\\_operating\\_instructions\\_fr-FR\\_fr-FR.pdf](https://media.automation24.com/manual/fr/hmi_basic_panels_operating_instructions_fr-FR_fr-FR.pdf).
- [22]. *Manuel : Démarreurs Progressifs Sirius 3rw30 / 3rw40*. 2010;  
[https://media.automation24.com/manual/fr/38752095\\_Manual\\_SIRIUS\\_softstarter.pdf](https://media.automation24.com/manual/fr/38752095_Manual_SIRIUS_softstarter.pdf).
- [23]. *Variateur de fréquence Siemens SINAMICS V20*.  
<https://www.automation24.fr/variateur-de-frequence-siemens-sinamics-v20-6sl3210-5be21-5cv0>.