



الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد  
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي  
Institut de Maintenance et de Sécurité Industrielle

**Département de Maintenance en Instrumentation**

## **MÉMOIRE**

Pour l'obtention du diplôme de Master

**Filière : Génie Industriel**

**Spécialité : Génie Industriel**

### **Thème**

**Variation de vitesse d'un moteur asynchrone  
(Conception et programmation d'un prototype  
d'une station de pompage à base d'API et un  
variateur de vitesse SIEMENS)**

Présenté et soutenu par :

**AMRANI Mohammed Sofiane**

Devant le jury composé de :

<b>Nom et Prénom</b>	<b>Grade</b>	<b>Etablissement</b>	<b>Qualité</b>
Mme RAHIEL	MCB	Université Oran 2	<b>Présidente</b>
Mr KACIMI Abderrahmane	MCB	Université Oran 2	<b>Encadreur</b>
Mr MEKKI Ibrahim EL Khalil	MCA	Université Oran 2	<b>Examineur</b>

**Année 2022/2023**

# DÉDICACES

---

**Avant tous, je remercie Dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.**

**Je dédie ce modeste travail :**

**A mes très chers parents, que Dieu les garde et les protège pour leurs soutien moral et financier, pour leurs encouragements et les sacrifices qu'ils ont endurés.**

**A mes collègues de travail .**

**A tous mes chers amis.**

**A tous mes amis d'études.**

**AMRANI Mohammed Sofiane**

# **Remerciements**

---

*Je tiens à remercier dieu de m'avoir donné la force morale, physique et l'aide pour accomplir ce modeste travail.*

*Je tiens à remercier mon promoteur **Pr.KACIMI** pour m'avoir acceptée encadrée et dirigée durant l'élaboration de ce travail ainsi que pour leur assistance et tous leurs conseils.*

*Je remercie chaleureusement les membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer mon projet.*

*Je souhaite aussi remercier tous les enseignants de l'institut de maintenance et sécurité industriels , et en particulier, Mes professeurs de GÉNIE INDUSTRIEL qui m'ont encadrée auparavant et tous mes enseignants pour les connaissances qu'ils m'ont transmis, leur disponibilité et leurs efforts.*

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



# ملخص

---

الهدف الرئيسي من هذا المشروع هو التحكم في سرعة المحرك غير المتزامن باستخدام إعدادات تجريبية ل PLC الذي تم تصميمه و تنفيذه لنموذج أولي لمحطة الضخ. في عملنا، أتاحت لنا الفرصة للتعامل مع SINAMICS G120C وهو نظام عاكس مدمج، ولهذا بدأنا في إنشاء التحكم HMI الخاص بنا باستخدام WINCC Runtime وتفعيل تنظيم PID الخاص بنا من خلال محاكاة البرنامج على PLCSIM. للتفاعل بين IHM الخاص بنا ووحدة التحكم في السرعة SINAMICS G120C استخدمنا SIMATIC S7-1500 PLC كوحدة تحكم، باستخدام برنامج البرمجة TIA Portal V16.

## Resumé

---

L'objectif principal de ce projet est le contrôle de la vitesse du moteur asynchrone à l'aide d'une configuration expérimentale PLC qui a été conçu et mis en œuvre pour un prototype d'une station de pompage.

Dans notre travail on a eu la chance a manipulé SINAMICS G120C qui est un système d'onduleur compacte, pour cela on a commencé de créer notre propre IHM de commande en utilisant WINCC Runtime et activer notre régulation PID en simulant le programme sur PLCSIM.

Pour interagir entre notre IHM et le variateur de vitesse siemens G120C on a utilisés un API SIMATIC S7-1500 comme contrôleur, grâce au logiciel de programmation TIA Portal V16.

Mots clés : API, Commande IHM, SINAMICS G120C, PID Compact.

## Abstract

---

The main objective of this project is the speed control of the asynchronous motor using an experimental PLC setup which was designed and implemented for a prototype of a pumping station.

In our work we had the chance to handle SINAMICS G120C which is a compact inverter system, for this we started to create our own control HMI using WINCC Runtime and activate our PID regulation by simulating the program on PLCSIM.

To interact between our HMI and the Siemens G120C speed controller we used a SIMATIC S7-1500 PLC as a controller, using the TIA Portal V16 programming software.

Keywords: PLC, HMI control, SINAMICS G120C, PID Compact.



# Sommaire

---

Introduction Générale.....	1
<i>Chapitre I Les variateurs de vitesse industriels</i>	
Introduction .....	2
I.2. Historique .....	2
I.3. Définition: .....	2
I.4. Constitution : .....	3
I.5. Quadrants de fonctionnement des variateurs .....	4
I.6. Les principales fonctions des variateurs de vitesse électroniques .....	5
I.7. Principe de fonctionnement du variateur de fréquence .....	5
I.8. Convertisseur de fréquence pour moteur asynchrone.....	6
I.9. Commande du moteur asynchrone .....	6
I.9.1. Commande scalaire.....	6
I.9.2. Commande vectorielle .....	8
I.10.1. Variateur vitesse marque VFD .....	9
I.10.2. Variateur vitesse ACS5500 de ABB .....	10
I.10.3. Etude d'un cas de variateur industriel standard : SINAMICS G120C de marque Siemens : cas de notre étude .....	10
CONCLUSION :.....	12
<i>Chapitre II Etude détaillée du système de variation de vitesse considéré</i>	
Introduction.....	13
II.2. Aperçu détaillé sur la commande du Moteur .....	13
II.2.1. Commande U/f.....	13
II.2.2. Régulation vectorielle .....	14
II.3.1. Présentation variateur vitesse G 120C.....	16
II.3.2. Description de variateur vitesse G120C.....	17
II.3.3. Composant du module de puissance .....	17
II.3.4. Panneau de commande intelligent (IOP) .....	18

II.4. Généralités sur les automates programmables.....	19
II.4.1. Architecture des automates programmables industriels.....	20
II.5. Choix d'un automate programmable .....	23
II.5.1.CPU 1512C-1PN utilisé.....	23
II.5.3. Objet technologique PID Compact .....	23
II.5.4. Paramètres PID .....	24
II.5.5. Valeur de réglage.....	24
II.6. Système de régulation de débit : .....	25
II.7. Description du système de régulation du débit (Variateur-pompe-débit) .....	26
II.7.1. Les caractéristiques de la pompe Taifu QB50 :.....	26
II.7.2. Débitmètre à turbine : .....	27
II.8. Communication entre l'API et control unit de variateur de vitesse G120C :.....	28
II.8.1. Définition de PROFINET.....	28
II.8.2. Définition Ethernet .....	29
II.8.3. Télégramme utilisé pour le mode de régulation de vitesse.....	30
II.9. Présentation des différents logiciels et langage.....	31
II.9.1. Description du logiciel TIA (Totally Integrated Automation) portal....	31
II.9.2. STEP 7 sur TIA portal :.....	31
II.9.3. Win CC sur TIA portal :.....	31
II.9.4. PLCSIM.....	32
II.10. Blocs de programme.....	32
Conclusion.....	34
<i>Chapitre III Conception et programmation de notre prototype</i>	
Introduction : .....	35
III.2. Eléments de commande, de gestion et de supervision de la Maquette de la station de pompage .....	35
III.3. Développement du projet pour gestion du prototype de la station de pompage .....	37
III.4. Création du projet.....	38

III.4.1. Paramétrage général de l'interface PG/PC .....	38
III.4.2. Procédure de création du projet.....	39
III.5. Ajout d'API.....	39
III.6. Ajoute du support d'IHM .....	41
III.7. Ajout du variateur de vitesse et ces composants .....	42
III.8. Vérification des adresses IP des différents appareils .....	44
III.9. Création d'IHM de commande du système.....	44
III.9.1. Présentation de l'IHM .....	44
III.9.2. Procédure de Création d'IHM de commande du système .....	45
III.10. Principe de Régulation de débit (PID) à partir d'un variateur de vitesse G10C.....	48
III.11. Développement de notre programme au niveau de Step.....	49
III.11.1. Tables de mnémoniques.....	49
III.11.2. Architecture de notre programme .....	49
III.11.3. Bloc d'organisation OB1 (programme Main).....	50
III.11.4. Bloc d'organisation OB30 (Cyclic interrupt ).....	51
III.11.4. Bloc d'organisation OB100(Startup) .....	53
III.11.5. Présentation des fonctions FC .....	54
III.11.6. Fonction : Control VFD G120C [FC1] .....	54
III.11.7. Fonction : Les électrovannes [FC2] .....	56
III.11.8. Blocs de données DB (Data Bloc) .....	56
III.11.9. Blocs de données DB3 (Data Global).....	56
III.11.10. Blocs de données DB1 (PID Data ) .....	57
III.12. Commande du variateur à travers le Télégramme.....	57
III.13. Enregistrer et compiler le programme.....	59
III.14. Charger le programme.....	60
III.15. Visualiser des blocs de programme .....	61
III.16. Démarrage de simulation.....	61
III.16.1. Vue compacte.....	62
III.16.2. Vue du projet.....	62

III.17.Reglage des paramètre PID.....	63
III.18. Analyse des résultats.....	66
Conclusion:.....	66
Conclusion Générale .....	67
Reference bibliographique et Références web graphiques .....	69
Annex.....	71

# Liste des figures

---

Figure I-1 Schéma de principe d'un variateur de vitesse.....	2
Figure I-2 : Structure générale d'un variateur de vitesse électronique [1].....	3
Figure I-3 Module de puissance d'un variateur de vitesse.....	3
Figure I-4 :Quadrants de fonctionnement des variateurs.....	4
Figure I-5: Schéma électrique d'un variateur de vitesse [8].....	6
Figure I-6: Présentation de la commande en boucle ouverte.[18] .....	7
FigureI-7 Commande scalaire en boucle fermée de la machine [8].....	8
Figure I-8 Schéma de principe de découplage de la MAS/MAC [9].....	8
Figure I-9 Variateur vitesse VFD.....	9
Figure I-10 Variateur vitesse ACS550.....	10
Figure I-11 : Variateurs de vitesse G120C.....	11
Figure II-1 : Diagramme fonctionnel simplifié de la commande U/f [13].....	13
Figure II-2 : par défaut de la commande U/f après sélection de Standard Drive Control [13].....	14
Figure II-3 : Diagramme fonctionnel simplifié pour la régulation vectorielle avec régulateur de vitesse [3].....	15
Figure II-4: Filtre de réseau PM240 [15].....	18
Figure II-5 Inductance de réseau PM240 [15].....	18
Figure II-6 : Inductance de sortie PM240[15].....	18
Figure II-7: Représente Panneau de commande intelligent (IOP) [16].....	19
Figure II-8: Automate programmable industriel. [17].....	19
Figure II-9: Automate programmable Siemens S7-1500, CPU 1512C-1PN.....	20
Figure II-10: Structure interne d'un API [18].....	21
Figure II-11 :Module d'alimentation d'un API.....	21
Figure II-12 : Module CPU d'un API.....	22
Figure II-13 : Interfaces d'entrée/ sortie.....	22
Figure II-14 : S7-1500 1512C-1PN.....	23
Figure II-15: schéma fonctionnel de PID Compact [20].....	25
Figure II-16: Courbe des hauteurs manométriques en fonction de débit de pompe Taifu QB35.....	27
Figure II-17 : Débitmètre à turbine.....	27

Figure II-18 : Constitution d'un débitmètre à turbine[21].....	28
Figure II-19: Exemples de connecteurs PROFINET [23].....	29
Figure II-20: Schéma représentant la connexion entre CU et S7 1500 sous-réseau PN/IE.....	29
Figure II-21: Création d'un nouveau bloc.....	33
Figure III-1 Vue de la maquette de station de pompage.....	36
Figure III-2 : Synoptique du prototype.....	36
Figure III-3 : Description du Schéma Electrique de la station.....	37
Figure III-4: Paramétrage de l'interface PG/PC.....	39
Figure III-5: Création du projet.....	39
Figure III-6: Procédure d'ajout du contrôleur.....	40
Figure III-7: Adresse IP de l'API.....	41
Figure III-8: Choix du system PC et de WinCC Runtime.....	41
Figure III-9: Dépôt du module de communication.....	42
Figure III-10: Configuration du protocole IP du module de communication.....	42
Figure III-11 :Choix du variateur.....	43
Figure III-12: Configuration du protocole IP de variateur vitesse.....	44
Figure III-13: Vérification des adresses IP.....	44
Figure III-14: Méthode d'ajout d'une vue.....	45
Figure III-15 : Représentation de la fenêtre d'IHM.....	46
Figure III-16 : Configuration d'évènement d'un bouton.....	46
Figure III-17 :Vue IHM de station de pompage.....	47
Figure III-18 :vue IHM de la boucle de regulation .....	47
Figure III-19 :vue IHM de la courbe de regulation PID.....	48
Figure III-20 :Vue IHM des parametres PID.....	48
Figure III-21 : Table des variables.....	49
Figure III-22 : Architecture de notre programme.....	50
Figure III-23 : Les réseaux de Bloc d'organisation principale OB1.....	51
Figure III-24 :Shema fonctionel du PID Compact.....	52
Figure III-25 : Le bloc de PID Compact.....	52
Figure III-26 :Réseau 02 du OB 30 qui represente la valeur d'Erreur de la boucle.....	53
Figure III-27 :Réseau 03 du cyclic interrupt avec des blocs MOVE pour transferer les variable en IHM.....	53

Figure III-28 :Réseau 04 qui represente l'action d'arrêt du moteur quand le debit est moins de 5 L/MIN .....	53
Figure III-29 :Le reseau de bloc OB100 pour rendre le PID par default.....	54
Figure III-30 :Réseau 01 pour le demmarage et l'arrêt du moteur avec le bloc SR (SET/RESET).....	55
Figure III-31 : Réseau 02 pour inverser le sens de rotation du moteur.....	55
Figure III-32 : Réseau 03 Pour acquitter les defauts au variateur.....	55
Figure III-33 : Réseau 04 d'equation de signal de sortie en langage SCL.....	55
Figure III-34 :Les fonctions de programmation des electrovanes manuelles.....	56
Figure III-35 :La base de doonnées de notre programme.....	57
Figure III-36 : Les blocs de données du PID.....	57
Figure III-37 :Capture d'écran de la table de mnémoniques comportant les variables du télégramme-1.....	59
Figure III-38 : Activation/Désactivation des bits du télégramme 1 pour la commande du variateur.....	59
Figure III-39 :Les étapes de compilation.....	60
Figure III-40 : Signalisation des erreurs de compilation.....	60
Figure III-41 :Le chargement du programme.....	61
Figure III-42 :Le bloc PID Compact apres la visualisation.....	61
Figure III-43 :Fenêtre de la vue compacte.....	62
Figure III-44 :Fenetre de la vue du projet.....	63
Figure III-45 : Méthode Ziegler Nichols.....	64

# Liste des tableaux

---

<b>Tableau II-1</b> :Les paramètres de l'équation de régulation PID.....	24
<b>Tableau II.2</b> : Caractéristique des pompes Taifu QB.....	26
<b>Tableau II-3</b> :Rapport Régime/Débit de la pompe taifu QB50.....	27
<b>Tableau III.1</b> : les étapes du développement du projet.....	38
<b>Tableau III-2</b> : Représentation des bits du télégramme commandant notre variateur.....	40

# Symboles et abréviation

AC/CA	Courant alternatif
DC/CC	Courant continue
V <sub>c</sub>	Tension charge
V <sub>s</sub>	Tension source
U <sub>v</sub>	Tension vide
V <sub>r</sub>	Tension référence
I <sub>a</sub>	Courant induit « Rotor »
I <sub>d</sub>	Courant inducteur « stator »
I <sub>sq</sub>	Courant inducteur « stator »
GTO	Gate-Turn-Off Thyristor
IGBT	Insulated Gate Bipolar
PAM	Pulse Amplitude Modulation
PWM	Pulse width Modulation
SPWMS	Sinusoidal Pulse Width Modulation
f.em	force électromotrice
W <sub>s</sub>	Pulsation statorique
W <sub>r</sub>	pulsation rotorique
W <sub>ref</sub>	pulsation référence
C <sub>em</sub>	Couple électromagnétique
C <sub>r</sub>	Couple résistance
L <sub>r</sub>	Inductance rotorique
L <sub>s</sub>	Inductance statorique
R	Resistance de charge
R <sub>s</sub>	Résistance statorique
R <sub>r</sub>	Résistance rotorique
<i>p</i>	Nombre paire de pole
N	Nombre de spire
$\varphi_s$	Flux statorique
P	Correcteur proportionnel
I	Correcteur intégrateur.
PID	Correcteur proportionnel, intégrateur et dérivateur
MAS	Moteur asynchrone
MAC	Moteur courant alternatif
CU	Control Unit
PM	Power Module
HMI	Human Machine Interface
OB	Organisation Bloc
PZD	Données process
PKW	Paramètre -Kennung-Wert
ZSW	Mot d'état
API	Automate Programmable Industriel

# Introduction Générale

Un API est un appareil intégré avec prise en charge de la programmation qui définit le traitement des données, le comptage, la base de données, et attribue des entrées à la sortie après une ou plusieurs opérations logiques.

De nos jours, on trouve les automates programmables industriels qui répondent aujourd'hui à toutes les exigences de l'industrie, Par rapport à d'autres méthodes de contrôle, l'utilisation d'API nous permet de résoudre des problèmes complexes pour tous les secteurs d'activité. En effet, il est plus fiable, plus résistant, il est moins endommagé, plus facile à entretenir et présente de nombreux avantages, car c'est pourquoi il est souvent préféré dans l'industrie.

L'objectif principal de notre projet de fin d'étude est la conception et la simulation d'une maquette d'une station de pompage, avec une régulation de débit par variation de vitesse d'un moteur asynchrone via un variateur de vitesse et la commander par un contrôleur API S7 1500 Siemens tout en assurant le bon fonctionnement du processus.

Pour atteindre cet objectif, nous sommes passés de la programmation en langage "LADDER" et la supervision "système-PC " sous PROFINET avec API SIEMENS S7- 1500 à la simulation avec logiciel PLCSIM .

Notre modeste travail comporte trois chapitres et une conclusion générale.

Dans le premier chapitre, on a donné un aspect général des différentes topologies du circuit de puissance des variateurs de vitesse et du circuit de commande, ainsi qu'un aperçu historique, et quelques exemples des différents variateurs

Dans le deuxième chapitre, nous avons donné un aperçu des communications entre API et unités control du variateur de vitesses G120C ainsi que différents logiciels qui on utilisé utilisons, et des différentes structures qu'un API peut contenir.

Le dernier chapitre sera consacré à présenter le programme, la simulation et la supervision de notre système par contrôleur API S7 1500.



# **Chapitre I**

## **Les variateurs de vitesse industriels**

**I.1. Introduction**

Dans ce chapitre, on présente une étude sur les variateurs de vitesse électroniques dans laquelle on explique tout d’abord le principe de fonctionnement et les différentes structures puis les composants de variateur électronique de puissance.

Ensuite, on décrit la gamme des variateurs de fréquences SINAMICS G120C de Siemens.

**I.2. Historique [1]**

Historiquement, le variateur électronique pour moteur à courant continu a été la première solution offerte. Les progrès de l’électronique de puissance et de la microélectronique ont permis la réalisation de convertisseurs de fréquence fiables et économiques. Les convertisseurs de fréquence modernes permettent l’alimentation de moteurs asynchrones standards avec des performances analogues aux meilleurs variateurs de vitesse à courant continu. Certains constructeurs proposent même des moteurs asynchrones avec des variateurs de vitesse électroniques incorporés dans une boîte à bornes adaptée ; cette solution est proposée pour des ensembles de puissance réduite (quelques kW).

**I.3. Définition: [2]**

Un variateur de vitesse est un convertisseur d’énergie permettant de moduler l’énergie électrique fournie au moteur. Les variateurs de vitesse sont constitués principalement d’un convertisseur statique et d’une électronique de commande. Les variateurs récents contiennent aussi un étage de correction du facteur de puissance afin de respecter les normes de compatibilité électromagnétique.

L’électronique de commande réalise la régulation et l’asservissement de la machine à travers le convertisseur statique de sorte que l’utilisateur puisse commander directement une vitesse. Sa conception dépend essentiellement de la stratégie de commande choisie (commande vectorielle, commande scalaire, etc.).

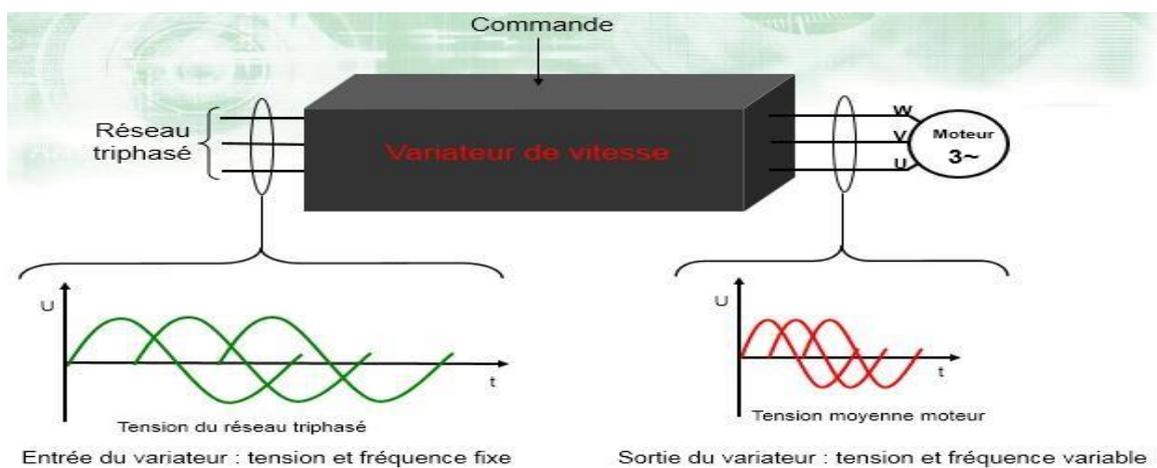


FIGURE I-1 : Schéma de principe d’un variateur de vitesse.

I.4. Constitution :

Le variateur de vitesse est constitué d'un module de contrôle et de puissance.

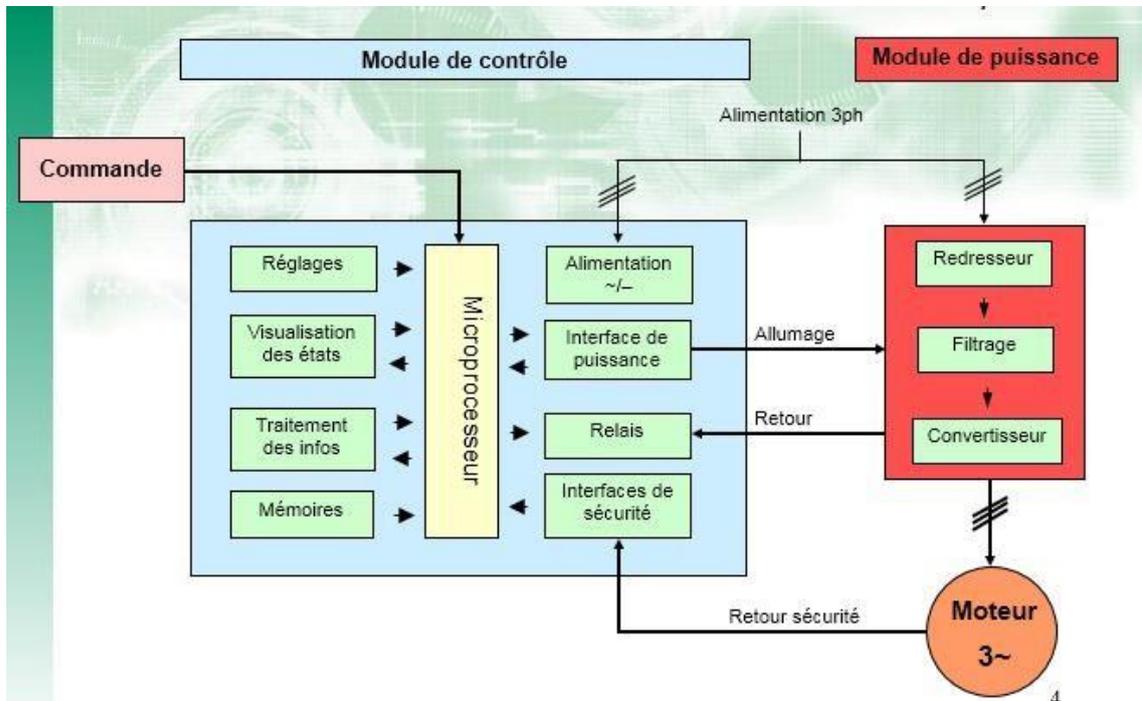


FIGURE I-2 : Structure générale d'un variateur de vitesse électronique [1].

- Le module de puissance est constitué en 3 parties :
  - Le redresseur composé de diodes de redressement
  - Le Filtrage composé de condensateurs
  - L'onduleur composé de transistors bipolaires de puissance (IGBT)

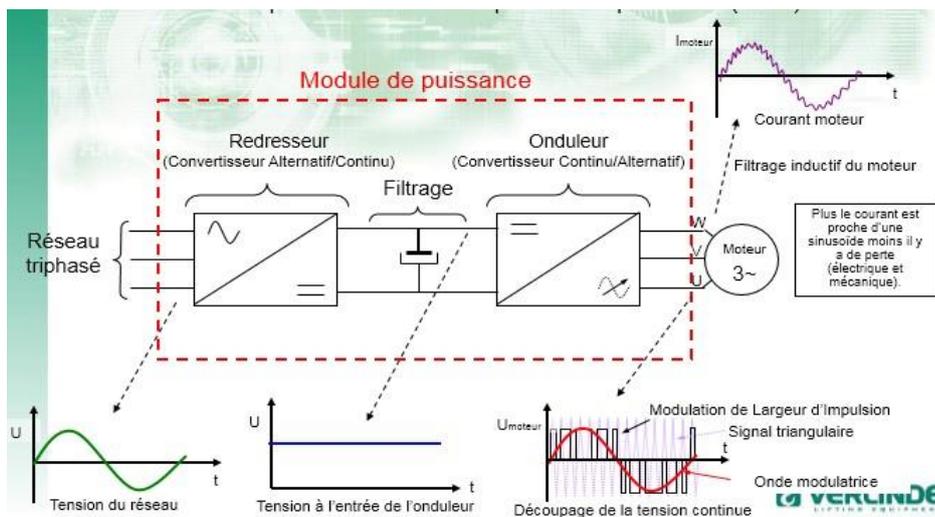


Figure I-3 Module de puissance d'un variateur de vitesse

- L'unité de traitement du signal permet de varier et réguler la vitesse :

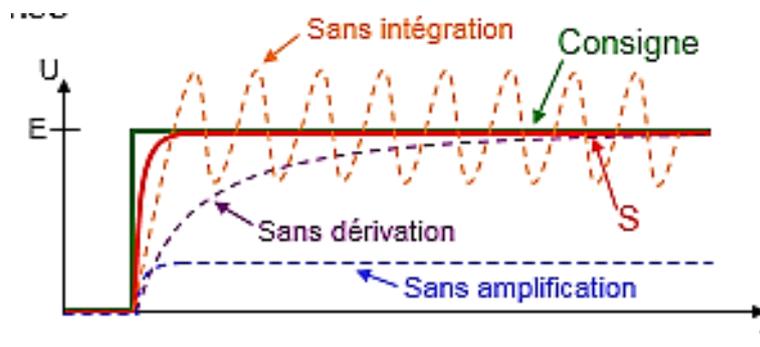
Basés sur des circuits de hautes technologies tels que des ASIC (circuits numériques programmables) associés à un microprocesseur permettant un contrôle précis.

- Contrôle en fréquence, en tension, en courant
- Contrôle en vitesse, en couple, en flux
- La régulation de vitesse est réalisée par correction du type PID toutes les 1 ms

P : proportionnel → amplification du signal

I : Intégrale → stabilisation du système

D : Dérivée → réduction du temps de réponse



### I.5. Quadrants de fonctionnement des variateurs [3]

Deux paramètres définissent le fonctionnement des systèmes donc des moteurs. Ces deux paramètres sont le couple et la vitesse. Le couple dépend de la charge qui peut être entraînée ou entraînant et le signe de la vitesse dépend du sens de rotation du moteur.

4 quadrants définissent les zones de fonctionnement :

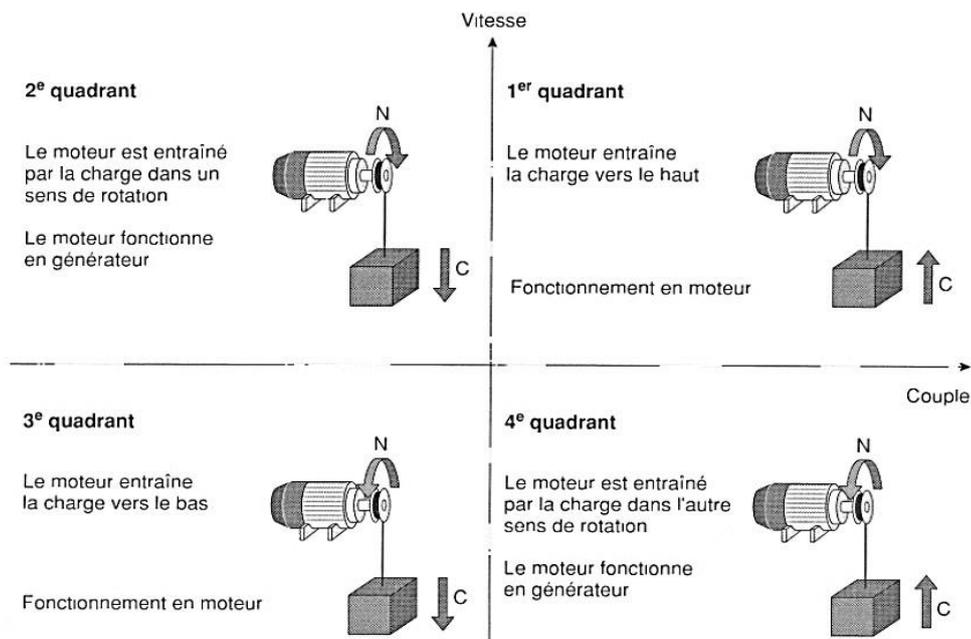


FIGURE I-4 : Quadrants de fonctionnement des variateurs.

## I.6. Les principales fonctions des variateurs de vitesse électroniques [4]

Les variateurs de vitesses sont des systèmes qui convertissent les caractéristiques d'une alimentation en fonction d'une consigne donnée. Ils ont plusieurs fonctions parmi lesquelles :

- Le **démarrage** : le moteur passe de la vitesse nulle jusqu'à sa vitesse établie en un temps prédéfini et en évitant les pointes d'intensité.
- La **variation de vitesse** : modification de la fréquence de rotation du moteur par accélération ou décélération en un temps donné.
- La **régulation** : la fréquence de rotation du moteur est maintenue constante quelles que soient les fluctuations de la charge (dans certaines limites).
- Le **freinage** : le moteur passe d'une vitesse établie à une vitesse inférieure (ralentissement) ou à la vitesse nulle (arrêt) avec maintien en position possible.
- L'**inversion du sens de rotation** : permet de faire fonctionner le moteur dans les deux sens de rotation.
- La **recupération d'énergie** : permet lors d'un ralentissement ou d'un freinage des systèmes de transformer l'énergie mécanique en énergie électrique. Dans ce cas, le moteur fonctionne en génératrice et l'énergie récupérée peut être soit dissipée dans des résistances, soit utilisée pour recharger des batteries ou encore réinjectée dans le réseau.

## I.7. Principe de fonctionnement du variateur de fréquence [5]

Le convertisseur de fréquence utilise le principe de la variation de la fréquence des sinusoïdes et de la tension afin de maintenir un couple constant à toutes les fréquences. Cette technique nous permet de varier la vitesse de nos machines de manière régulière, sans à-coup, y compris à basse vitesse. La forme du courant de sortie est proche de la sinusoïdale.

### I.7.1. Description de fonctionnement

La tension alternative triphasée est convertie en tension continue par l'intermédiaire du pont redresseur et des condensateurs de filtrage. Cette tension continue est alors découpée par un pont onduleur à transistors. L'ajustage de la largeur des impulsions et leur répétition permet d'ajuster l'alimentation du moteur en tension et en fréquence afin de garantir un rapport tension/fréquence constant dans le moteur.

Un signal externe de vitesse est transmis au bloc de commande ASIC (action par microprocesseurs). Le bloc de commande, après comparaison des signaux internes et externes,

actionne la commande des transistors.

Le dialogue avec l'opérateur est facilité par la console LCD du variateur ou par la connexion d'un PC.

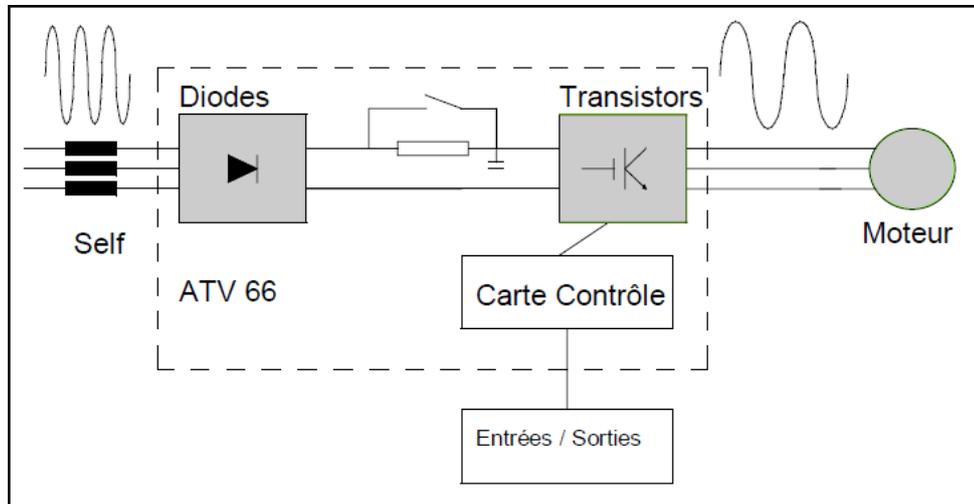


FIGURE I-5 Schéma électrique d'un variateur de vitesse [8]

## I.8. Convertisseur de fréquence pour moteur asynchrone [6]

Les variateurs de type «convertisseurs de fréquence» sont utilisés pour alimenter les moteurs asynchrones. Ils permettent essentiellement de faire varier la vitesse de rotation de ces moteurs, mais aussi d'obtenir des caractéristiques de fonctionnement du moteur radicalement différentes de celles obtenues lors de l'utilisation normale à amplitude et fréquence constantes (moteurs alimentés en direct par la tension du réseau). La vitesse de rotation du moteur varie en fonction de la fréquence de la tension d'alimentation. Les variateurs utilisent cette caractéristique pour obtenir une régulation en vitesse. L'évolution de l'électronique de puissance et de la vitesse des processeurs a permis de développer de très bons systèmes de contrôle des caractéristiques de fonctionnement.

## I.9. Commande du moteur asynchrone

### I.9.1. Commande scalaire

Les stratégies d'une commande scalaire d'une MAS sont basées sur la commande simple. Des grandeurs variables de la machine comme la tension, le courant et la fréquence sans considération pour la phase. Bien que ces stratégies aient l'avantage d'être facilement réalisables, elles ont l'inconvénient d'offrir de faibles performances dynamiques. La stratégie de commande scalaire la plus connue est celle où lorsqu'on varie la fréquence  $f_s$ , on ajuste la tension d'alimentation  $V_s$  au stator de telle manière que le flux ou le rapport entre la tension et la fréquence reste constant. Cette stratégie de commande peut être appliquée en boucle ouverte ou en boucle fermée avec régulation de la fréquence de glissement.

**Principe de la commande scalaire**

Dans la machine asynchrone les champs stator et rotor restent synchronisés du fait que, par induction, la fréquence des courants rotoriques varie automatiquement avec la vitesse du rotor pour compenser cette variation.

Le champ rotorique est mobile par rapport au stator à pulsation  $\omega_{sl}$ , contrairement au cas des machines synchrones où la fréquence du courant d'excitation reste constante et nulle quelle que soit la vitesse du rotor et où, par conséquent, le champ rotorique est fixe par rapport à celui-ci.

Dans les machines asynchrones il n'y a donc pas de risque de décrochage par couple moyen nul mais seulement de blocage par couple résistant excessif, supérieur au couple maximal, qui provoque un arrêt et un fonctionnement en court-circuit puisqu'il n'y a plus de f.e.m. de rotation pour s'opposer à la tension statorique.

Les machines asynchrones présentent des instabilités en régime transitoire. C'est le cas lors d'une alimentation directe en tension lorsque le filtre d'entrée présente des valeurs d'inductance et de capacité élevées. C'est aussi le cas lors d'une alimentation directe en courant avec fréquence statorique imposée.

**Principe de la commande en boucle ouverte**

La commande en boucle ouverte par onduleur du moteur asynchrone s'effectue par la pulsation  $\omega_s$  et la valeur efficace de la tension  $V_s$  phase- neutre du stator pour un onduleur de tension.[7]

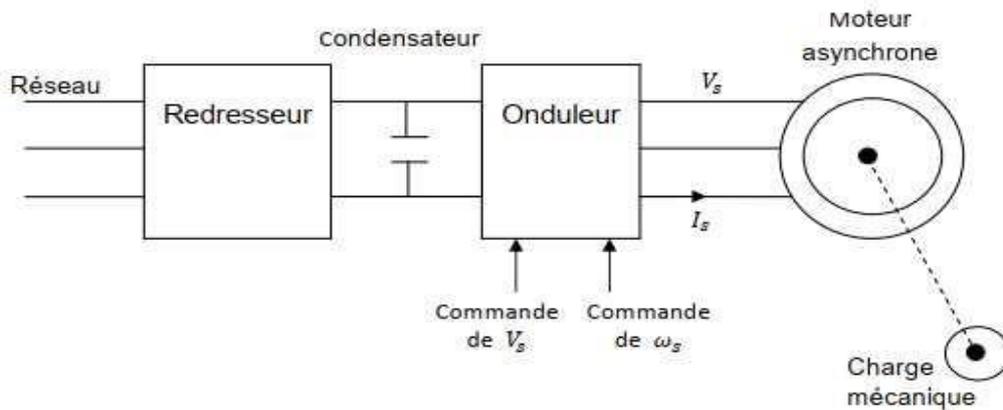


Figure I-6: Présentation de la commande en boucle ouverte.[18]

**Commande scalaire en boucle fermée**

Si on veut réaliser un asservissement, on rajoute une boucle externe qui, à partir de l'erreur de vitesse, permet d'augmenter la fréquence des tensions statoriques de façon à annuler l'erreur de vitesse due au glissement. La structure d'un tel montage est donnée à la figure ci-dessous.

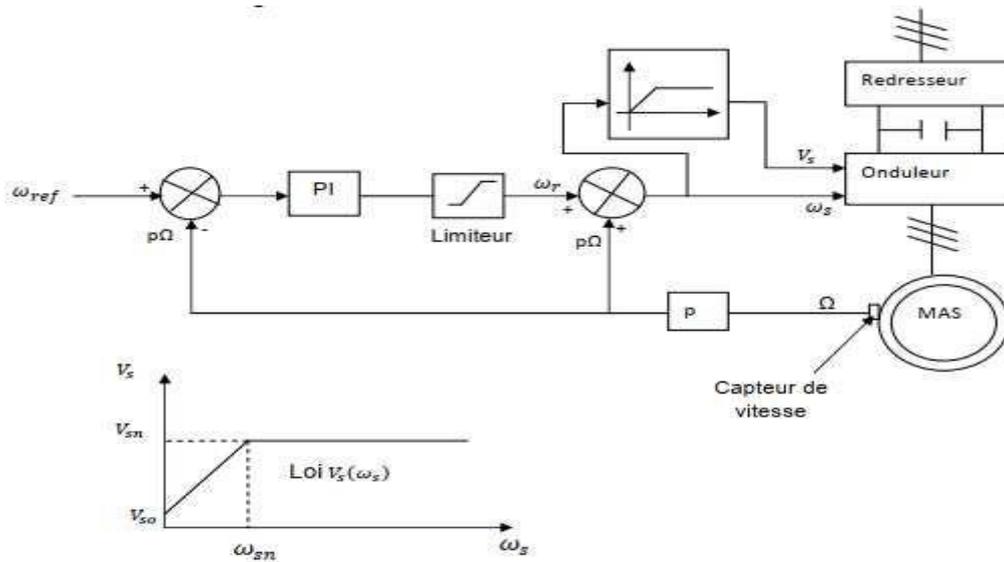


Figure I-7 Commande scalaire en boucle fermée de la machine [8]

### 1.9.2. Commande vectorielle

#### Principe de la commande vectorielle

Le principe de cette commande est de réduire l'équation de couple électromagnétique de la machine afin d'être comparable à celle d'une machine à courant continu.

Le référentiel (d, q) lié au champ tournant est choisi de telle façon que l'axe (d) coïncide avec la direction du flux ( $I_{sd}$ ) et que l'axe (q) coïncide avec la direction de la composante ( $I_{sq}$ ) donnant le couple électromagnétique.

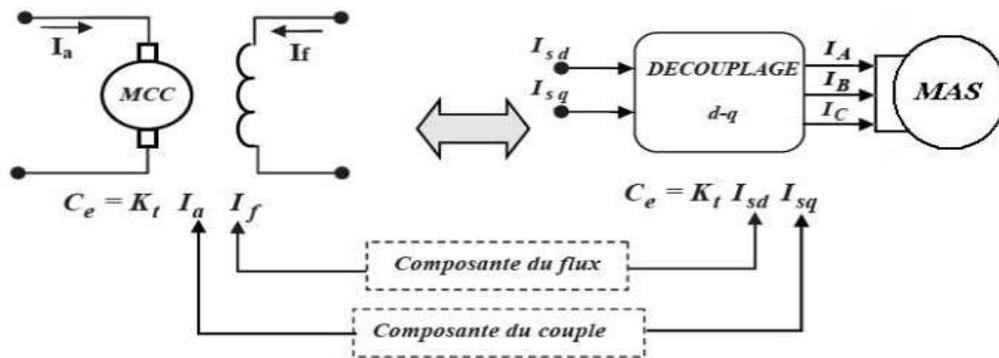


Figure I-8 Schéma de principe de découplage de la MAS/MAC [9]

Il existe trois types de commande vectorielle à flux orienté selon la nature du flux :

1. Commande vectorielle à flux rotorique orienté.
2. Commande vectorielle à flux statorique orienté.
3. Commande vectorielle à flux de magnétisation orienté.

Le plus fréquemment utilisée est la commande vectorielle par orientation du flux rotorique, puisque dans le cas de la MAS non saturée, on obtient les équations du rotor complètement découplées.[10]

### I.10. Différents variateurs existants dans l'industrie

Les variateurs de fréquence sont surtout utiles pour économiser de l'énergie. Il s'agit donc de dispositifs en mesure d'agir pour l'environnement. Savez-vous que même de légères modifications de la vitesse peuvent aboutir à d'importantes économies d'énergie ? C'est pourquoi les variateurs de fréquence, également appelés variateurs de vitesse, offrent une méthode efficace de commande de la vitesse électrique. Ainsi, la réponse à la demande de charge reste pertinente et les performances des appareils demeurent optimales. [11]

#### I.10.1. Variateur vitesse marque VFD

Ce variateur de fréquence utilise une méthode de contrôle unique pour réaliser des moments de force élevés, une grande précision et une conduite à grande plage de régulation de la vitesse, avec des performances élevées.

- Il offre de bonnes performances anti-trébuchement et une capacité d'adaptation aux interférences de puissance, de température, d'humidité et de poussière, avec une stabilité grandement améliorée.
- En optimisant la technologie de contrôle PWM et la compatibilité électromagnétique, il répond aux besoins des utilisateurs en matière d'interférences électromagnétiques à faible bruit et faibles.
- Facile à installer, Simple à utiliser.
- Conçu avec un grand dissipateur thermique pour une meilleure dissipation de la chaleur [12]



Figure 1-9 Variateur vitesse VFD

### I.10.2. Variateur vitesse ACS5500 de ABB

Est simple à installer, à configurer et à utiliser, offrant un gain de temps considérable. Le variateur ACS550 dispose de l'inductance d'amplitude brevetée, qui réduit de 25 % les émissions de distorsion harmoniques totales en faisant correspondre l'inductance adéquate à la charge.

Le variateur ACS550 nécessite un dispositif de programmation. Les options incluent le panneau de base, le panneau ou le logiciel DriveWindow.

Le variateur ACS550 :

- Améliore la commande de vitesse du moteur
- Aide à éliminer les problèmes de démarrage et d'arrêt
- Peut être mis en réseau avec d'autres équipements d'automatisation
- Caractéristiques de compensation de charge
- Entrées d'alimentation monophasées ou triphasées



Figure 1-10 Variateur ACS550

### I.10.3. Etude d'un cas de variateur industriel standard : SINAMICS G120C demarque Siemens : cas de notre étude

Le variateur SINAMICS G120C est un dispositif de contrôle de la vitesse qui appartient à la gamme de produits de Siemens.



Figure I-11 : Variateurs de vitesse G120C

Voici une description détaillée du variateur SINAMICS G120C :

**Conception Compacte** : Le SINAMICS G120C est connu pour sa conception compacte, ce qui le rend idéal pour les applications où l'espace est limité. Son boîtier est généralement en métal pour une robustesse accrue et une meilleure dissipation de la chaleur.

- **Plage de Puissance** : Ce variateur est disponible dans une gamme de puissances allant de quelques kilowatts à plusieurs dizaines de kilowatts, ce qui le rend adapté à une grande variété de moteurs électriques.
- **Tension d'Alimentation** : Il peut être alimenté par une tension monophasée ou triphasée, en fonction des besoins de l'application.
- **Commande de Vitesse** : Le SINAMICS G120C permet de contrôler avec précision la vitesse du moteur. Il prend en charge plusieurs méthodes de commande, notamment la commande vectorielle, la commande V/f (tension/fréquence), et la commande à boucle ouverte.
- **Interface Utilisateur** : Le variateur est généralement équipé d'un écran LCD ou d'une interface utilisateur conviviale qui permet aux opérateurs de configurer et de surveiller facilement les paramètres du moteur.
- **Communication** : Il prend en charge divers protocoles de communication, ce qui facilite son intégration dans des systèmes de contrôle plus larges. Des options de communication telles que le Profibus, le Profinet, l'EtherNet/IP, etc., sont couramment disponibles.

- **Protection et Sécurité** : Le variateur est doté de fonctions de protection avancées pour le moteur et le variateur lui-même. Cela inclut la protection contre les surintensités, les surtensions, les sous-tensions, les surchauffes, etc.

**CONCLUSION :**

Dans ce chapitre, nous avons examiné en détail la conception des variateurs de vitesse ainsi que leur principe de fonctionnement. Nous avons également étudié la commande des moteurs asynchrones, en mettant en lumière deux approches : la commande scalaire et la commande vectorielle. La dernière méthode est généralement considérée comme la plus efficace par rapport à la première, qui présente certaines limitations. L'une de ces limitations réside dans le maintien du flux statorique à un niveau constant, indépendamment du couple demandé et de la vitesse du moteur. Cependant, il est important de noter que cette condition devient particulièrement difficile à maintenir à des vitesses réduites en raison de l'influence de la résistance du stator, ce qui entraîne une chute de tension significative.

Enfin nous avons donné une vue sur les variateurs de vitesse siemens G 120 et Les différents variateurs existant dans l'industrie en détaillant sur notre variateur SINAMICS G120C.

## **Chapitre II**

### **Etude détaillée du système de variation de vitesse considéré**

**II.1. Introduction**

Le deuxième chapitre présente la commande utilisée dans notre système avec laquelle nous allons travailler et qui est aussi présente dans le contrôle de la vitesse du moteur via un variateur vitesse, ainsi que la description matérielle des différents appareils utilisés par exemple : le variateur de vitesse G120C et API S7-1500, le débitmètre et le moteur asynchrone qui est la pompe Taifu QB50.

Ensuite nous parlons sur la Communication entre API et variateur de vitesses G120C, Cette connexion est faite par télégramme -1.

En fin, nous présenterons les différents logiciels et langages utilisés dans ce travail, parmi eux TIA Portal V16.1, S7-PLCSIM V16.1 et WinCC Runtime. Leur but est de faciliter le contrôle du processus du système.

**II.2. Aperçu détaillé sur la commande du Moteur**

Le variateur de vitesse utilise deux méthodes pour régler la vitesse de moteur asynchrone.

- ✓ Commande U/f
- ✓ Régulation vectorielle

La commande U/F est utilisée lorsque la fréquence est inférieure à la fréquence nominale. Par contre, la commande vectorielle sera utilisée lorsque la fréquence est égale à la fréquence nominale.

**II.2.1. Commande U/f**

Dans la variante U/f "Régulation du courant d'excitation (FCC)", le variateur régule le courant moteur lors des phases à faible vitesse (courant de démarrage).

La commande U/f est une commande de vitesse qui possède les caractéristiques suivantes :

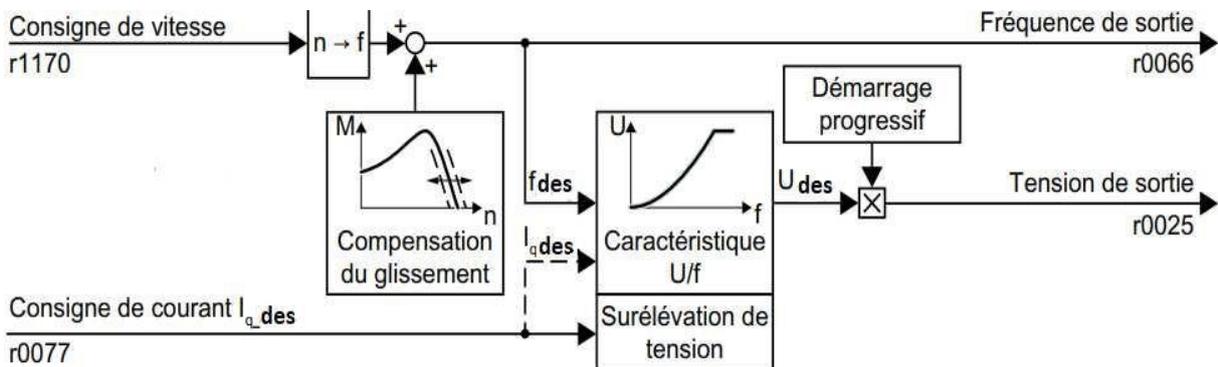


FIGURE II-1 : Diagramme fonctionnel simplifié de la commande U/f [13]

- ✓ Le variateur ajuste la tension de sortie à l'aide d'une caractéristique U/f de control U/f.

- ✓ La fréquence de sortie résulte principalement de la référence vitesse et du nombre de paires de pôles du moteur.
- ✓ La compensation de glissement corrige la fréquence de sortie en fonction de la charge et augmente donc la précision de la vitesse.
- ✓ Le fait d'avoir appliqué une boucle de régulation rend le contrôle U/f stable dans tous Cas.
- ✓ Dans les applications qui exigent une précision de vitesse extrêmement élevée, un survoltage en fonction de la charge peut être sélectionné (régulation de courant d'excitation, FCC).

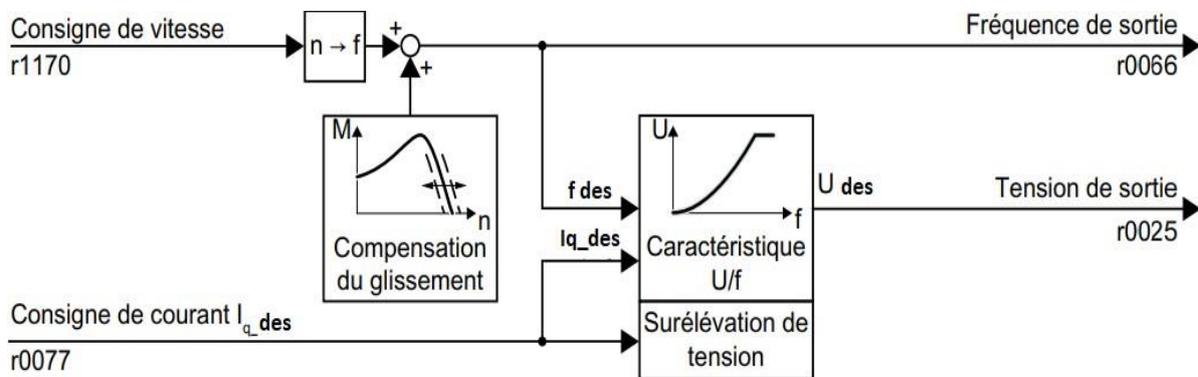


Figure II-2 : par défaut de la commande U/f après sélection de Standard Drive Control [13]

La sélection de la classe d'applications Standard Drive Control lors de la mise en service rapide adapte la structure et les possibilités de réglage de la commande U/f comme suit :

- ✓ Régulation du courant de démarrage : Au cours des phases à faible vitesse, un courant moteur régulé réduit la tendance aux vibrations du moteur.
- ✓ Lors de la montée en vitesse, le variateur passe de la régulation du courant de démarrage à une commande U/f avec surélévation de tension en fonction de la charge.
- ✓ La compensation du glissement est activée
- ✓ Un démarrage progressif n'est pas possible.
- ✓ Possibilités de réglage réduites [13]

### II.2.2. Régulation vectorielle

La régulation vectorielle se compose d'une régulation de courant et d'une régulation de vitesse de niveau supérieur.

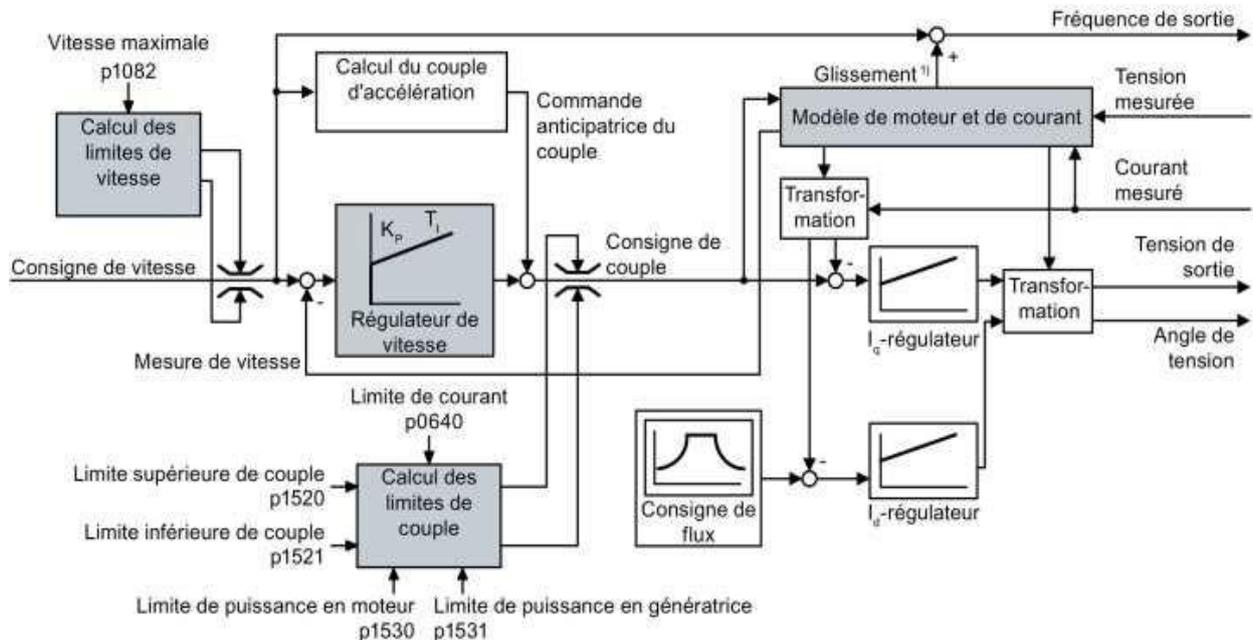


Figure II-3 : Diagramme fonctionnel simplifié pour la régulation vectorielle avec régulateur de vitesse [3]

Le variateur calcule les signaux de régulation suivants à l'aide du modèle de moteur à partir des courants de phase mesurés et de la tension de sortie :

- Composante de courant  $I_d$ .
- Composante de courant  $I_q$ .
- Mesure de vitesse pour la régulation de vitesse sans capteur.

La consigne de la composante de courant  $I_d$  (consigne de flux) est obtenue à partir de paramètres moteur. A des vitesses supérieures à la vitesse assignée, le variateur réduit la consigne de vitesse en fonction de la caractéristique de défluxage.

Lorsque la consigne de vitesse est augmentée, le régulateur de vitesse réagit par l'augmentation de la consigne de la composante de courant  $I_q$  (consigne de couple). La régulation réagit à l'augmentation de la consigne de couple par l'ajout d'une fréquence de glissement plus importante à la fréquence de sortie. La fréquence de sortie plus élevée provoque en outre un glissement plus important, proportionnel au couple d'accélération, dans le moteur. Les régulateurs  $I_q$  et  $I_d$  maintiennent le flux du moteur constant au moyen de la tension de sortie et règlent la composante de courant  $I_q$  appropriée dans le moteur.

Pour obtenir un comportement satisfaisant du régulateur, vous devez régler au moins les sous-fonctions indiquées sur fond gris dans la figure ci-dessus sur des valeurs adaptées à votre application :

- **Modèle de moteur et de courant** : lors de la mise en service rapide, réglez correctement les paramètres moteur de la plaque signalétique en fonction du type de raccordement (Y/ $\Delta$ ) et procédez à l'identification des paramètres moteur à l'arrêt.
- **Limites de vitesse et limites de couple** : lors de la mise en service rapide, réglez la vitesse maximale (p1082) et la limite de courant (p0640) en fonction de votre application.
- A la fin de la mise en service rapide, le variateur calcule les limites de couple et de puissance en fonction de la limite de courant. Les limites de couple effectives sont obtenues à partir des limites de courant et de puissance calculées et des limites de couples réglés.
- **Régulateur de vitesse** : utilisez la mesure en rotation de l'identification des paramètres moteur. Lorsque la mesure en rotation n'est pas possible, vous devez optimiser manuellement le régulateur.

### II.3. Description matérielle des différents appareils utilisés

#### II.3.1. Présentation variateur vitesse G 120C

SINAMICS G120C est le variateur universel répondant aux exigences les plus diverses de l'industrie et des métiers. La construction mécanique, le secteur automobile, le textile, l'impression, l'emballage et la chimie font tous confiance aux solutions SINAMICS G120C. Ce variateur est aussi mis en œuvre dans le monde entier pour des applications transversales, telles que dans la manutention, la sidérurgie, l'extraction du pétrole/gaz, ainsi que dans la production d'énergie renouvelable.

La compacité du SINAMICS G120C découle de plusieurs facteurs de conception visant à minimiser l'encombrement spatial tout en maintenant des performances élevées et une dissipation de chaleur efficace. Cette caractéristique de compacité en fait un choix populaire pour les applications où l'espace est limité, tout en offrant une solution de contrôle de la vitesse fiable et efficace pour une plage de puissance de 0,55kW à 18.5kW, en fait le système idéal pour les applications standards. Les différents constituants disponibles vous permettent d'assembler le variateur optimal pour vos besoins.

Des innovations permanentes permettent de faire évoluer sans cesse le système G120C en y ajoutant de nouveaux éléments et de nouvelles possibilités :

- ✓ Facilité d'emploi de l'installation à la maintenance.
- ✓ Robuste et endurant pour les environnements difficiles.

- ✓ Econome en énergie grâce à une multitude de fonctions.
- ✓ Nombreuses fonctions de sécurité. [14]

### II.3.2. Description de variateur vitesse G120C

SINAMICS G120C est un système d'onduleur compact comprenant un module de puissance intégré en Unité de contrôle (CU)

#### Unité contrôle

Cette unité contrôle et surveille le module d'alimentation et le moteur connecté en utilisant différents types de commande pouvant être sélectionnés. Il prend en charge la communication avec un local ou dispositifs de contrôle et de surveillance centralisés.

L'unité de contrôle exécute des fonctions de contrôle en boucle fermée pour l'onduleur. En plus de la commande en boucle fermée, il dispose de fonctions adaptables à l'application particulière par paramétrage.

Le Control Unit comprend une structure de quantité d'E / S définie, une interface de bus de terrain spéciale et éventuelles fonctions de sécurité supplémentaires.

#### Module de puissance

Le module de puissance alimente le moteur dans une plage de puissance de 0,55 kW à 18.5 kW. Le module d'alimentation est contrôlé par un microprocesseur dans l'unité de contrôle. La technologie IGBT de pointe avec modulation de largeur d'impulsion est utilisée pour atteindre le plus haut degré de fiabilité et de fonctionnement flexible du moteur. Les fonctions de protection complètes offrent un haut degré de protection pour le module de puissance et le moteur.

### II.3.3. Composant du module de puissance

#### ✓ Filtre de réseau

Avec un filtre réseau, le variateur atteint une classe d'antiparasitage supérieure. Pour les variateurs avec filtre réseau intégré, aucun filtre externe n'est nécessaire comme par exemple les filtres réseau ci-contre. Ces derniers correspondent à la classe A ou B selon EN55011



Figure II-4: Filtre de réseau PM240 [15]

✓ **Inductance de réseau**

L'inductance réseau prend en charge la protection contre les surtensions, lisse les harmoniques dans le réseau et compense les creux de commutation.



Figure II-5 Inductance de réseau PM240 [15]

✓ **Inductance de sortie**

Les inductances de sortie réduisent la charge de tension appliquée aux enroulements du moteur et la charge du variateur due aux courants capacitifs d'inversion dans les câbles.



Figure II-6 : Inductance de sortie PM240[15]

**II.3.4. Panneau de commande intelligent (IOP)**

Le pupitre opérateur « Intelligent Operator Panel (IOP) » a été conçu pour améliorer l'interface et les capacités de communication des variateurs SINAMICS. L'IOP se connecte au variateur via une interface RS232. Il a été conçu pour reconnaître automatiquement les dispositifs de gamme SINAMICS.

Le choix de la console IOP dépend du Control Units utilisé. Dans notre cas, la console IOP connectée à notre CU a la référence suivante : 6SL3255-0AA00-4HA0. [30]

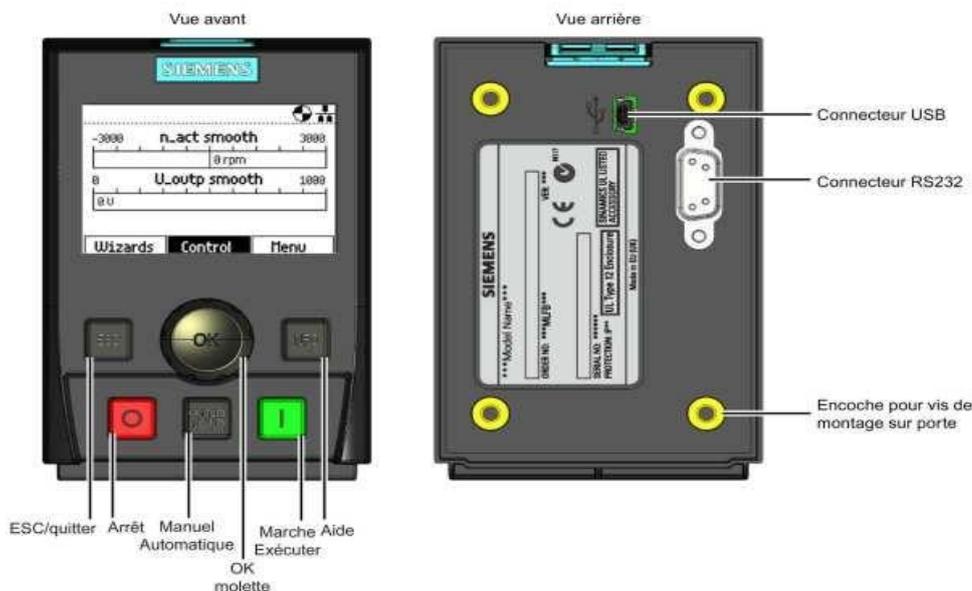


Figure II-7: Représente Panneau de commande intelligent (IOP) [16]

#### II.4. Généralités sur les automates programmables

L'automate programmable industriel A.P.I (Programmable Logic Controller PLC) est un appareil électronique programmable. Il réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services et dans l'agriculture. [17]

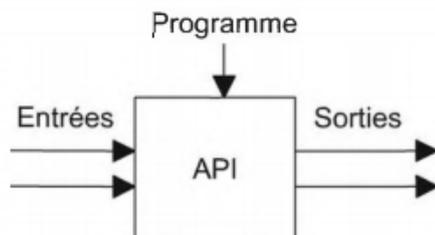


Figure II-8: Automate programmable industriel. [17]

La principale force d'un automate programmable industriel (API) réside dans sa remarquable capacité à établir des communications au sein de l'environnement industriel. En plus de son unité centrale et de son alimentation, il est principalement composé de modules d'entrées/sorties qui agissent comme des interfaces de communication avec le système de commande industriel. Ses rôles clés au sein d'un processus comprennent :

- D'assurer l'acquisition de l'information fournie par les capteurs
- En faire le traitement.
- Elaborer la commande des actionneurs.
- Assurer également la communication pour l'échange d'informations avec l'environnement [17]

### II.4.1. Architecture des automates programmables industriels

#### Aspect extérieure

Les automates peuvent être de type compact ou modulaire.

De type compact, il intègre le processeur, l'alimentation, les entrées et les sorties selon les modèles. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes

De type modulaire, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules). Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.



Figure II-9: Automate programmable Siemens S7-1500, CPU 1512C-1PN

#### Structure interne

Un automate programmable industriel (API) est typiquement composé de modules disposés en rangée, incluant une alimentation, une unité centrale (CPU) basée sur un microprocesseur munie d'une carte mémoire, des interfaces d'entrées et de sorties, des interfaces de communication, des cartes spéciales, et un dispositif de programmation. On peut également le concevoir comme une unité qui intègre un grand nombre de relais, de compteurs, de temporisateurs, ainsi que des unités distinctes de stockage de données, généralement de type EEPROM. La Figure II.10 illustre la disposition de base d'un automate programmable industriel.

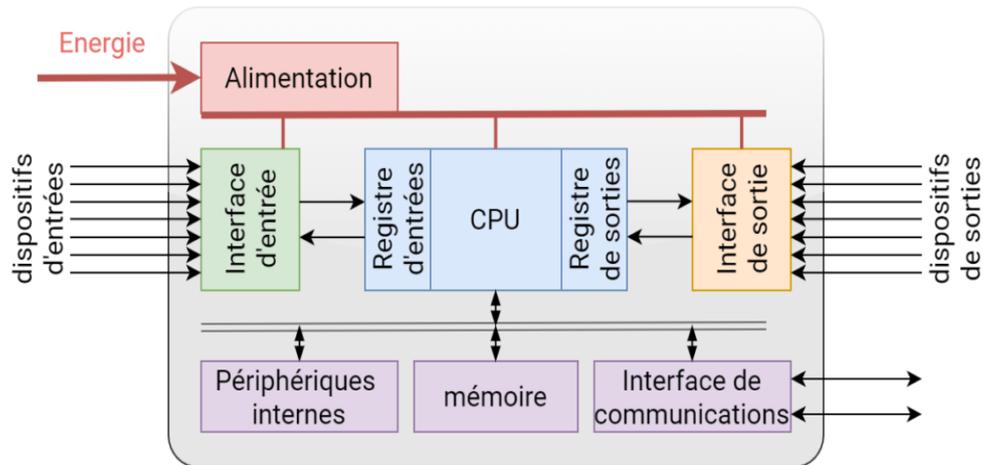


Figure II-10: Structure interne d'un API [18]

Module d'alimentation

Le bloc d'alimentation (Power Supply (PS) (Figure II. 11) est nécessaire pour convertir la tension d'entrée alternative (220 V) du secteur en une tension continue (24V, 48V) nécessaire au processeur et aux circuits des modules d'interface d'entrée et de sortie.

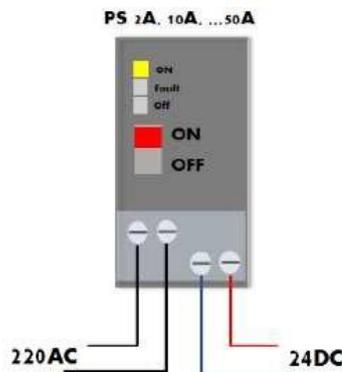


Figure II-11 :Module d'alimentation d'un API

La puissance des alimentations varie entre un API et un autre et demandent un courant allant de 2A à 50A, en fonction du nombre d'interfaces d'E/S alimentées par cette alimentation.

- **Le module CPU** est l'unité contenant le microprocesseur. Cette unité interprète les signaux d'entrée et exécute les actions de commande en fonction du programme enregistré dans sa mémoire, communiquant les décisions sous forme des signaux d'actions aux sorties, Aussi, ce module contient une interface de programmation afin de communiquer avec la console de programmation suivant un protocole bien déterminé.

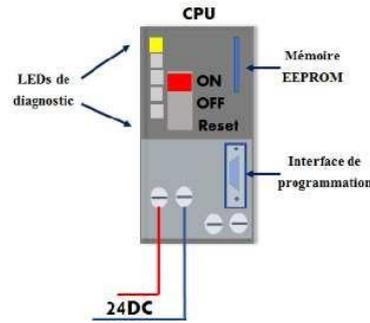


Figure II-12 : Module CPU d'un API

- **Une mémoire** : Elle permet le stockage des instructions constituant le programme de fonctionnement ainsi que diverses informations. Il peut s'agir de mémoire vive RAM (modifiable à volonté, mais perdue en cas de coupure de tension) ou de mémoire morte EEPROM (seule la lecture est possible).
- **Interfaces d'entrée/sortie** : Les cartes d'E/S permettent au processeur de recevoir des informations de périphériques externes (capteurs) et de les communiquer aux périphériques externes (Pré-actionneurs et actionneurs), généralement il y a deux types d'E/S, type Tout ou Rien (DI/DO) et analogique (AI/AO).

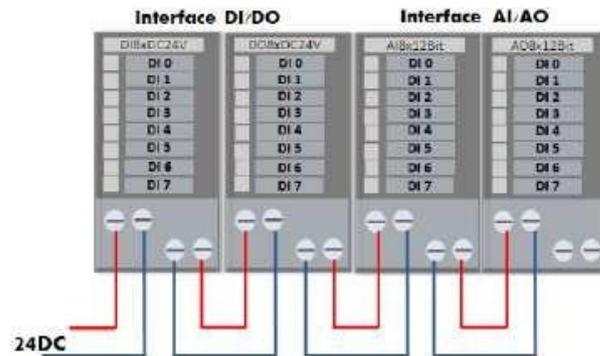


Figure II-13 : Interfaces d'entré/ sortie

Plus de ces modules, on trouve des modules spécieux d'E/S (carte PID, carte de comptage rapide ...etc.), ce type des cartes dotés des microprocesseurs, afin de simplifier les taches et soulager le module CPU.

**Console de programmation** : Le dispositif de programmation est utilisé pour introduire le programme souhaité dans la mémoire programmable. Généralement le programme est développé dans un PC ou une console spéciale donnée par le constructeur, puis transféré dans la mémoire du CPU par l'intermédiaire d'un câble de communication adéquat.

**Module de communication** : L'interface de communication est utilisée pour recevoir et transmettre des données sur des réseaux de communication depuis ou vers d'autres systèmes distants tels que les capteurs.

**SCADA, HMI, serveur OPC ...etc** : Il concerne des actions telles que la vérification du périphérique, l'acquisition de données, la synchronisation entre les systèmes et la gestion de la

connexion.

## II.5. Choix d'un automate programmable

Pour choisir un automate programmable, l'automaticien doit préciser :

- ✓ Le nombre et la nature des entrées et des sorties.
- ✓ Le type de programmation souhaitée et les besoins de traitement permettant le choix de l'unité centrale et la taille de la mémoire utilisateur
- ✓ La nature de traitement (temporisation, couplage, etc.).
- ✓ Le dialogue (la console détermine le langage de programmation)
- ✓ La communication avec d'autres systèmes.

### II.5.1.CPU 1512C-1PN utilisé

L'API est largement utilisée dans les applications de contrôle industriel, il emprunte la simplicité, la souplesse, et faible coût et l'adaptabilité conférée par leur nature programmable. Dans cette étude, nous avons utilisé un automate de type Siemens S7-1500 1512C-1PN, Ce PLC contient 32 entrées numériques et 32 sorties numériques, 5 entrées analogiques, deux sorties analogiques. [19]



Figure II-1 : S7-1500 1512C-1PN

### II.5.3. Objet technologique PID Compact [20]

L'objet technologique PID Compact met à disposition un régulateur PID continu avec optimisation intégrée. De manière alternative, vous pouvez configurer un régulateur à impulsion. Les modes de fonctionnement manuel et automatique sont possibles.

Dans une boucle de régulation, PID-Compact réalise l'acquisition continue de la mesure et la compare à la consigne souhaitée. A partir du signal d'écart en résultant, l'instruction PID

Compact calcule une valeur de réglage par laquelle la mesure est ajustée à la consigne de la façon la plus rapide et la plus stable possible.

**II.5.4. Paramètres PID**

Les paramètres PID sont affichés dans la fenêtre de configuration "Paramètres PID". Les paramètres PID sont adaptés à votre système réglé pendant l'optimisation. Vous n'avez pas besoin d'indiquer les paramètres PID manuellement.

L'algorithme PID fonctionne selon la formule suivante :

$$y = K_p \left[ (b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

Symbole	Description
y	Valeur de réglage de l'algorithme PID
K <sub>p</sub>	Gain proportionnel
s	Opérateur de Laplace
b	Pondération de l'action P
w	Consigne
x	Mesure
T <sub>i</sub>	Temps d'intégration
a	Coefficient de l'action par dérivation (action par dérivation T1 = a × T <sub>D</sub> )
T <sub>D</sub>	Temps de dérivation
c	Pondération de l'action D

Tableau II-1 :Les paramètres de l'équation de régulation PID

**Gain proportionnel**

La valeur indique le gain proportionnel du régulateur. PID\_Compact ne fonctionne pas avec un gain proportionnel négatif. Inversez le sens de régulation dans Réglages de base > Type de régulation.

**Temps d'intégration**

Le temps d'intégration détermine le temps de réponse de l'action I. La désactivation de l'action I s'obtient avec un temps d'intégration = 0.0.

**Temps de dérivation**

Le temps de dérivation détermine le temps de réponse de l'action D. La désactivation de l'action D s'obtient avec un temps de dérivation = 0.0.

**II.5.5. Valeur de réglage**

PID\_Compact met trois valeurs de réglage à disposition. La valeur de réglage que vous

utilisez dépend de votre actionneur.

### Output PER

L'actionneur est adressé via une sortie analogique et est commandé à l'aide d'un signal continu, par exemple 0...10 V, 4...20 mA.

### Output

La valeur de réglage doit être mise en forme dans le programme utilisateur, par ex. parce que l'actionneur présente un comportement non linéaire.

### Output PWM

L'actionneur est commandé par une sortie TOR. Des temps d'activation et de désactivation variables sont formés à partir d'une modulation de largeur.

Le graphique suivant illustre l'intégration des paramètres dans l'algorithme PID (FigureII-15)

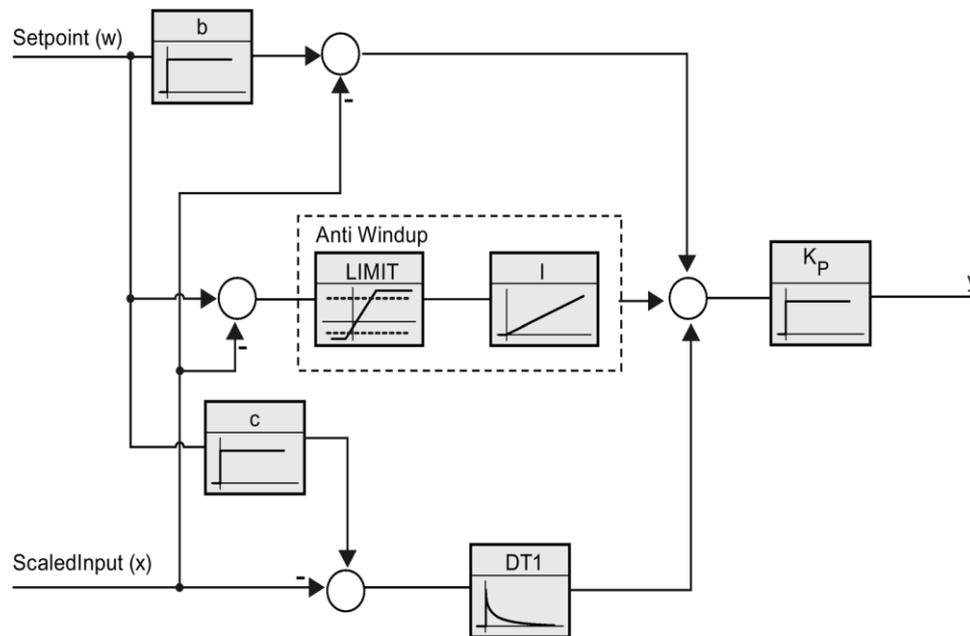


Figure II-15: schéma fonctionnel de PID Compact [20]

## II.6. Système de régulation de débit :

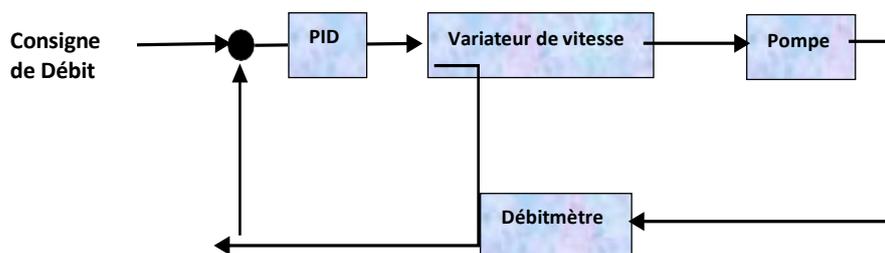
Dans un système de canalisation, le débit de l'écoulement à la sortie de la pompe doit être contrôlé, pour se faire on utilise : La régulation de débit par le biais d'un variateur de vitesse à l'entrée de la pompe QB50.

Après avoir ajusté les paramètres pour la commande en  $U/f = \text{constante}$  de la vitesse de la pompe, ceci va constituer la première boucle de régulation interne de vitesse. La deuxième boucle externe, en cascade par rapport à la première, sera pour la régulation de débit dans l'installation, en s'appuyant sur les mesures de débit récolté par le S7-1500, à partir du débitmètre.

### II.7. Description du système de régulation du débit (Variateur-pompe-débit)

Le débit du fluide délivré par la pompe à travers le système de canalisation est calculé par le biais d'un capteur de débit (débitmètre à turbine). Dans ce qui va suivre nous allons étudier le contrôle continu du débit par le moyen de la pompe. En effet la vitesse de la pompe est contrôlée par un variateur de vitesse de la gamme Siemens et le mot de commande de vitesse est calculé à travers un régulateur PID intégré dans logiciel STEP 7.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit :



#### II.7.1. Les caractéristiques de la pompe Taifu QB50 :

MODEL	OUTPUT POWER		MAX. FLOW l/min	MAX. HEAD m	MAX. SUCT m	INLET/OUTLET In	GW kg	PACKING DIMENSION mm	QUANTITY set
	kw	hp							
QB50	0.22	0.3	30	20	8	1" x 1"	4.6	265 x 135 x 160	4100
QB60	0.37	0.5	35	32	8	1" x 1"	5.0	295 x 130 x 175	3000
QB70	0.55	0.75	45	48	8	1" x 1"	8.5	337 x 190 x 210	1800
QB80	0.75	1.0	45	53	8	1.5" x 1.5"	8.8	337 x 190 x 210	1800

Tableau II.2 : Caractéristique des pompes Taifu QB.

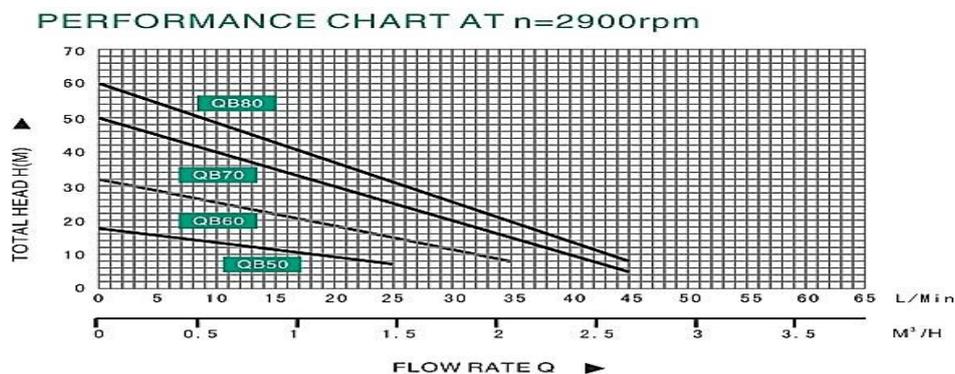


Figure II-16: Courbe des hauteurs manométriques en fonction de débit de pompe Taifu QB.

La pompe considérée est QB50 :

D’après le tableau le débit maximal de la pompe QB50 est :

30L/Min-→le régime de fonctionnement est donc 100% Hauteur manométrique → H(m)

Débit → Q(L/Min)

Régime de la pompe→R(%)

D’après la courbe on a :

H(m)			8	10	12	14	16	18	20
Q(L/Min)	30	27	25	21	17	12	8	1	0
R(%)	100	90	83.33	70	56.67	40	26.67	3.33	0

Tableau II-3 :Rapport Régime/Débit de la pompe taifu QB50

### II.7.2. Débitmètre à turbine[21] :

Un débitmètre à turbine pour liquides est utilisé pour mesurer le débit ou le débit total volumétrique et son principe de fonctionnement est relativement simple. Quand le liquide s’écoule dans le débitmètre, il entre en contact avec les pales de la turbine qui tournent autour de l’axe central du corps de la turbine. La vitesse angulaire (de rotation) du rotor de la turbine est directement proportionnelle à la vitesse du liquide traversant la turbine. La sortie résultante est mesurée par un détecteur électrique monté sur le corps du débitmètre.

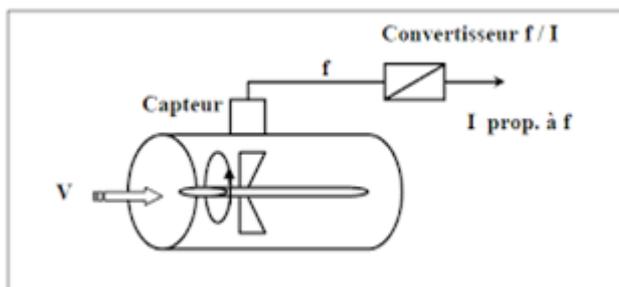


Figure II-2 : Débitmètre à turbine

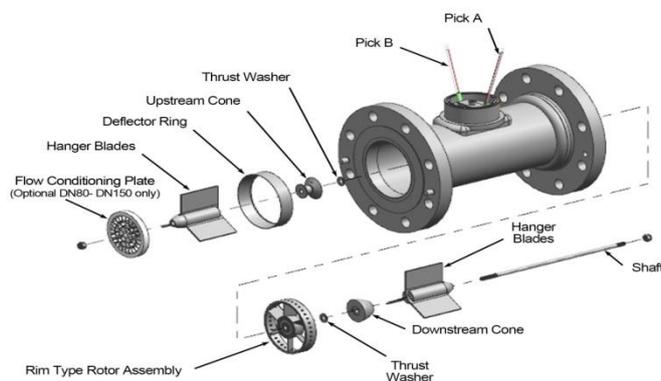


Figure II-3 : Constitution d'un débitmètre à turbine[21].

## II.8. Communication entre l'API et control unit de variateur de vitesse G120C :

Avec l'aide du logiciel TIA Portal, nous avons configuré notre contrôleur en lui attribuant l'adresse IP 192.168.0.1 (vous pouvez définir toute adresse IP qui ne peut pas entrer en conflit). [23]

On commence par ajouter la CPU et SIEMENS G120 Drive voulu sur l'environnement TIA PORTAL. Dans le catalogue de matériel TIA, nous sélectionnerons le module de commande SIEMENS G120C Drive module selon le numéro de série matériel exact.

Catalogue/Autres appareils/Profinet IO/Variateurs /Siemens AG / SINAMICS / SINAMICS G120C 1PN V4.5

Tout d'abord c'est quoi un Réseau PROFINET : SIEMENS adopte PROFINET comme une technologie de communication pour connecter les différents appareils d'un réseau.

### II.8.1. Définition de PROFINET

PROFINET est un réseau de haut niveau utilisé pour les applications d'automatisation industrielle. Il est basé sur des technologies Ethernet standard.

Il utilise un matériel et des logiciels Ethernet traditionnels pour définir un réseau qui structure la tâche d'échanger des données des alarmes et des diagnostics avec des contrôleurs programmables et d'autres contrôleurs d'automatisation [23].



Figure II-4: Exemples de connecteurs PROFINET [23]

### II.8.2. Définition Ethernet

Ethernet est la technologie de réseau local la plus largement installée. C'est un protocole de couche liaison dans la pile TCP / IP, décrivant comment les périphériques en réseau peuvent formater les données pour la transmission à d'autres périphériques réseau sur le même segment réseau et comment mettre ces données sur la connexion réseau d'erreurs pour détecter les problèmes de transmission [23].

Ensuite, nous connecterons à la fois l'API de l'appareil et le module d'entraînement au sous-réseau PROFINET (PN / IE), dans la vue réseau, le sous-réseau PN/IE est représenté dans la Figure II. 20. Il faut noter que l'adresse IP donnée à l'API et à l'unité de contrôle du variateur doit être différente et dans le même sous-réseau.

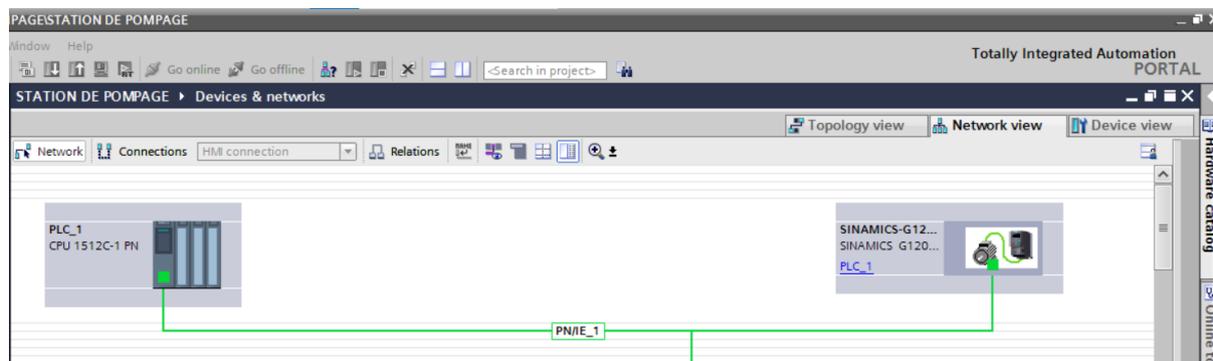


Figure II-5: Schéma représentant la connexion entre CU et S7 1500 sous-réseau PN/IE

Sur un réseau PROFINET on a des échanges de données avec une période de bien défini dans le sens réception c'est la CPU qui envoie des informations au variateur et dans le sens émission c'est l'inverse transfert du variateur vers la CPU.

Nous allons maintenant ajouter un télégramme standard avec le module de commande Drive à partir de la vue du réseau dans TIA. [24]

Le télégramme comporte un en-tête et une terminaison qui est un format général des trams PROFINET généré par le système. A l'intérieur, les données sont séparées en deux morceaux, une première zone contenant des informations pour lire ou modifier directement les registres interne du variateur ; c'est la PKW zone qui est optionnel. Un télégramme peut contenir

seulement des données PZD.

PZD est le nom de la seconde zone où on va envoyer les ordres de marche, les consignes et aussi recevoir des informations sur l'état de fonctionnement du variateur comme par exemple une vitesse, une position, une puissance, etc.....

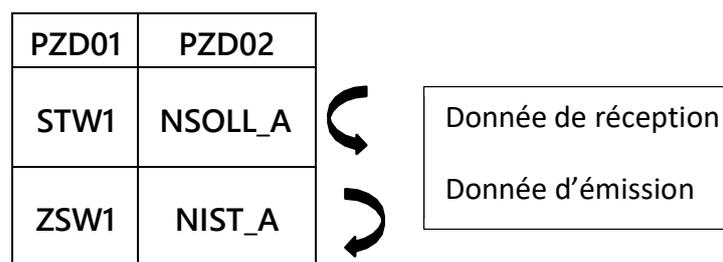
En résumé, un télégramme est donc un assemblage de différentes zones ; zones qui sont elles-mêmes découpées en mots de 16 bits.

Il existe différents types de télégramme contenant plus ou moins de PZD ; c'est à nous de choisir le type qui contiendra toutes les informations dont on a besoin.

### II.8.3. Télégramme utilisé pour le mode de régulation de vitesse

Voici le tableau de choix pour le mode régulation de vitesse uniquement :

Au niveau API            Au niveau CU



On travaille avec le télégramme de type 1, on a deux mots 'PZD' en réception et 2 autres mots en émission.

**Remarque :** Il faut noter que :

- ❖ *STW1 et NSOLL\_A sont 02 mots de sorties d'API qui vont commander un périphérique physique, par conséquent, ils doivent être assigné à la zone mémoire 'Q'. Dans notre cas d'étude, on a choisi le mot 'Q256'*
- ❖ *ZSW1 et NIST\_A sont 02 mots d'entrée à l'API qui ont été envoyés par le même périphérique physique, par conséquent, ils doivent être assigné à la zone mémoire 'P'. Dans notre cas d'étude, on a choisi le mot 'I256'*

**En réception :**

1. PZD1= STW1 ⇒ Mot de commande (envoyé par l'API, **reçu** par le CU) qui comporte les bits de contrôle de notre CU (variateur de vitesse). Exemple : Activation/Désactivation, Marche/Arrêt, Commande par rampe, Contrôle par

API, etc...

2. PZD2= NSOLL\_A ⇒ Consigne de vitesse envoyée au CU.

#### **En émission :**

1. PZD1= ZSW1 ⇒ Mot d'état ⇒ (**émis** par le CU, reçu par l'API) donne les informations ou l'état actuel en temps réel de notre CU (en marche/arrêter, rotation avant/arrière, freiner activer/désactiver, etc....)
  2. PZD2= NIST\_A ⇒ valeur de vitesse réelle ⇒ mot donnant la valeur exacte de la vitesse de rotation du moteur à chaque instant.
5. Au final deux mouvements arrivés à l'automate et c'est tout naturellement qu'on va aller trouver en entrée zone mémoire (I 256) et deux mots en sortie de l'automate donc zone mémoire (Q 256).

## **II.9. Présentation des différents logiciels et langage**

### **II.9.1. Description du logiciel TIA (Totally Integrated Automation) portal**

La plateforme Totally Integrated Automation Portal est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP 7 V16 et SIMATIC WinCC [25].

### **II.9.2. STEP 7 sur TIA portal :**

TIA Portal représente une version abordable et simplifiée du logiciel STEP 7 Professional Controller Software au sein du TIA Portal. Il sert à l'ingénierie des microcontrôleurs SIMATIC S7-1500 et à la configuration des SIMATIC HMI Basic Panels. Il est important de noter que Win CC Basic est inclus dans l'ensemble des logiciels pour une intégration fluide.

### **II.9.3. Win CC sur TIA portal :**

Win CC (TIA portal) est un logiciel d'ingénierie pour la configuration de pupitres SIMATIC, de PC industriel SIMATIC et de PC standard par le logiciel de visualisation. Le SIMATIC Win CC dans le TIA portal fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement d'ingénierie homogène pour la programmation et la configuration de solution de commande, de visualisation d'entraînement, c'est le logiciel pour toutes les applications IHM allant de solution de commande simple avec basic panel aux applications SCADA pour système multipostes basé sur PC [23].

#### II.9.4. PLCSIM

En l'absence de l'automate et des moyens, nous n'avons pas pu réaliser et tester réellement l'exécution du programme.

Nous avons utilisé un logiciel optionnel de TIA Portal, ce logiciel nommé PLCSIM permet de simuler un automate de la famille SIEMENS avec tous ces modules. Le simulateur présente une interface simple et accessible, en effet pour changer l'état d'une entrée, il suffit de cocher la case correspondante, les états des sorties changent automatiquement selon l'évolution du programme. Lors de la simulation et dans la fenêtre de programmation (CONT), chaque contact représentant une variable active est affiché en vert (Les contacts non actifs en pointillé). Ceci permet de suivre l'évolution du programme en détails. La simulation nous a permis de tester les différentes situations que peut affronter le système.

Nous concluons à la fin que notre programme répond exactement aux exigences du cahier des charges et qu'il peut donc être transféré du PC vers l'automate qui lui correspond.

L'application de simulation S7-PLCSIM nous a permis d'exécuter et de tester notre programme qu'on a simulé sur ordinateur. La simulation a été complètement réalisée au sein du logiciel TIA portal, cette application permet de tester des programmes destinés aux CPU S7, et de remédier à d'éventuelles erreurs.

Pour la CPU, la simulation est complètement réalisée au sein du logiciel TIA Portal V16. En effet, S7-PLCSIM dispose d'une interface comportant une CPU S7-1512C-1PN virtuelle et des modules d'entrées/sorties qui permettent de visualiser et forcer les différents états du programme.

#### II.10. Blocs de programme

Dans la programmation d'automate Siemens la structure de programme est séparée en quatre types de blocs différents :

- ✓ Blocs d'organisation
- ✓ Blocs de fonctions
- ✓ Blocs fonctionnels
- ✓ Blocs de données

Nous allons voir les différences et l'utilité de chaque bloc.

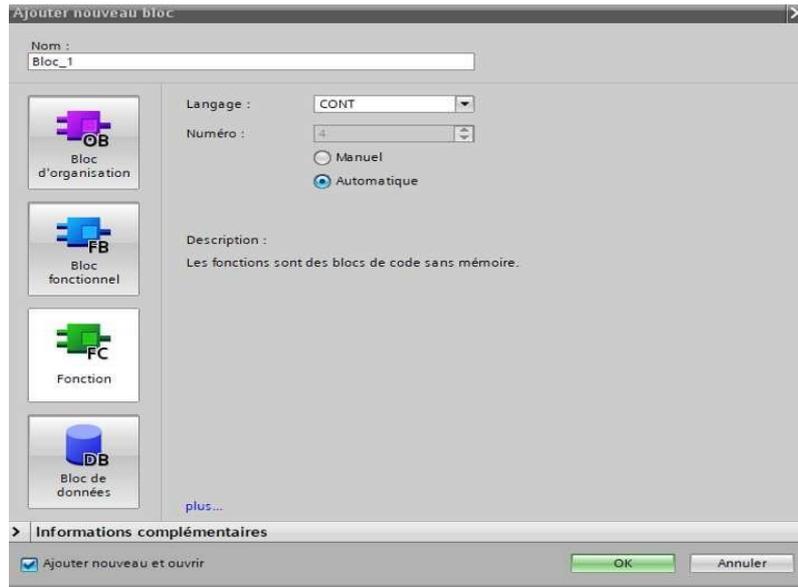


Figure II-6: Création d'un nouveau bloc.

### Blocs 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 (par exemple l'OB 1) [26].

### Blocs de fonctions (FC)



Ce sont des blocs de code sans mémoire. Elles n'ont pas de mémoire de données dans laquelle il est possible d'enregistrer les valeurs de paramètres de bloc. Les données des variables temporaires sont perdues après l'exécution de la fonction.

Si on veut mémoriser ces données, il faut utiliser des opérandes globaux [26].

### 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 [26].

### Blocs de données d'instance (DB)



L'appel d'un bloc fonctionnel est une instance. Les données avec lesquelles opère l'instance sont mémorisées dans un bloc de données d'instance. La taille maximale des blocs de données d'instance varie selon la CPU. Les variables déclarées dans le bloc fonctionnel déterminent la structure du bloc de données d'instance [26].

## **Conclusion**

Dans ce chapitre, nous avons introduit les détails sur la régulation de débit par la variation de vitesse de la pompe comme suit :

- ✓ La commande U/F utilisée lorsque la fréquence est inférieure à la fréquence nominale
- ✓ La commande vectorielle utilisée lorsque la fréquence est également la fréquence nominale.

Puis on a décrit les éléments principaux utilisés dans notre boucle de régulation :

(Le variateur G120C / L'automate S7-1512C-1PN / La pompe Taifu QB50 / Le débitmètre à turbine)

Ensuite, nous avons touché un aspect général des systèmes automatisés et certains des concepts de notre projet. Nous avons aussi parlé sur la communication entre API et le contrôle unitaire en expliquant le rôle important du télégramme -1.

Finalement, nous avons présenté les différents logiciels et langages utilisés qui vont nous faciliter le contrôle de notre système.

## **Chapitre III**

### **Conception et programmation de notre prototype**

### III.1. Introduction :

La commande de la pompe est assurée grâce à un environnement constitué de débitmètres, variateur de vitesse...etc. Ces accessoires permettent de collecter des informations sur le fonctionnement, et aide à mieux gérer ce dernier.

Dans ce chapitre, nous allons décrire la gestion et la régulation de vitesse de la pompe qui influence directement sur le débit dans le prototype de la station de pompage déjà décrit dans le projet de fin d'étude et effectuée à base d'un automate programmable Siemens.

Cette application décrit le fonctionnement hydraulique d'une station de pompage et met en relief le choix de la commande et la technique de régulation afin de développer le programme qui sera implémenté au sein de l'automate pour répondre au cahier de charge.

On s'intéresse dans ce chapitre sur les points suivants:

- fixer le type de commande du variateur de vitesse selon les possibilités offertes par ce dernier.
- Paramétrage de la communication entre le variateur de vitesse G120 et API Siemens.
- Ecriture du programme pour la gestion de l'installation et la gestion de variateur de vitesse
- Choisir un mode de régulation de débit par le bloc PID Compact dans Tia portal .

### III.2. Eléments de commande, de gestion et de supervision de la Maquette de la station de pompage

La paire Automate/Variateur constitue les deux éléments principaux pour la gestion et l'entraînement dans les installations de pompage.

Pour cette application nous avons utilisé :

- Un automate Siemens de la gamme SIMATIC S7-1500 doté des caractéristiques suivant :
  - CPU 1512C-1 PN , doté d'une mémoire de travail 250Ko,
  - Un port Profinet.
  - Une alimentation S307 2A alimenté en monophasé 220 V et qui délivre du 24V.
- PC, système d'exploitation Windows.
- Logiciel Tia portal V16.
- Interface Profinet pour PC (adaptateur PC)
- Une station PC est programmée avec le logiciel HMI Win CC Advanced, pour la vision des états de chaque élément et la gestion de toute l'installation.
- la communication entre automate et le SINAMICS G120C et Profinet par Télégramme-1.

On commande la pompe centrifuge QB50 par un variateur vitesse SINAMICS G120C de SIEMENS,

et par un débitmètre de sortie analogique, qui mesure le débit et le transmet vers l'automate S7-1500. D'une manière générale, la figure qui suit illustre la disposition de chaque élément :

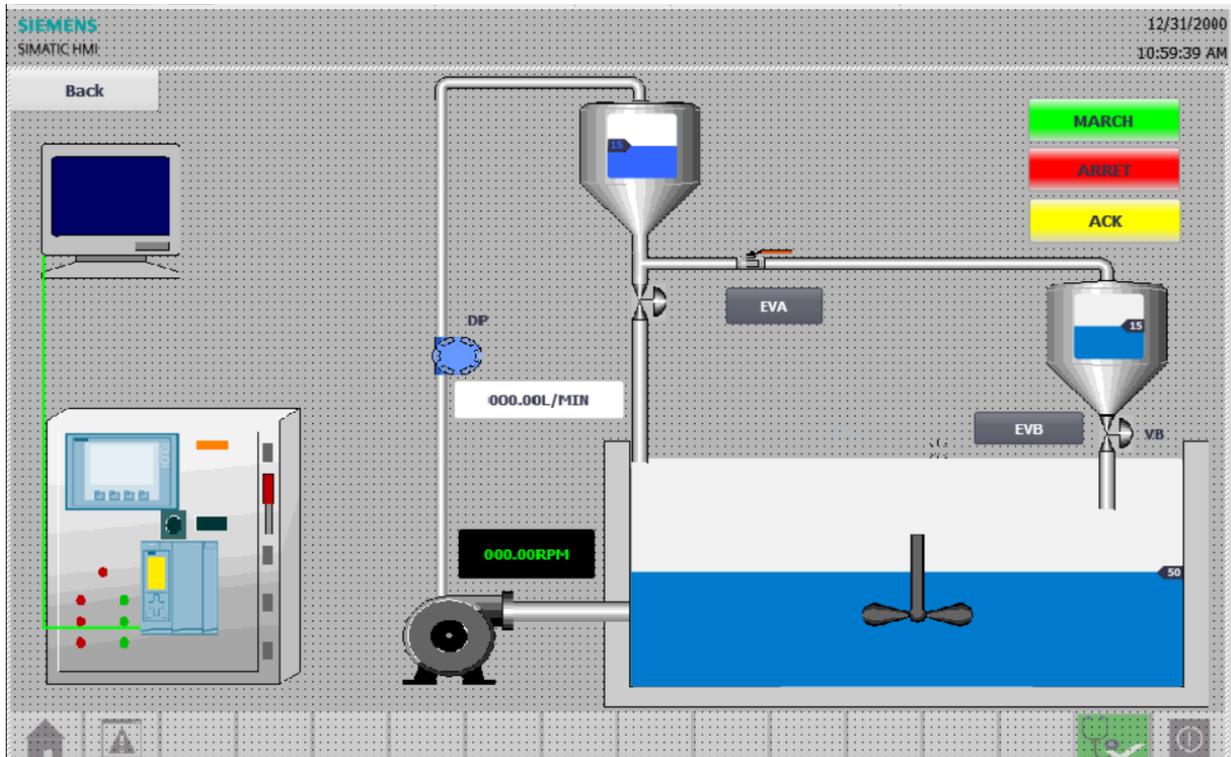


Figure III-1 : Vue de la maquette de station de pompage.

Le schéma qui suit montre les différents constituants de la maquette :

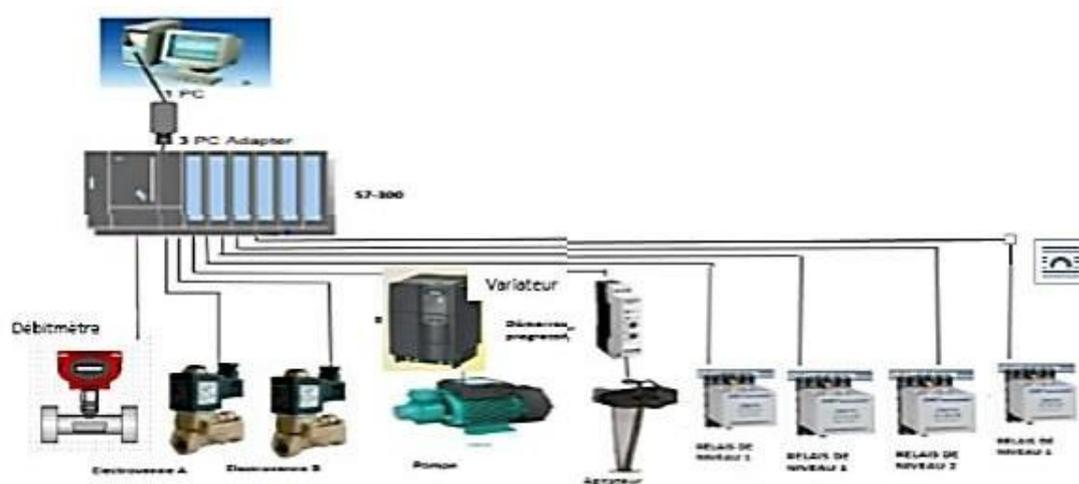


Figure III-2 : Synoptique du prototype.

- Description du Schéma électrique de la station

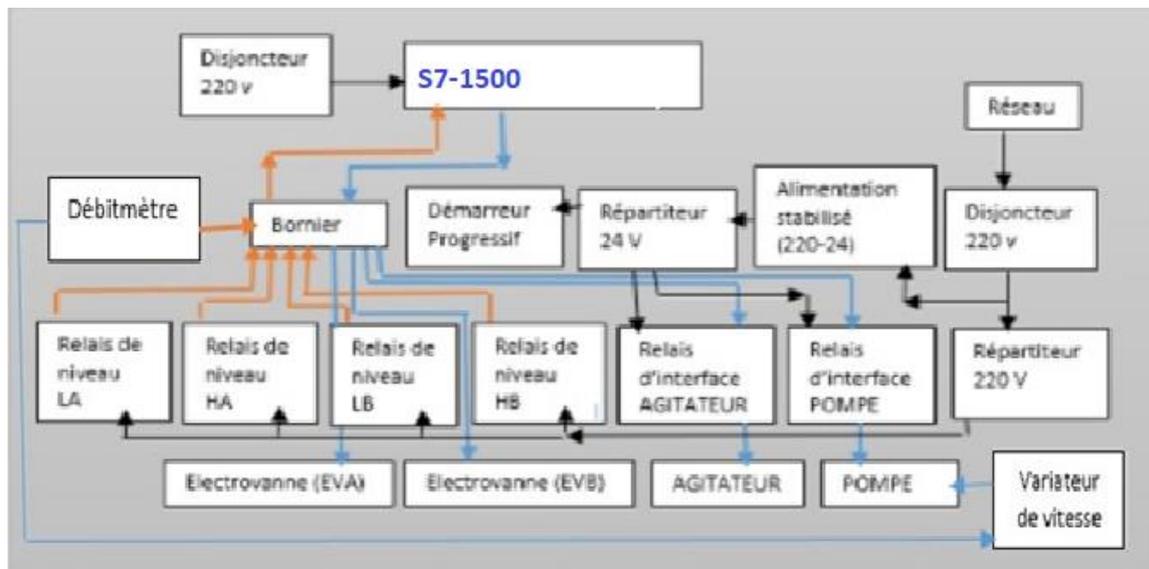


Figure III-3 : Description du Schéma Electrique de la station

### III.3. Développement du projet pour gestion du prototype de la station de pompage

L'étape suivante consiste à créer et développer le programme de contrôle et de commande qui sera implémenté dans l'automate afin de gérer notre installation.

Le graphique ci-après montre ces étapes de configuration de projet et les objets à créer :

Etape	Contenu de la tache	Réalisation
1	Création du projet « Station de pompage »	<ul style="list-style-type: none"> <li>• Démarrer le Tia portal</li> <li>• Créer un nouveau projet.</li> </ul>
2	Insertion et configuration d'un matériel	<ul style="list-style-type: none"> <li>• Insérer la CPU</li> <li>• adressages</li> </ul>
3	Programmer API	<ul style="list-style-type: none"> <li>• Créer des tables des mnémoniques.</li> <li>• Créer un bloc global de données.</li> <li>• Créer les différentes fonctions.</li> <li>• Appeler les blocs de programme dans le bloc principal [OB1].</li> </ul>
4	Visualisation du processus	<ul style="list-style-type: none"> <li>• Configurer le HMI Advanced .</li> <li>• Créer les vues.</li> </ul>
5	Tester en ligne	<ul style="list-style-type: none"> <li>• Tester le programme sur Plcsim</li> </ul>
		<ul style="list-style-type: none"> <li>• Tester la visualisation du processus sur WinCC flexible.</li> </ul>

Tableau III.1 : les étapes du développement du projet

### III.4. Création du projet

#### III.4.1. Paramétrage général de l'interface PG/PC

Avant de créer votre projet, vous devez impérativement définir le paramétrage de l'interface PG/PC de votre PC. Ce dernier est utilisé pour définir l'adresse locale des PG/PC, la vitesse de transmission dans le réseau Profibus ou Profinet, en vue d'une communication avec l'automate. Dans notre cas, on utilise la communication Profinet

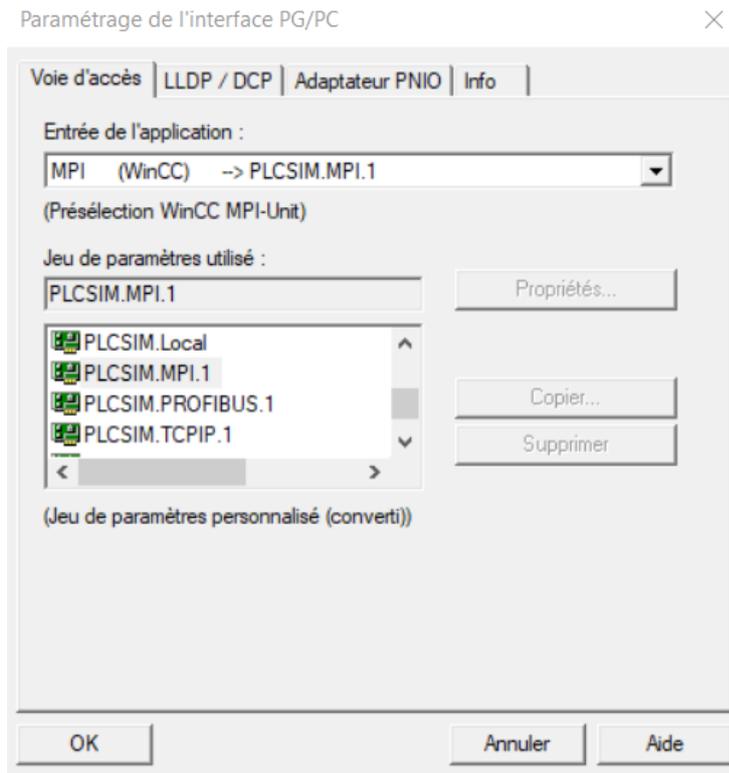


FIGURE III-4 : Paramétrage de l'interface PG/PC

### III.4.2. Procédure de création du projet

On commence par ouvrir logiciel TIA Portal V15, lorsque vous êtes sur la page d'accueil du logiciel, cliquez sur « Créer un projet », Après l'ouverture du logiciel TIA Portal :

- 1) Donnez un nom à votre projet
- 2) Spécifier le dossier dans lequel vous voulez enregistrer votre projet
- 3) Cliquez sur le bouton Créer

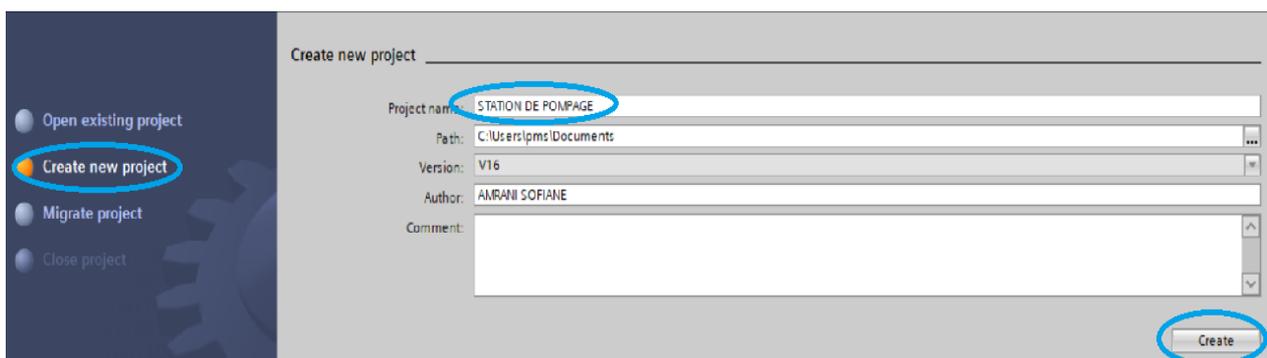


Figure III-5 : Création du projet.

### III.5. Ajout d'API

L'automate programmable industriel (API) est un appareil électronique programmable adapté à l'environnement industriel, permettant d'acquies des informations délivrées par des capteurs et délivrer des signaux de commande et des fonctions d'automatisme pour assurer la commande d'actionneur au biais du pré-actionneur. Les critères de choix essentiels d'un automate programmable

industriel API sont :

- Capacités de traitement de processeur (vitesse, données,..)
- Type d'entrée sortie nécessaire
- Nombre d'entrées sorties nécessaire
- Disponibilité et le prix

Le choix, dans notre projet, porte sur une API SIEMENS S7 1500 CPU 1512C-1PN dans les caractéristiques techniques sont citées dans le tableau ci-dessous :

Référence	6ES7 512-1CK01-0AB0
Mémoire programme	250 Kbyte
Mémoire de Travail « intègre » /enfichable	1 Mbyt/2 Gbyte; Carte mémoire SIMATIC
Tension d'alimentation	230 VAC
Courant d'entrée nominal	0.1 A
Puissance dissipée	14W

**Tableau III-2 :** Caractéristiques de notre contrôleur API

L'ajout d'un contrôleur est fonction de la référence de la CPU choisie. Pour ce faire, il suffit de cliquer sur "Contrôleurs", une liste de CPU apparaît, Dans notre cas, nous avons sélectionné la CPU 1512C-1PN de type S7-1500. Une fois votre automate choisi, ne pas oublier de donner un nom à votre appareil.

La figure III. 6 illustre toute la méthode pour l'ajout d'un API à votre projet.

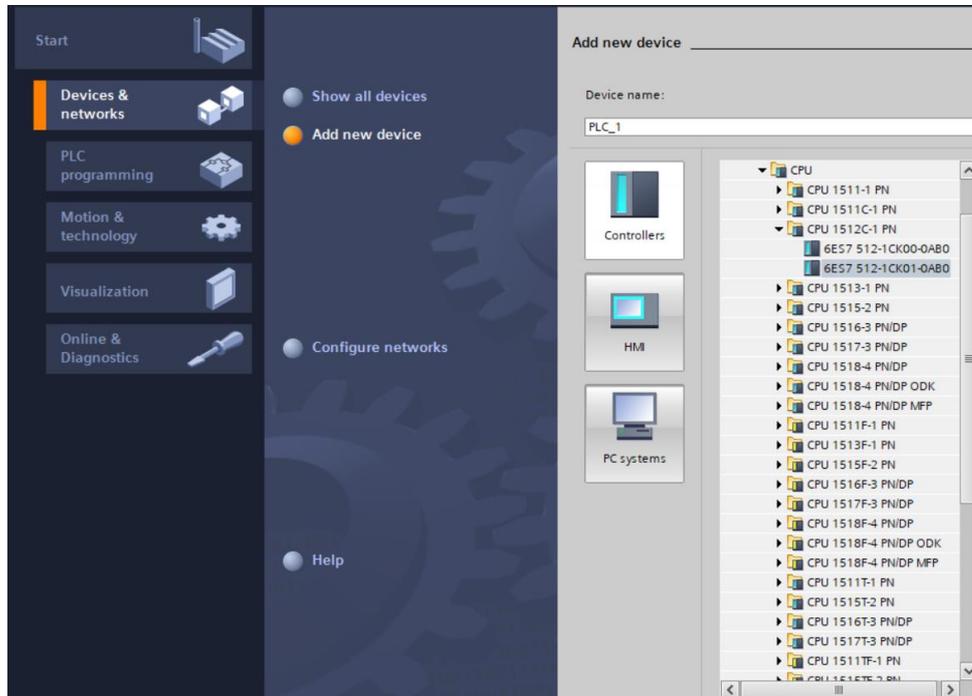


Figure III-6 : Procédure d'ajout du contrôleur

Pour configurer le CPU, double-cliquez sur celui-ci. Vous verrez toutes ses caractéristiques (sa taille

de mémoire de travail, les types de réseau de communication qu'il supporte, etc... Le plus important qui reste est la configuration de l'adresse IP de l'automate.

Dans la propriété de la CPU, double clic sur le contact Ethernet de la station fait apparaître la fenêtre d'inscription permettant de définir ses propriétés (Figure III.7).

Pour établir une liaison entre la CPU et la console de programmation et d'autres appareils communication via PROFINET il faut afficher aux différents appareils des adresses différents appartenant au même réseau. On utilisera comme adresse pour l'automate « 192.168.0.1 ».

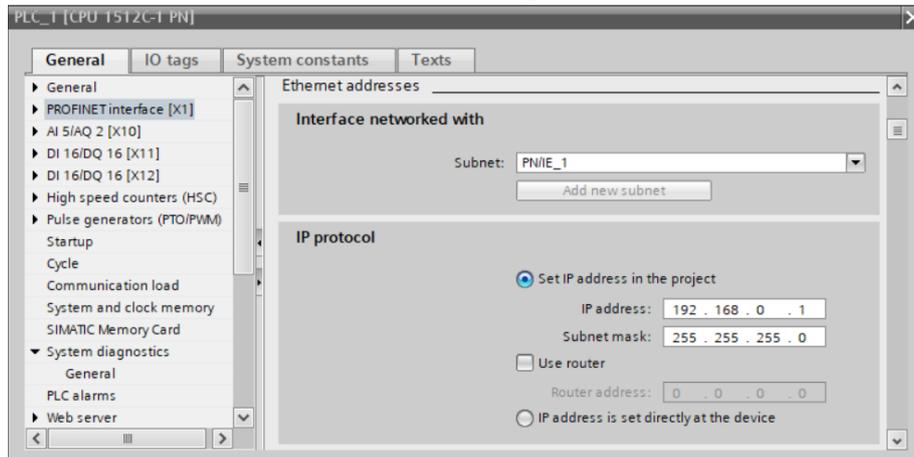


Figure III-7 : Adresse IP de l'API

### III.6. Ajoute du support d'IHM

#### Utilisation de système PC avec WinCC Runtime Advanced

Pour héberger notre IHM de commande du variateur, nous avons choisi dans notre cas le PC portable comme moyen. La figure ci-dessous montre clairement les étapes pour le choix demandé.

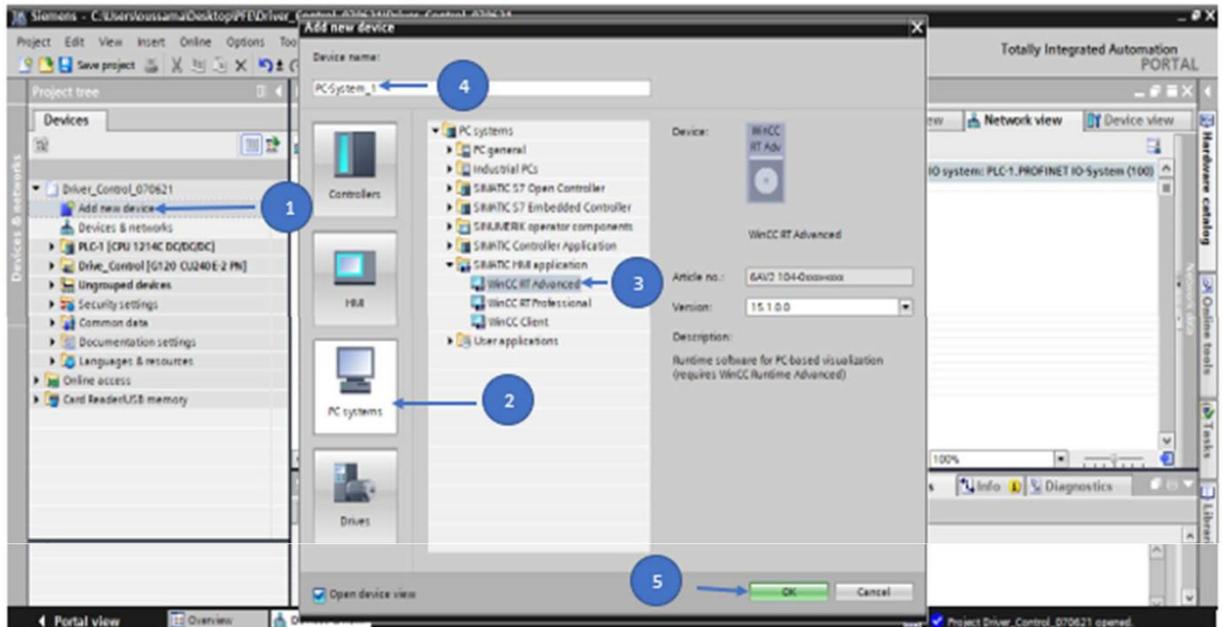


Figure III-8 : Choix du system PC et de WinCC

#### Ajout du module de Communication : module Profinet/Ethernet

Après ajout d'appareil, nous devons ajouter un moyen de communication. En d'autres termes, nous

sommes en train d'ajouter la carte réseau de votre propre PC. Cela va nous permettre de configurer pour prendre l'adresse réel de votre carte. La figure III.53 représente laméthode de dépôt de la carte. L'étape suivante est le choix de l'adresse IP (Figure III.54). Il faut choisir la même adresse IP de votre carte pour établir la communication entre l'ensemble des appareils.

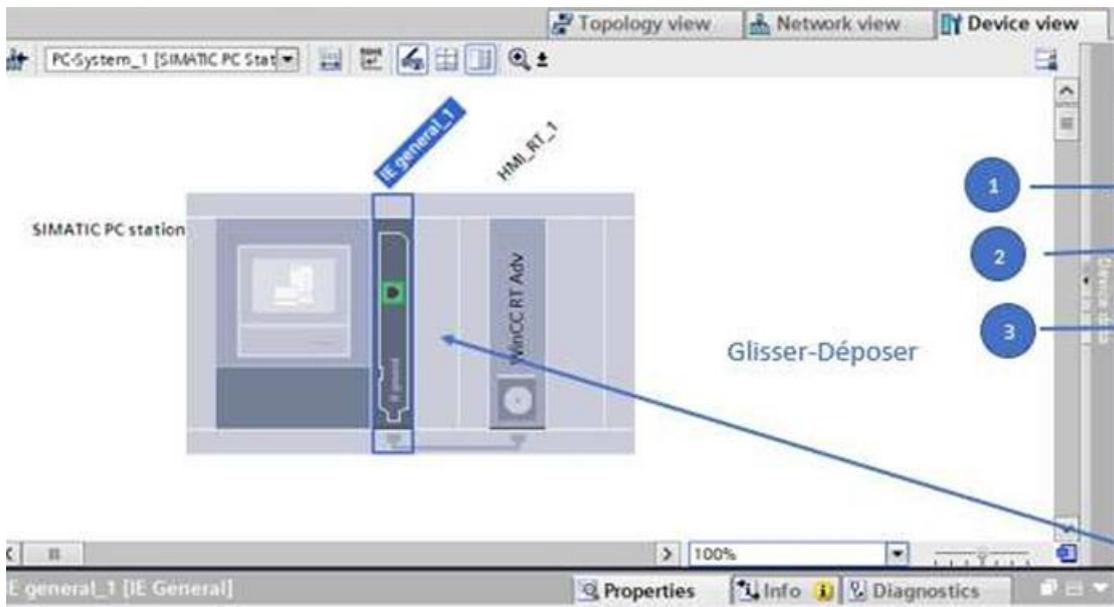


Figure III-9 : Dépôt du module de communication

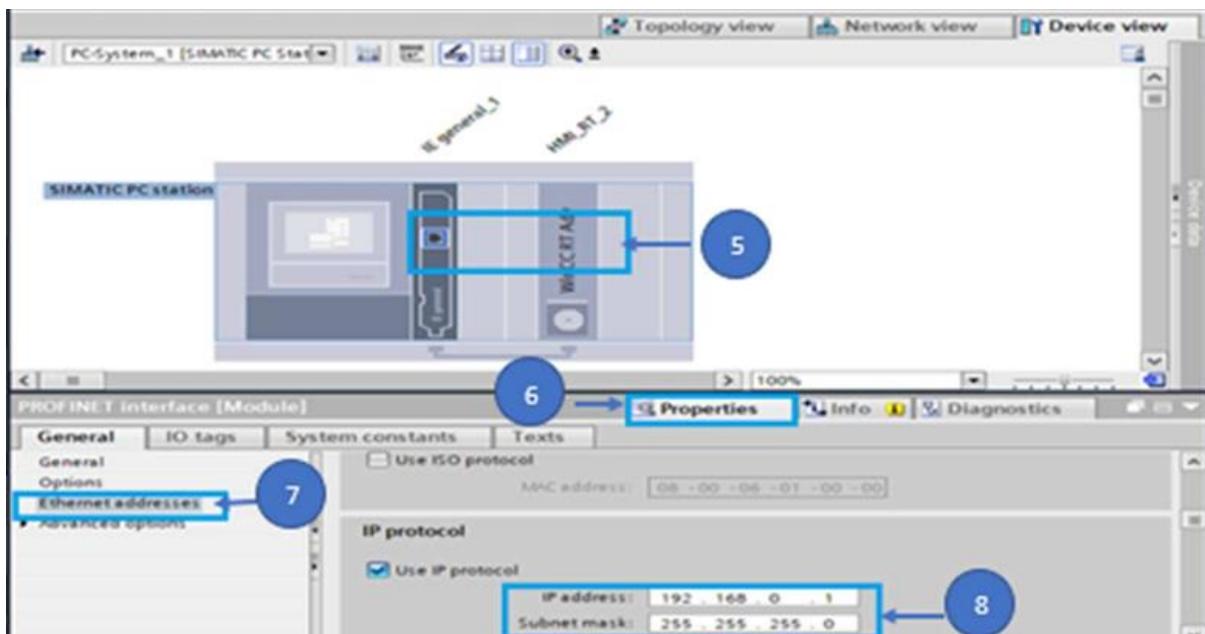


Figure III-10: Configuration du protocole IP du module de communication

### III.7. Ajout du variateur de vitesse et ces composants

Pour ajouter notre variateur ,on clique au côté droit du l'écran sur la barre d'outils sur -catalogue de matériel - puis sur Catalogue/Profinet IO/Drives/Siemens AG/SINAMICS/SINAMICS G120C PN 4.5V ,Comme il est montré dans la FigureIII-11 :

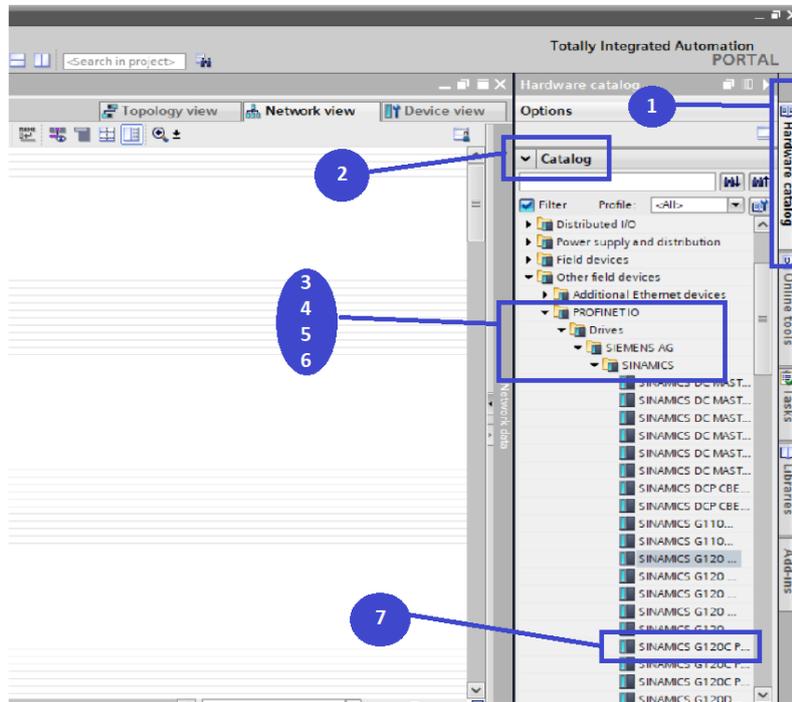


Figure III-11 :Choix du variateur

### Configuration du protocole IP

Cliquez à droite dans l'icône de profinet IO,puis choisir –Propriétés / Interface Profinet /Définir l'adresse IP dans le projet,l'adresse choisie est représentée dans la figure ci-dessous avec les étapes suivies

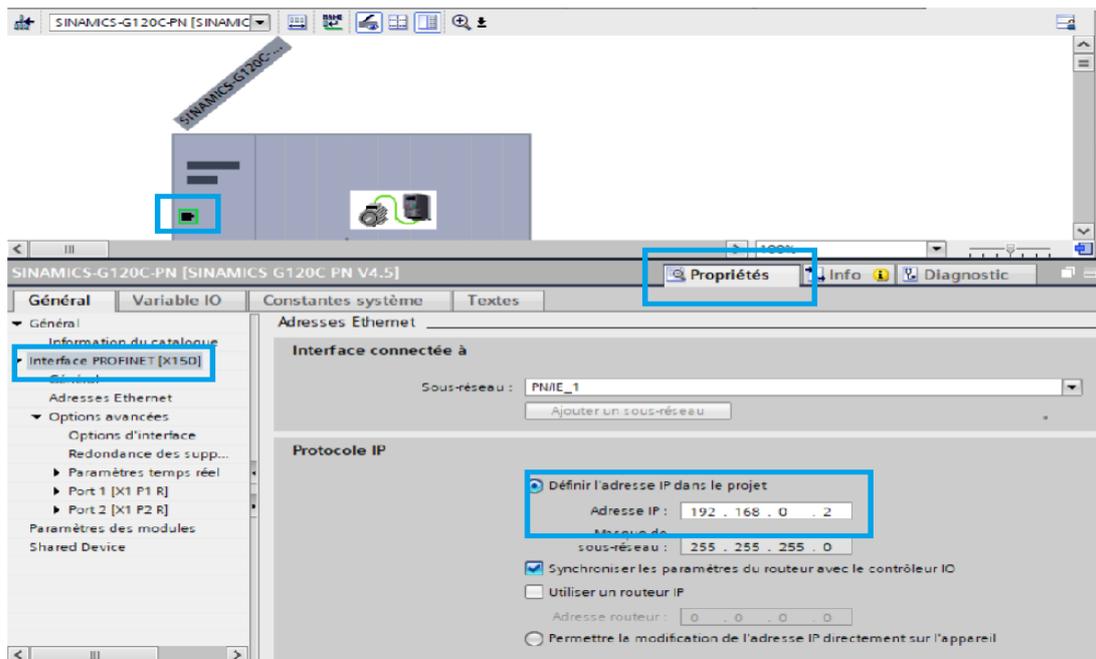


Figure III-12: Configuration du protocole IP de variateur vitesse

### III.8. Vérification des adresses IP des différents appareils

Comme phase finale, il faut mieux faire une vérification globale au niveau de la vue réseau pour éliminer tout conflit IP comme suit :

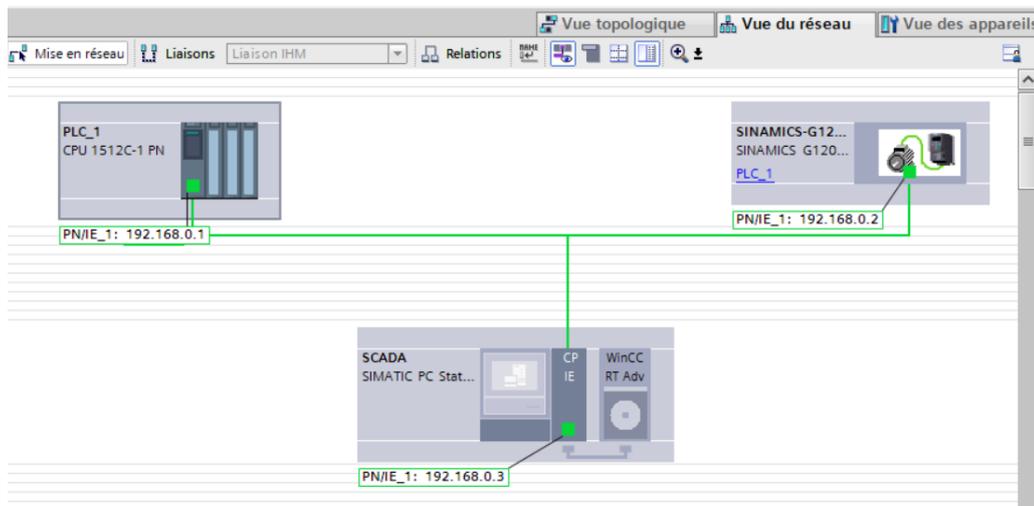


Figure III-13: Vérification des adresses IP

### III.9. Création d'IHM de commande du système

#### III.9.1. Présentation de l'IHM

L'Interface Homme-Machine (IHM) englobe un ensemble de matériels et de logiciels qui offrent à un utilisateur la possibilité d'interagir avec un système informatique. Au fil des décennies, l'évolution de l'interface homme-machine a été remarquable, passant des éléments rudimentaires tels que les boutons poussoirs et les afficheurs à sept segments, jusqu'aux écrans LCD de diverses tailles et capacités.

L'IHM a plusieurs fonctions différentes pour une utilisation industrielle, notamment les suivantes :

- ✓ Visualisation graphique complète des processus et des cas de processus,
- ✓ Signaux et reconnaissance d'événements.
- ✓ Archiver les valeurs mesurées et les messages dans des bases de données traitées.
- ✓ Enregistrement des données de processus et archivage des données.
- ✓ Gestion des utilisateurs et de leurs droits d'accès [27]

### III.9.2. Procédure de Création d'IHM de commande du système

La création de l'IHM du système de commande passe par plusieurs étapes qui seront expliquées dans ce qui va suivre.

#### Ajout d'une fenêtre

La fenêtre est ajoutée via les étapes montrées sur la figure ci-dessous :

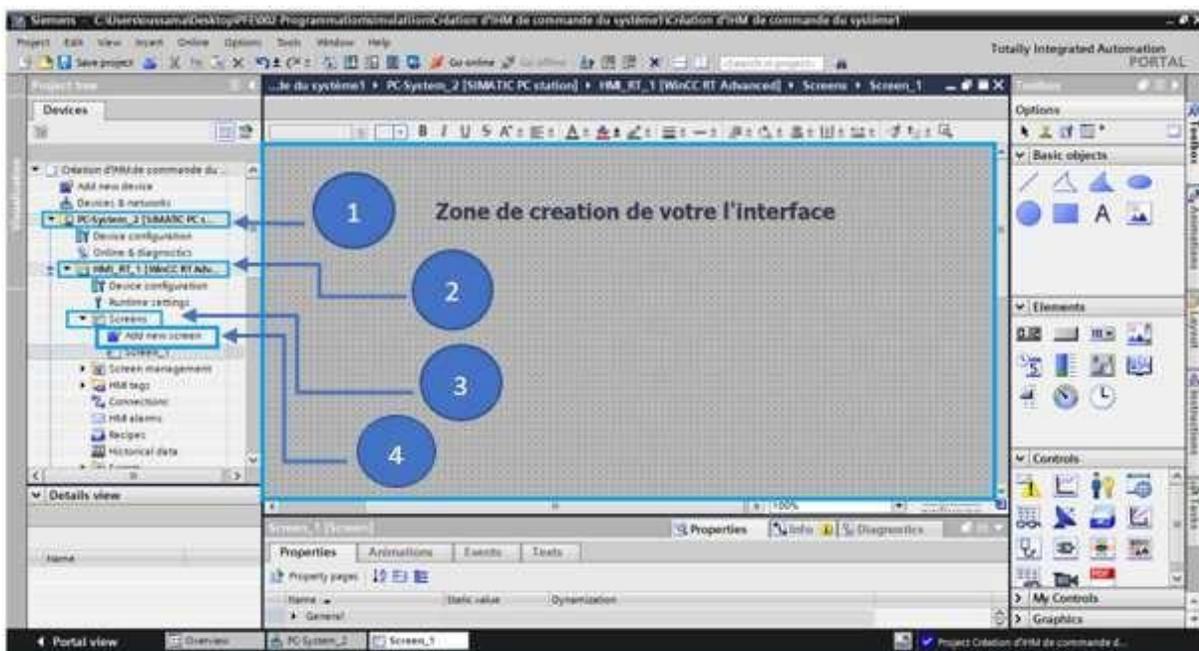


Figure III-14: Méthode d'ajout d'une vue

#### Ajout d'accessoires : Objet de base et Eléments

Les étapes représentées dans la figure ci-dessous montre comment ajouter par exemple un bouton « Activation » dans la vue IHM :

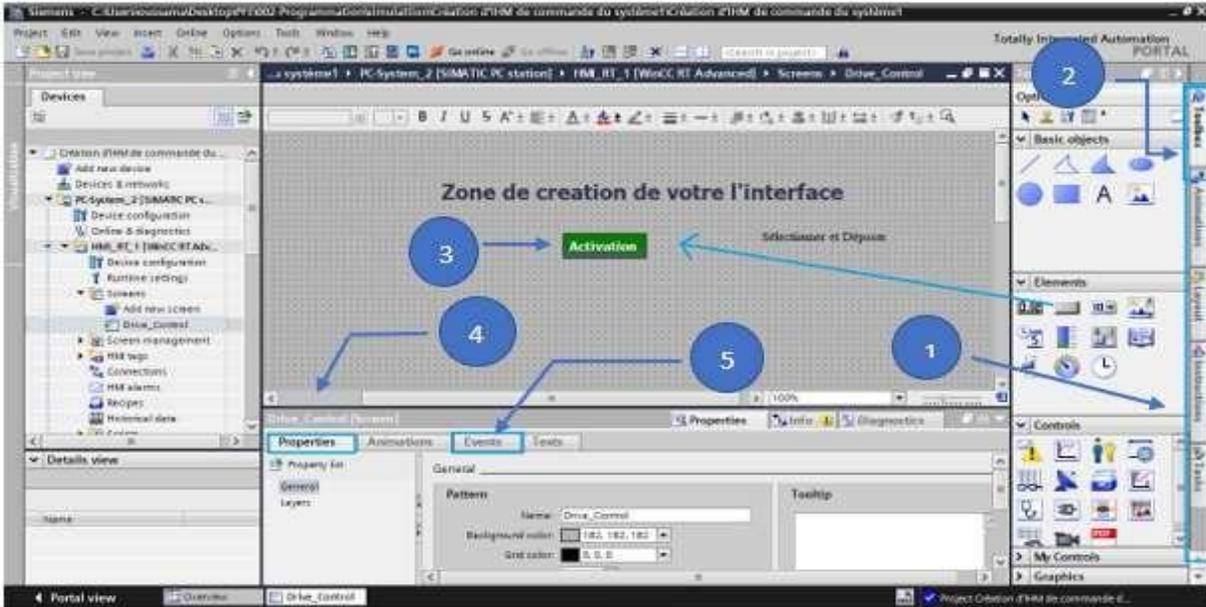


Figure III-15 : Représentation de la fenêtre d’IHM

- Cliquer sur la petite flèche pour afficher la rubrique Toolbox
- Cliquer sur Toolbox te permet d’afficher tous les objets et éléments que tu veux insérer
- Choisir l’élément voulu et déposer le dans la zone de création de votre interface
- Cliquer sur rubrique « Propriétés » pour modifier l’apparence par exemple la taille et le couleur de bouton activation.
- Cliquer sur rubrique « Events » pour insérer un évènement déclencheur, par exemple ‘Set bit’ et ‘Reset bit’ (Voir Figure ci-dessous)

En utilisant la méthode décrit précédemment pour ajouter les boutons et accessoires nécessaires au système, nous sommes arrivés à obtenir la forme finale de l’IHM qui va nous aider dans le bon contrôle du système comme le montre la figure suivante.

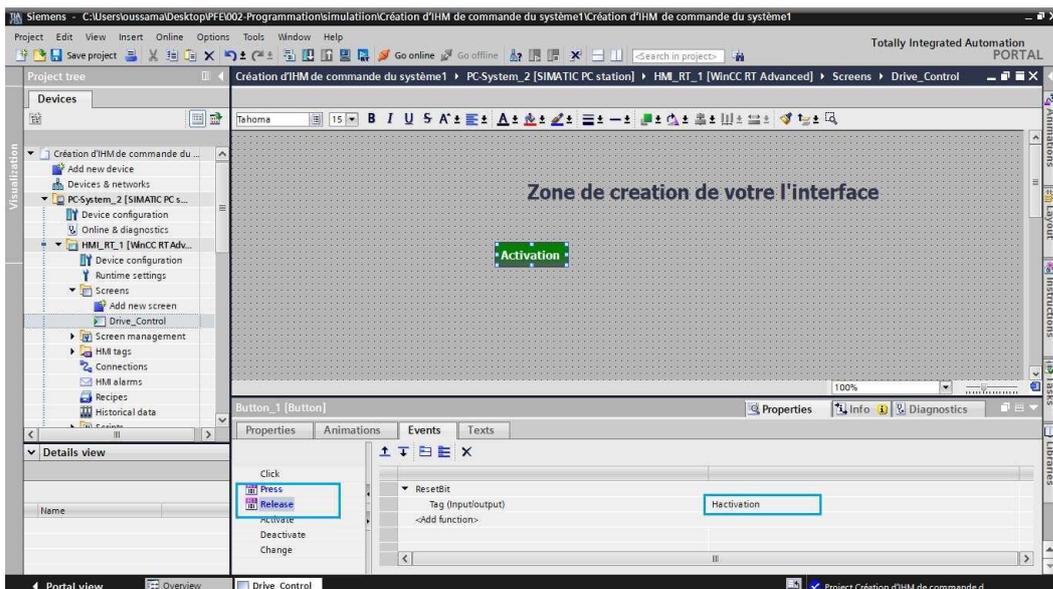
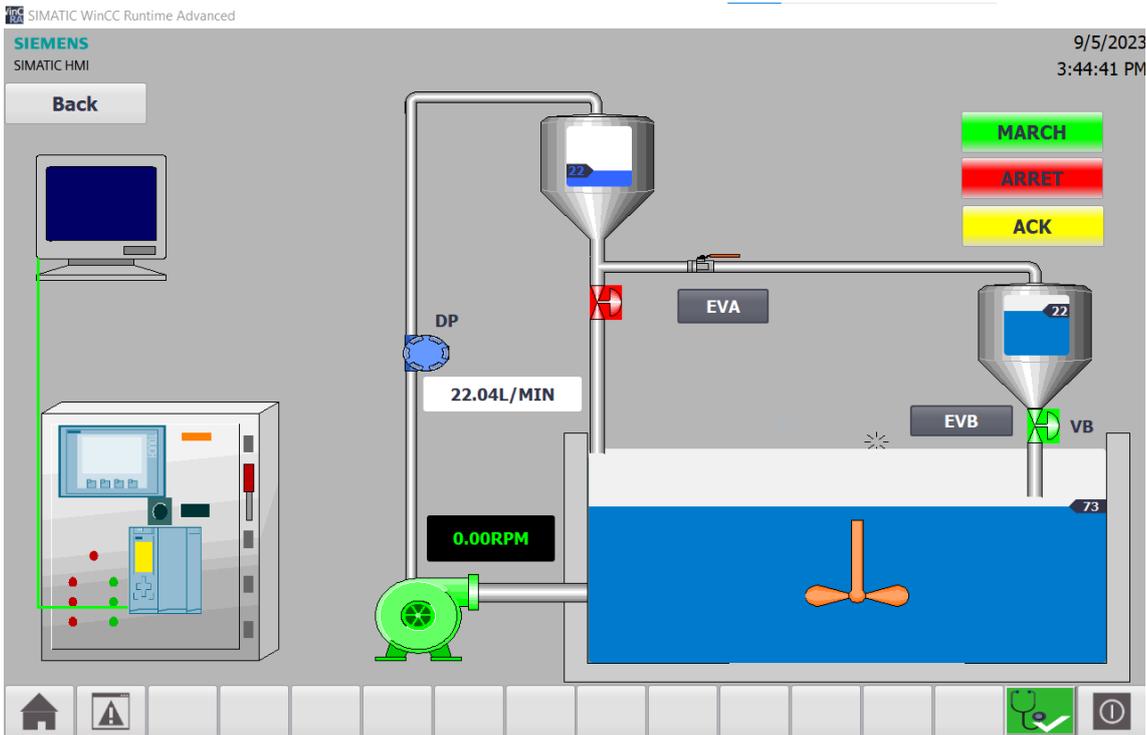


Figure III-16 : Configuration d’évènement d’un bouton



FigureIII-17 :Vue IHM de station de pompage

Description de Vue d’IHM : «Maquette Station» :

La vue d’IHM «Maquette » du Station comporte plusieurs éléments ayant les fonctionnalités suivantes :

- ✓ Boutons «**MARCH/ARRET** » : utilisés pour démarrer/arrêter respectivement le MAS
- ✓ Bouton « **ACK** » : l’action sur ce bouton efface tous les défauts dusystème au niveau du VFD
- ✓ Boutons « **EVA/EVB** » : utilisés pour activer/désactiver respectivement les électrovannes A et B
- ✓ Champ numérique au dessus de La pompe affiche la valeur de vitesse actionné par le variateur
- ✓ Champ numérique au dessus du débitmètre affiche la valeur de débit mesuré

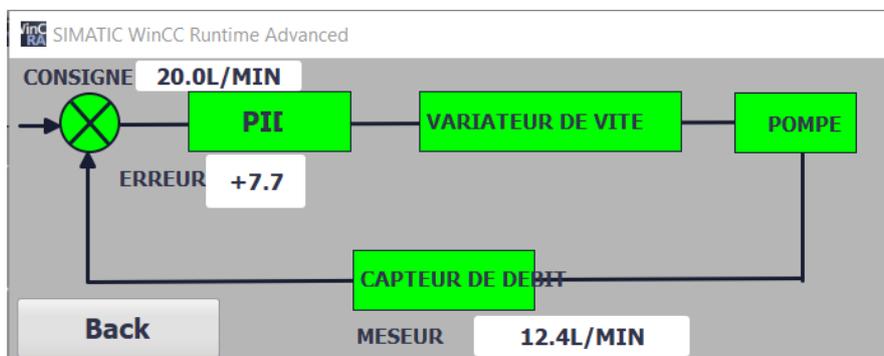
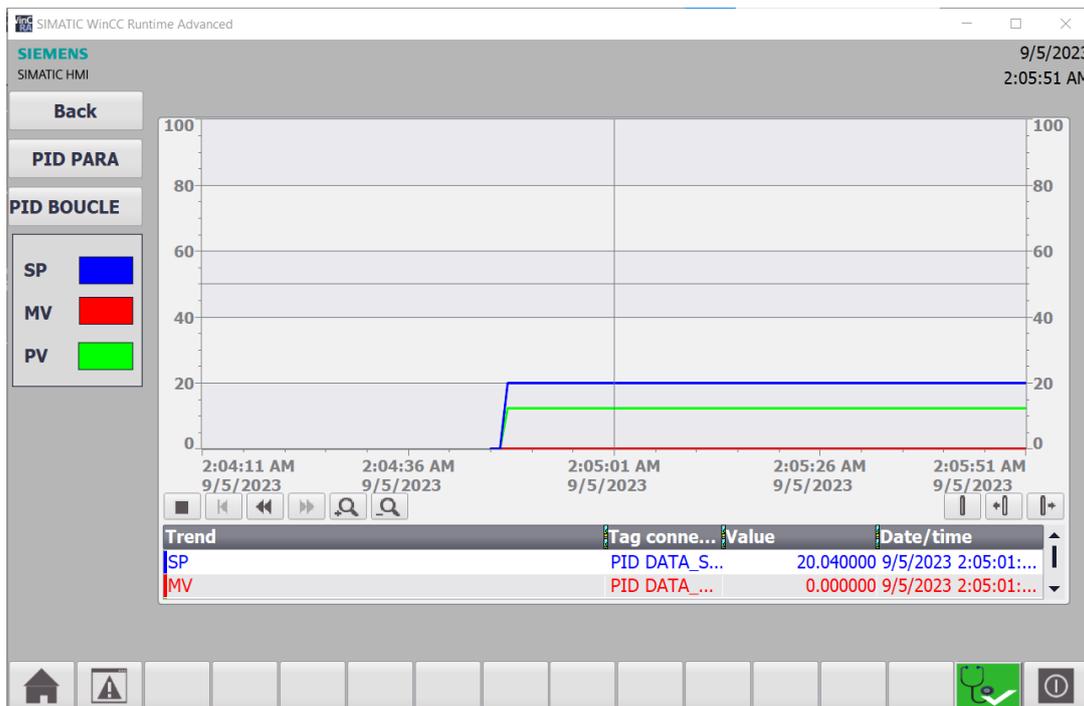


Figure III-18 :vue IHM de la boucle de regulation



FigureIII-19 :vue IHM de la courbe de regulation PID

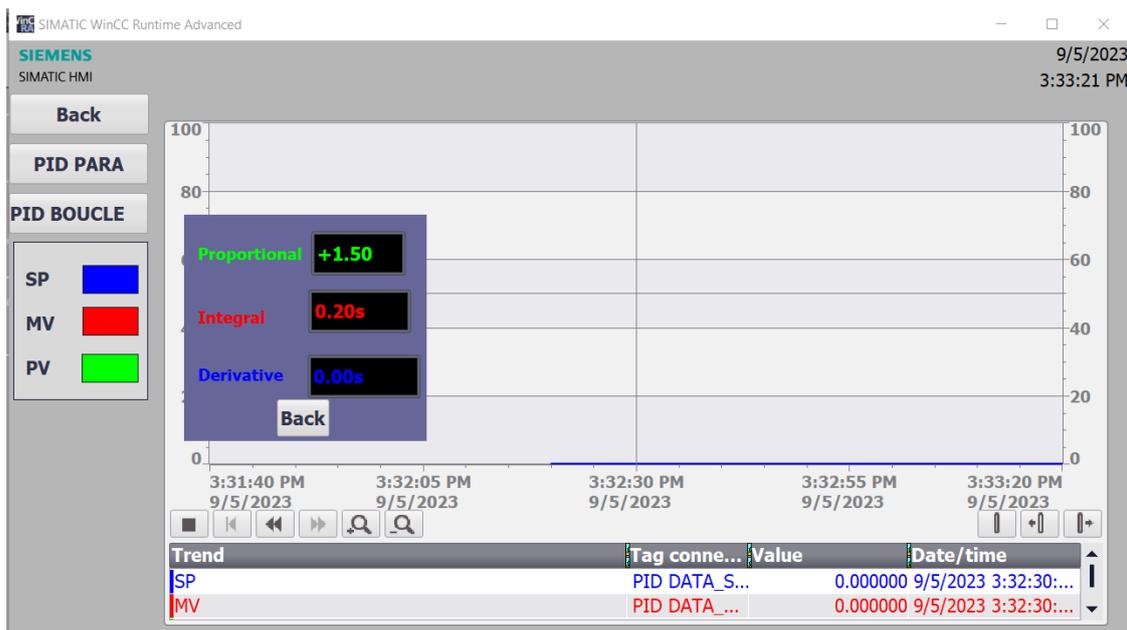


Figure III-20 :Vue IHM des parametres PID

### III.10. Principe de Régulation de débit (PID) à partir d'un variateur de vitesseG10C

on va traiter le cas de la régulation simple : la réaction (retour capteur) est dans le même sens que l'action (sortie consigne vitesse pompe).

Plus la sortie du régulateur croit, plus la vitesse de la pompe augmente, plus le débit augmente et donc

le signal de mesure augmente.

La boucle de régulation PID assurée par le variateur de vitesse comporte :

Une consigne de débit transmet au régulateur PID avec un retour mesure (débit ) via un capteur débitmètre qui fonctionne en courant 4-20mA.pour calculé une valeur de réglage transmet au variateur de vitesse ,et à la sortie de variateur une consigne de vitesse est envoyé à la pompe

Le retour de la mesure du débit sera câblé comme entrée analogique sur l’automate, il est configurée en courant 4-20 mA.et la valeur de réglage sera câblé comme sortie analogique sur l’automate pour être envoyer au variateur.

### III.11. Développement de notre programme au niveau de Step7

Après avoir réalisé notre IHM, on peut savoir par défaut quels sont les variables qu’on a besoin et quels sont les fonctionnalités qu’on doit obtenir. Pour cette raison, avant de commencer de programmer, nous vous conseillons de remplir votre table des mnémoniques (table des variables) qui sont nécessaires pour le développement de programme de contrôle. En fur et à mesure que vous avancez dans le programme, vous pouvez ajouter d’autres variables.

#### III.11.1. Tables de mnémoniques

La figure III. 21 montre une capture d’écran de la table partielle des variables obtenue à partir de TIA Portal.

STATION DE POMPAGE ▶ PLC_1 [CPU 1512C-1 PN] ▶ Variables API										
Variables										
Variables API										
	Nom	Table des variables	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	Comment...
1	flowmeter	Default tag table	Int	%IWD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	CONTROL VFD	Default tag table	*CTR WORD*	%Q 128.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	STATUS WORD	Default tag table	*STATUSWORD*	%I128.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	EVA	Default tag table	Bool	%Q4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	EVB	Default tag table	Bool	%Q4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	setpoint	Default tag table	Int	%IW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	SP	Default tag table	Int	%QW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	MV	Default tag table	Int	%QW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	<Ajouter>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure III-21: Table des variables

#### III.11.2. Architecture de notre programme

Notre programme est subdivisé en plusieurs blocs

- ✓ Trois bloc de programme principal
- ✓ Deux blocs de sous-programme
- ✓ Deux blocs de donnée

La figure ci-dessous montre les différents ‘OB’ (bloc organisationnel), les fonctions (FC) et les blocs de données (DB) utilisés pour le bon fonctionnement de notre système.

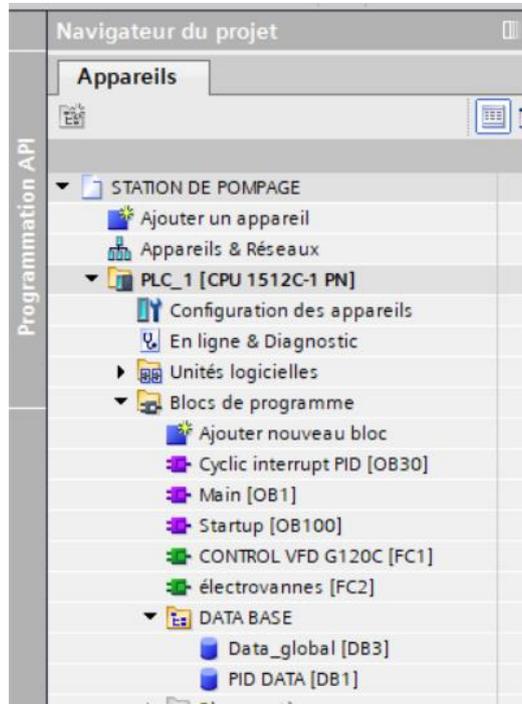
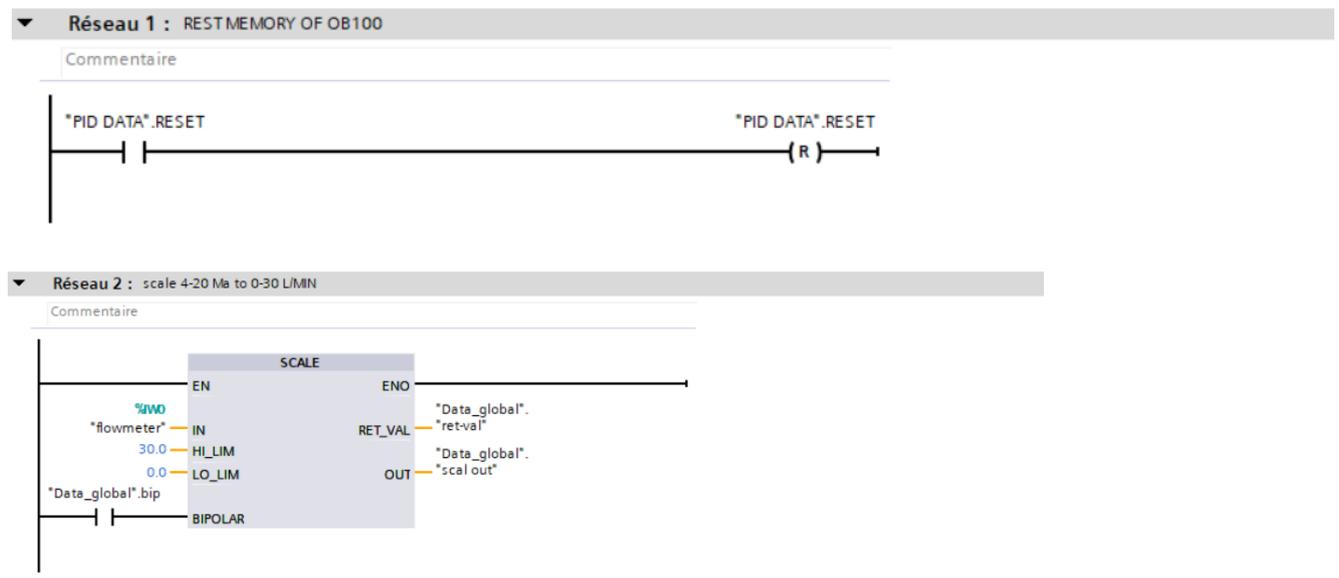


Figure III-22: Architecture de notre programme

### III.11.3. Bloc d’organisation OB1 (programme Main)

Le bloc standard OB1 est un bloc principal qui fait appel à tous les sous-programmes pour qu’ils soient exécuté cycliquement. Au niveau d’OB1, nous avons déposé les 05 fonctions (Figure ci-dessous) pour être appelées et exécutées cycliquement



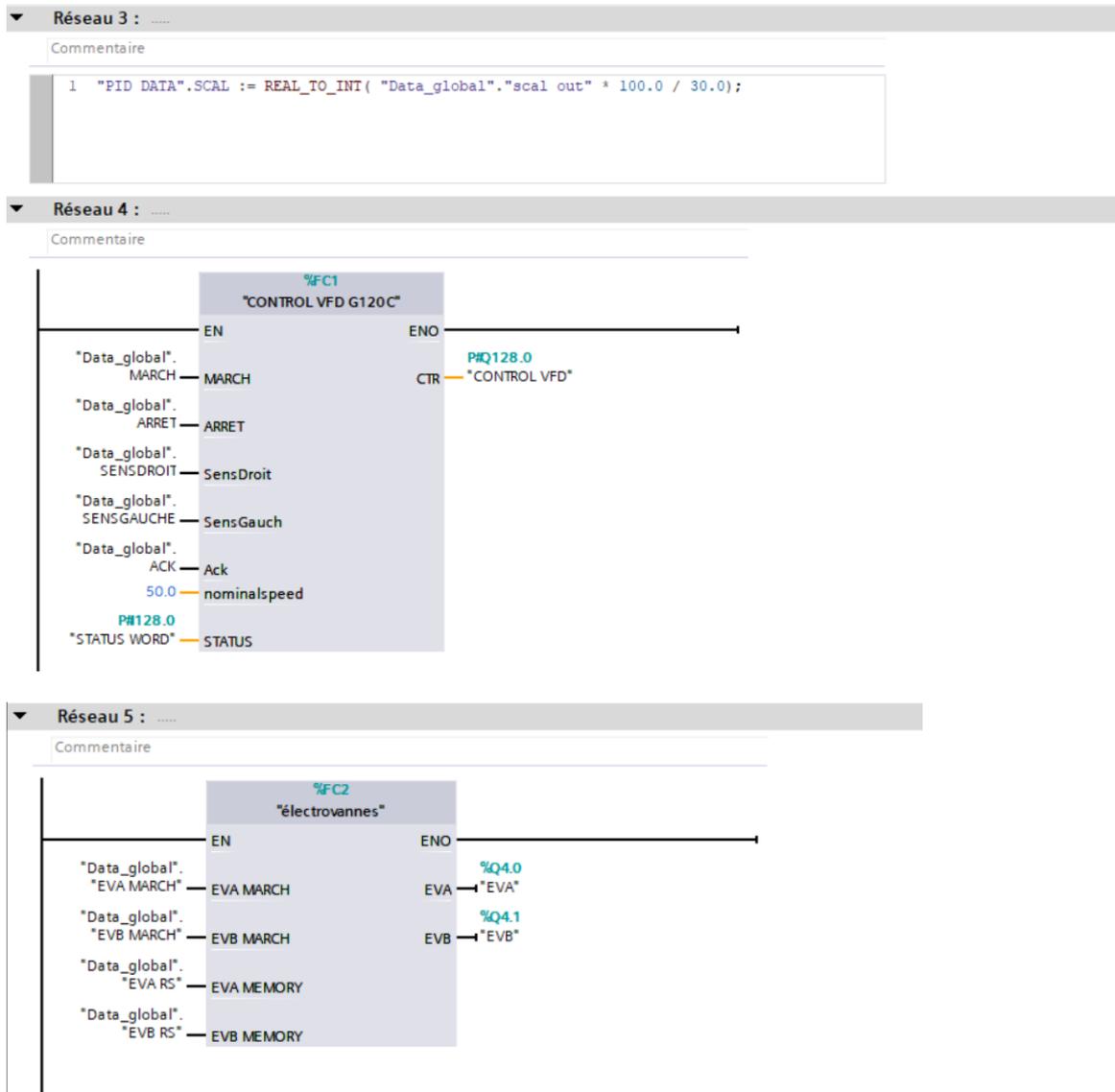


Figure III-23 : Les réseaux de Bloc d’organisation principale OB1

### III.11.4. Bloc d’organisation OB30 (Cyclic interrupt )

PID\_Compact met trois valeurs de réglage à disposition. La valeur de réglage que vous utilisez dépend de votre actionneur.

#### **Output PER :**

L'actionneur est adressé via une sortie analogique et est commandé à l'aide d'un signal continu, par exemple 0...10 V, 4...20 mA.

#### **Output :**

La valeur de réglage doit être mise en forme dans le programme utilisateur, par ex. parce que l'actionneur présente un comportement non linéaire.

#### **Output PWM :**

L'actionneur est commandé par une sortie TOR. Des temps d'activation et de désactivation variables sont formés à partir d'une modulation de largeur.

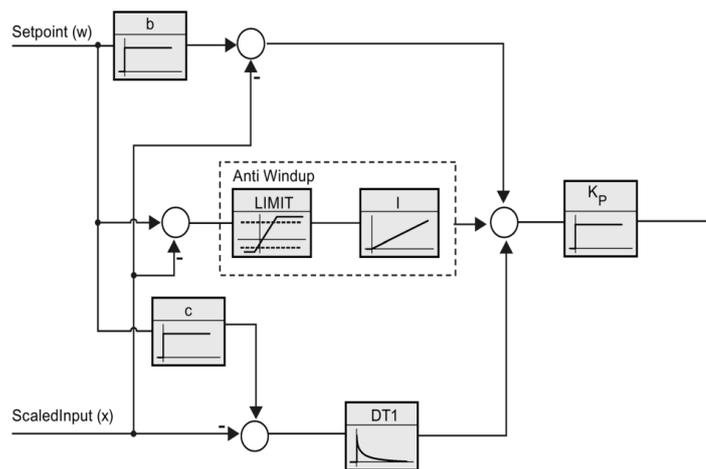


Figure III-24 : Schéma fonctionnel du PID Compact

Le bloc PID Compact est représenté dans la figure ci-dessous en exploitant les valeurs de réglages (Input/Output) :

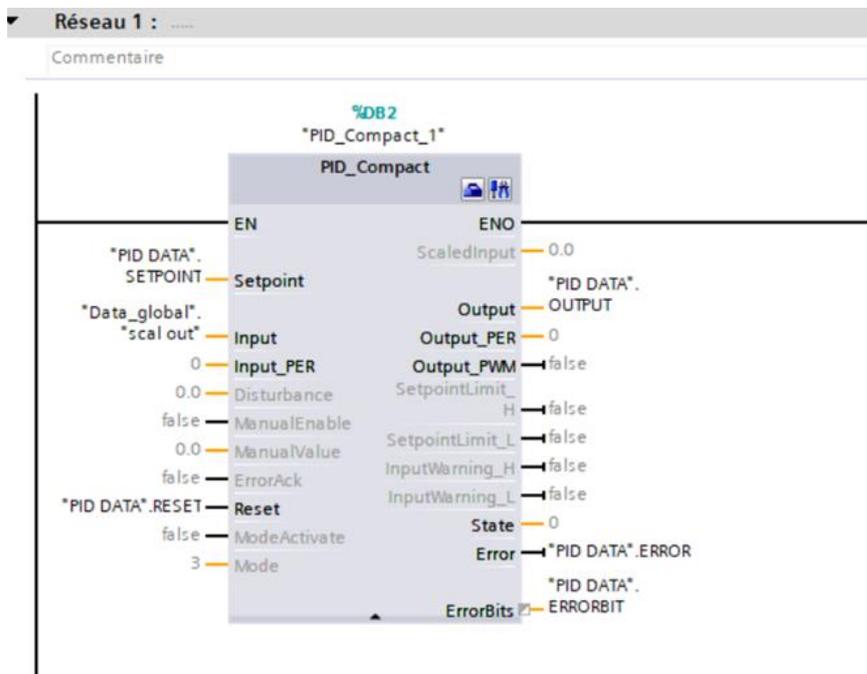


Figure III-25 : Le bloc de PID Compact

Le réseau 02 programme l'équation d'erreur par soustraction en utilisant le bloc SUB(Figure III-26)



Figure III-26 :Réseau 02 du OB 30 qui représente la valeur d’Erreur de la boucle

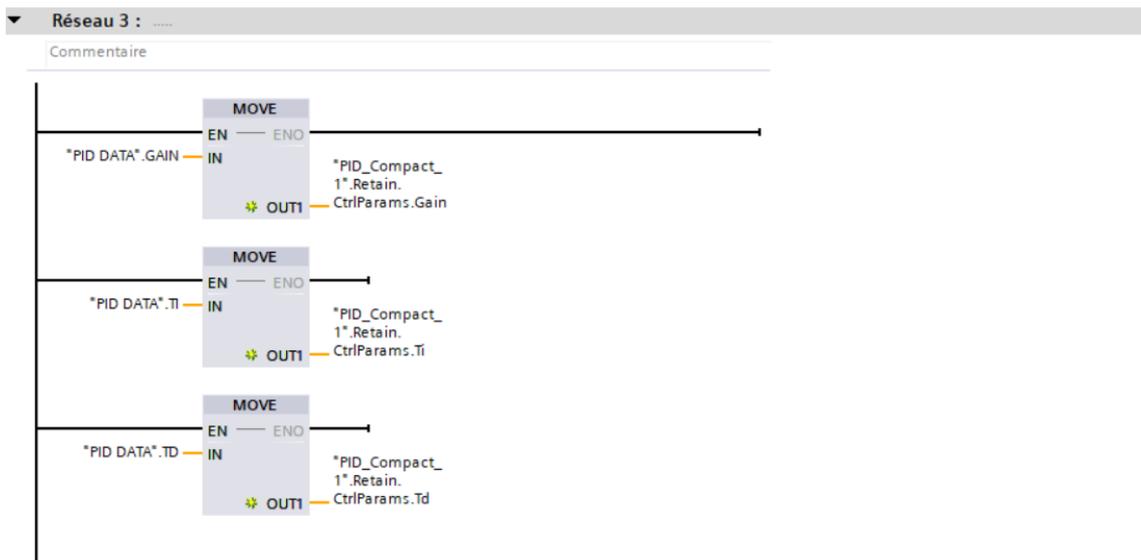


Figure III-27 :Réseau 03 du cyclic interrupt avec des blocs MOVE pour transferer les variable en IHM

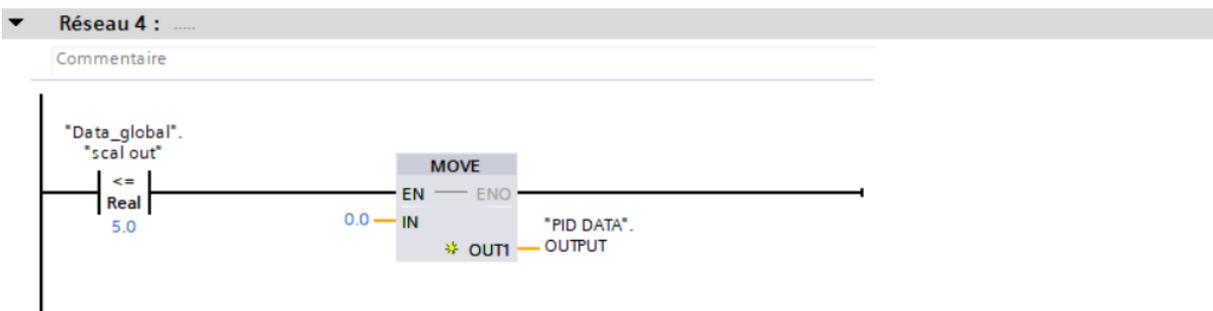


Figure III-28 :Réseau 04 qui représente l’action d’arrêt du moteur quand le debit est moins de 5 L/MIN

Le réseau 04 est une consigne de switch pour déclencher le moteur si le debit tombe sous 5L/min est transferer vers le variateur pour l’arrêter illustré au dessus (FigureIII-28) .

### III.11.4.Bloc d’organisation OB100(Startup)

c'est le plus couramment utilisé,il est compatible avec tous les types de processeurs,seules les

données rémanentes sont conservées. Les variables qui ont été configurées comme rémanentes conservent donc leur dernière valeur, les autres variables ou blocs non rémanentes sont réinitialisées. Un OB100 est généralement utilisé pour initialiser des variables ou modules d'E/S au démarrage du programme.

Le réseau –Reset The PID- représente l'acquittement des paramètres PID de l'automate illustré dans la Figure III-29

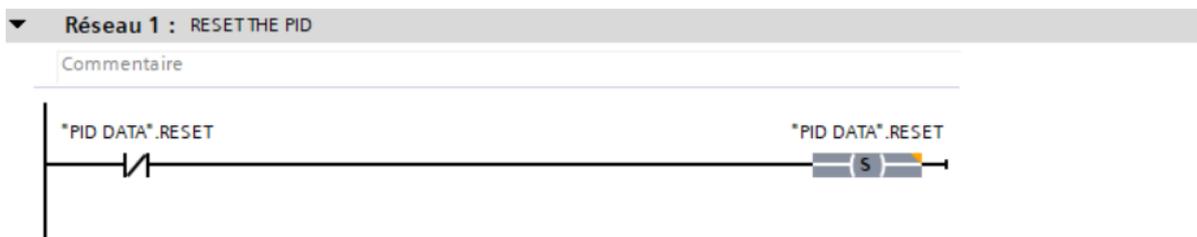


Figure III-29 : Le réseau de bloc OB100 pour rendre le PID par défaut

### III.11.5. Présentation des fonctions FC

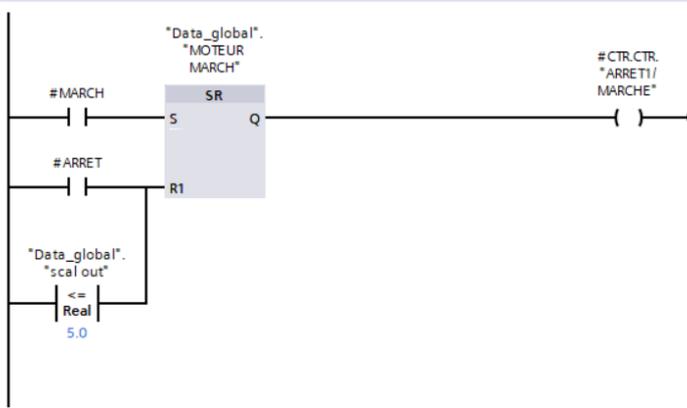
Une fonction FC est un bloc de code sans mémoire qui doit être appelé depuis un autre bloc de programme principal pour être exécuté depuis un OB1. Il doit être chargé dans l'automate avant d'être appelé.

Dans notre programme, il existe 02 fonctions :

- Control VFD G120C [FC1]
- Électrovannes [FC2]

### III.11.6. Fonction : Control VFD G120C [FC1]

Tout d'abord, pour activer notre système on appuie sur le bouton « MARCH » .A l'action, le programme envoie un signal pour activer le VFD et fournit l'état logique '1' pour l'opérande, Avec le même principe on choisit le sens de rotation en Réseau 2 comme c'est illustré ci-dessous :



FigureIII-30 :Réseau 01 pour le demmorage et l’arret du moteur avec le bloc SR (SET/RESET)

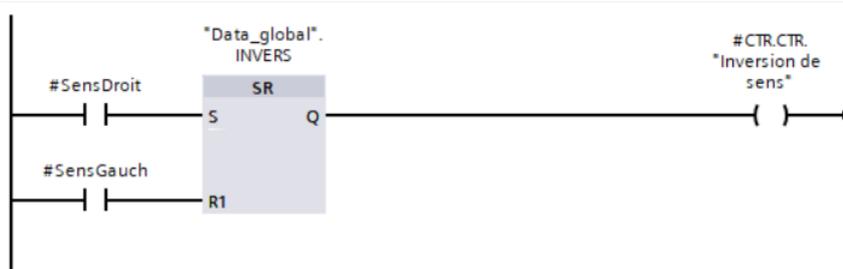


Figure III-31 : Réseau 02 pour inverser le sens de rotation du moteur



Figure III-32 : Réseau 03 Pour acquitter les defaults au variateur

```

1      #CTR.CTR.SetVilocity := REAL_TO_INT(#nominalspeed * 16384.0 / "PID DATA".OUTPUT);
2
3
4

```

Figure III-33 : Réseau 04 d’équation de signal de sortie en langage SCL

La figure si dessus represente le programme signale de sortie vitesse par une équation en langage SCL .

### III.11.7. Fonction : Les électrovannes [FC2]

Le programme du bloc fonction [FC2] «Les électrovannes » se compose de 02 réseaux, illustrés à la Figure III-34 :

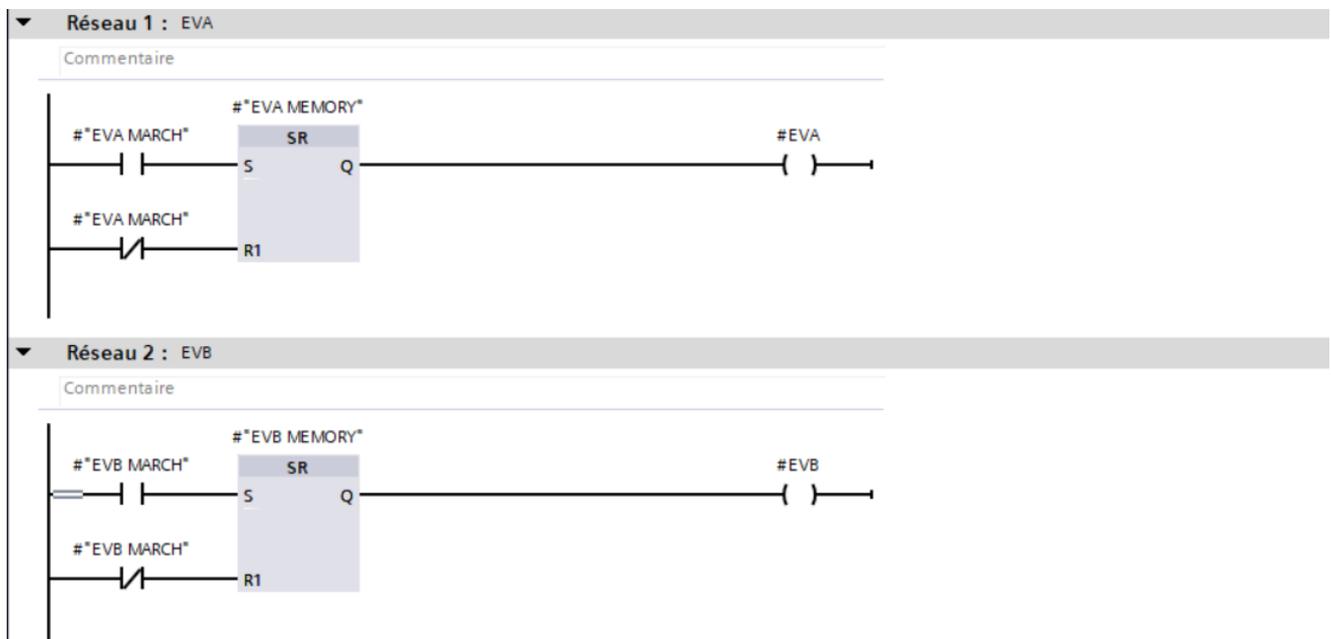


Figure III-34 :Les fonctions de programmation des electrovanes manuelles.

### III.11.8. Blocs de données DB (Data Bloc)

Le bloc de données joue un rôle essentiel en enregistrant des variables et des constantes utilisées dans notre programme. Ce type de base de données présente plusieurs avantages, notamment la facilité de modification ou d'initialisation de la valeur de n'importe quelle variable, contrairement à une table de mnémoniques où cela n'est pas possible. Un autre avantage est que lors de l'ajout d'un paramètre, l'adressage se fait automatiquement, éliminant ainsi tout risque de chevauchement. De plus, les données sont organisées de manière structurée, ce qui facilite leur repérage dans le programme. La figure ci-dessous illustre la base de données créée pour commander notre variateur de vitesse à l'aide de notre IHM via notre automate.

### III.11.9. Blocs de données DB3 (Data Global)

Pour modifier les variables dans la fenêtre de programmation, il faut faire un clic droit sur la variable.

Data_global										
	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ..	Surveilla...	Comment...
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	bip	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	ret-val	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4	scal out	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
5	ENABL	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	MOTEUR MARCH	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
7	INVERS	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	STATUS	*STATUSWORD*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
9	CTR	*CTR WORD*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	MARCH	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	ARRET	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
12	SENSDROIT	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
13	SENSGAUCHE	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
14	ACK	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
15	MODE AUTO	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
16	MODE MAN	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	COMMITATEUR	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
18	VALIDATION	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
19	ACK MODE	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
20	ERR MODE	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
21	EVA MARCH	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
22	EVB MARCH	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
23	EVA RS	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
24	EVB RS	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Figure III-35 :La base de doonnées de notre programme

### III.11.10. Blocs de données DB1 (PID Data )

Les donnés représentés au dessous au FigureIII-36 sont les paramètres de régulation de notre système comprenant les différents signaux , les paramètres du gain , le temps d’intégration et le temps de dérivation

PID DATA										
	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ..	Surveilla...	Comment...
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	SETPPOINT	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	INPUT	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4	OUTPUT	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
5	RESET	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	ERRORBIT	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
7	ERROR	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	err	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
9	SCAL	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
10	GAIN	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
11	TI	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
12	TD	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Figure III-36 : Les blocs de données du PID

### III.12. Commande du variateur à travers le Télégramme

Le réseau (Figure III. 37) représente les différents bits du télégramme 01 quifonctionne comme suit :

Bit	Adresse	Signification	Explication
0	%Q257.0	Marche_Arrêt1	Le variateur passe à l'état « Prêt à fonctionner » Si, en outre, le bit 3 = 1, le variateur met le moteur en marche
1	%Q257.1	Non_Arrêt2	La mise en marche du moteur (ordre de Marche) est possible
2	%Q257.2	Non_Arrêt3	La mise en marche du moteur (ordre de Marche) est possible
3	%Q257.3	Déblocage-fonctionnement	Mettre le moteur sous tension (déblocage des impulsions possible)
4	%Q257.4	Non_Blocage_Générateur	Le déblocage du générateur de rampe est possible
5	%Q257.5	Déblocage_Générateur	La sortie du générateur de rampe suit la consigne
6	%Q257.6	Déblocage_Consigne	Le moteur accélère jusqu'à la consigne avec le temps de montée
7	%Q257.7	Acquitter_Défaut	Acquitter le défaut. Si l'ordre ON est encore présent, le variateur passe à l'état "Blocage d'enclenchement"
10	%Q256.2	Pilotage par API	Commande via bus de terrain, le variateur reprend les données process du bus de terrain

**Tableau III-2 :** Représentation des bits du télégramme commandant notre variateur

En appuyant sur le bouton « MARCH » au niveau de notre IHM, tous les bits du Télégramme sont mis à « '1' » ce qui active notre variateur.

Vu que ces bits sont des sorties physiques du notre automate (Q 128.0 à Q 129.7 ), elles doivent être déclarées au niveau de la table de mnémoniques (figure ci-dessous).

Default tag table									
	Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibl...	Surveilla...	
1	flowmeter	Int	%IW0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	CONTROL VFD	"CTR WORD"	%Q128.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	CTR	Struct	%Q128.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	8 Réservé	Bool	%Q128.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	9 Réservé	Bool	%Q128.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Pilotage par API	Bool	%Q128.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Inversion de sens	Bool	%Q128.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	12 Non utilisé	Bool	%Q128.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	PotMot Augmenter	Bool	%Q128.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	PotMot Diminuer	Bool	%Q128.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	15 Réservé	Bool	%Q128.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12	ARRETI/MARCHE	Bool	%Q129.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13	pas d'ARRET2	Bool	%Q129.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
14	pas arrêt rapide (ARRET3)	Bool	%Q129.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	Débloquer le fonctionne...	Bool	%Q129.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	Ne pas bloquer le génér...	Bool	%Q129.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
17	Débloquer le générateur...	Bool	%Q129.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
18	Débloquer la consigne	Bool	%Q129.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
19	Acquitter les défauts	Bool	%Q129.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
20	SetVilocity	Int	%QW130		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure III-37: Capture d'écran de la table de mnémoniques comportant les variables du télégramme 1

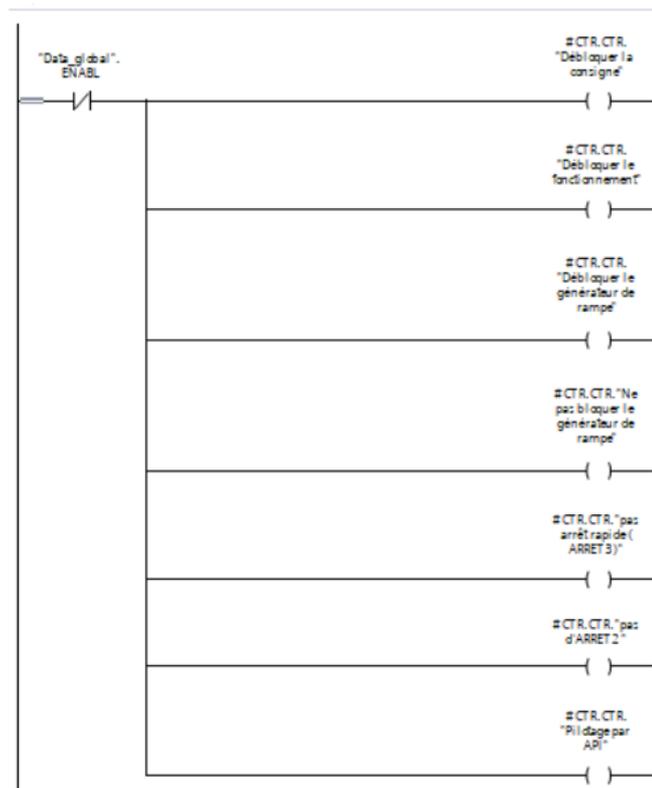


Figure III-38 : Activation/Désactivation des bits du télégramme 1 pour la commande du variateur

### III.13. Enregistrer et compiler le programme[28]

Pour enregistrer votre projet, sélectionnez le bouton Save project dans le menu. Pour compiler tous les blocs, cliquez sur le dossier "Programm blocks" (Blocs de programme) et sélectionnez l'icône Compile (Compiler) dans le menu. ( Save project  Programm blocks (Blocs de programme)  )

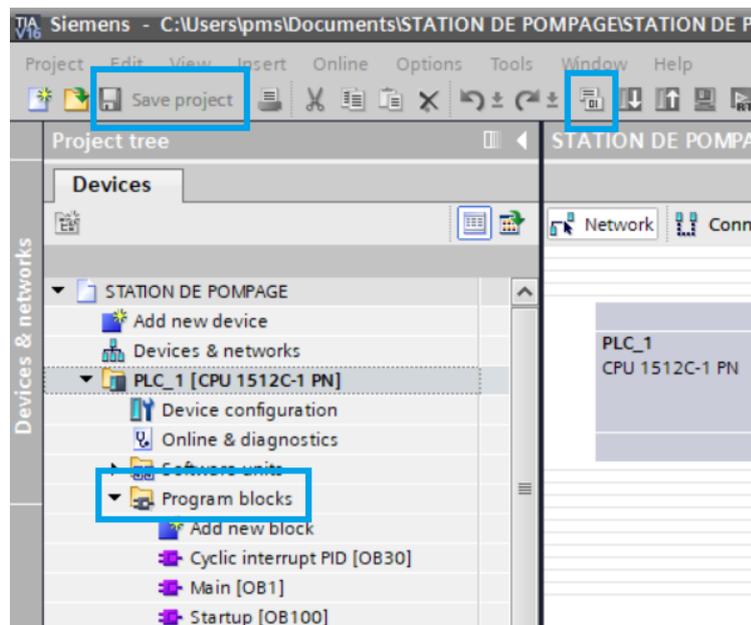


Figure III-39 :Les étapes de compilation

Les blocs correctement compilés sont ensuite représentés dans la zone "Compile" de l'onglet "Info".

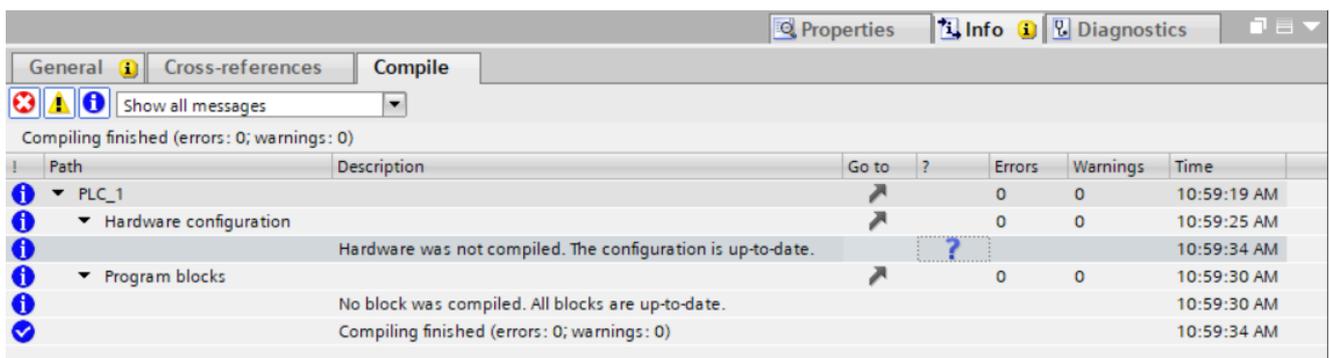
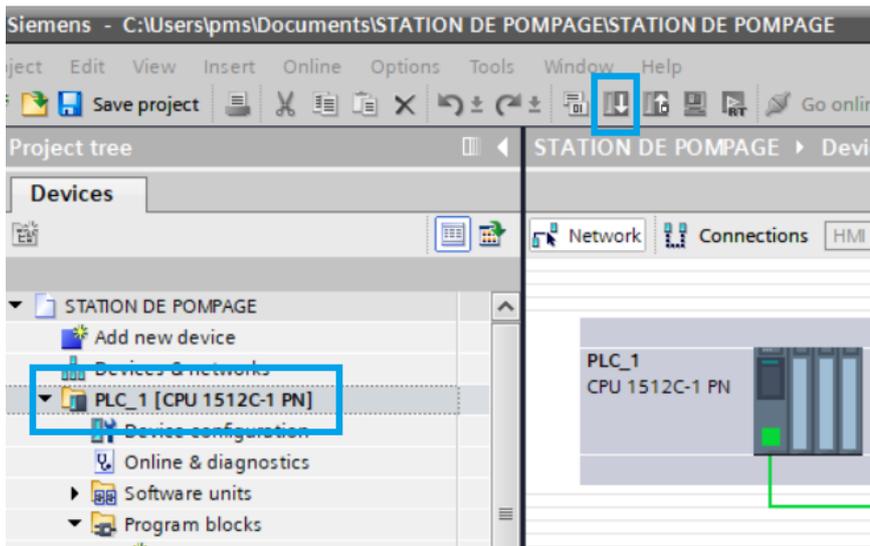


Figure III-40 : Signalisation des erreurs de compilation

### III.14. Charger le programme[28]

Une fois que la compilation s'est correctement déroulée, vous pouvez charger toute la commande avec le programme créé, comme cela a été décrit dans les modules sur la configuration matérielle (□ ).



FigureIII-41 :Le chargement du programme

### III.15. Visualiser des blocs de programme[28]

Pour visualiser le programme chargé, le bloc souhaité doit être ouvert. Vous pouvez ensuite désactiver/activer la visualisation en cliquant sur l'icône . ( PLC1 ).

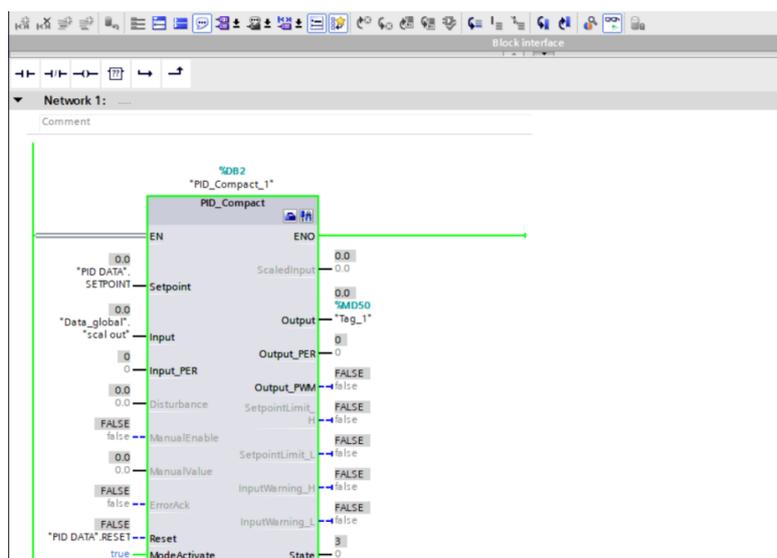


Figure III-42 :Le bloc PID Compact après la visualisation

**Remarque** : ici, la visualisation s'effectue sur la base des signaux et elle dépend de la commande. Les états logiques des opérandes sont représentés par TRUE ou FALSE.

### III.16.Démarrage de simulation[29]

Pour démarrer S7-PLCSIM depuis STEP 7, sélectionnez votre CPU depuis le navigateur du projet et choisissez l'une des méthodes suivantes :

- Cliquez sur le bouton "Démarrer la simulation"  dans la barre d'outils principale.
- Cliquez avec le bouton droit de la souris sur l'appareil dans le navigateur du projet, puis sélectionnez "Démarrer la simulation" dans le menu contextuel.
- STEP 7 charge le projet dans S7-PLCSIM. S7-PLCSIM s'ouvre en Vue compacte par défaut sans projet ouvert.

### III.16.1. Vue compacte

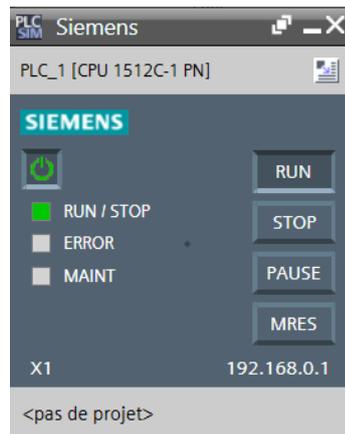


Figure III-43 : Fenêtre de la vue compacte

La vue compacte est constituée d'une petite fenêtre comportant un nombre restreint de commandes et de fonctionnalités illustre ci-dessus

Cette vue est utile si vous souhaitez corriger les erreurs de votre programme dans STEP 7 plutôt que dans la vue de projet S7-PLCSIM.

S7-PLCSIM s'ouvre par défaut dans la Vue compacte. Si vous souhaitez faire de la Vue du projet la vue par défaut, vous pouvez effectuer cette modification dans le menu principal de la Vue du projet, dans Options > Paramètres. [29]

### III.16.2. Vue du projet

La vue du projet contient la fonctionnalité complète de S7-PLCSIM. La Vue du projet offre le même aspect et la même convivialité que l'interface utilisateur de TIA Portal.

La vue du projet consiste en plusieurs éléments :

- Menu principal et barre d'outils principale
- Options et paramètres (accessible à partir du menu principal)
- Navigateur du projet
- Vue des appareils
- Éditeur de table SIM
- Éditeur de séquence
- Éditeur d'événement
- Outils en ligne, panneau de commande et paramètres de commande de scan inclus

La vue du projet est utile si vous préférez corriger les erreurs de votre programme en utilisant la pleine fonctionnalité de S7-PLCSIM comme il est illustré ci-dessous (FigureIII-44) :

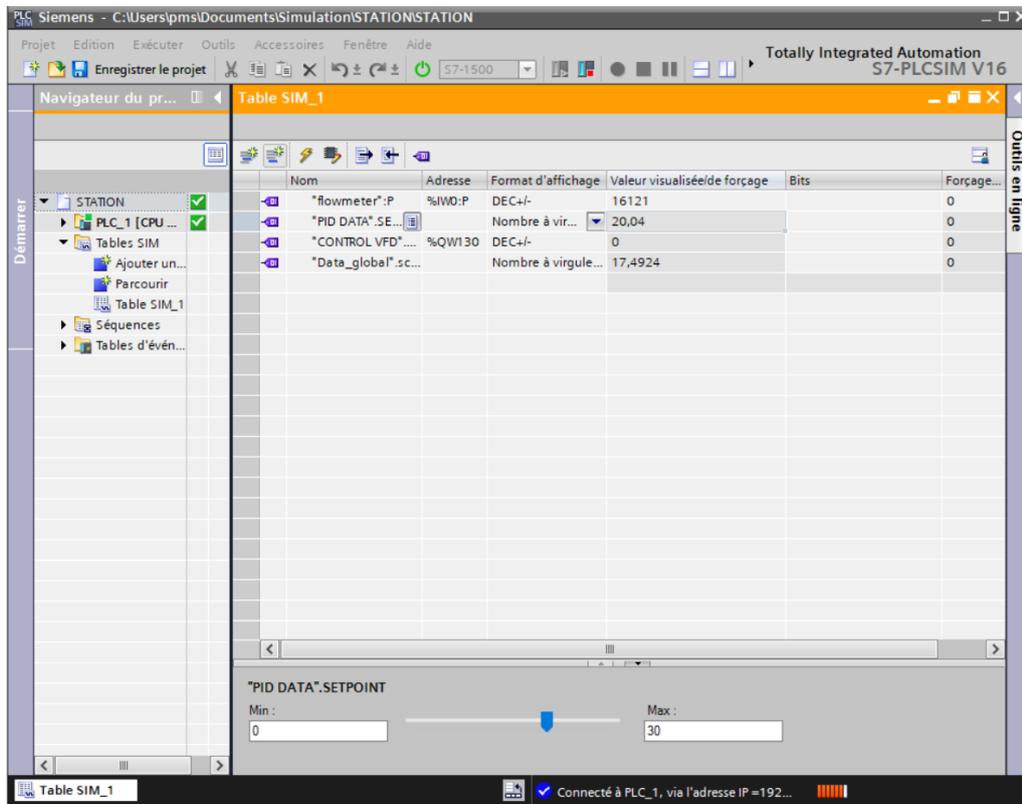


Figure III-44 :Fenetre de la vue du projet

Les variables utilise au simulation sont :

- "flowmeter" qui correspond au signal d'entrè en decimale cepandant que "Data\_global".scal out, le represente en valeur analogique en L/MIN (La courbe verte en PID)
- "PID DATA".SETPOINT ,qui correspond au signal de consigne(la courbe blue)
- "CONTROL VFD".CTR.SetVilocity,correspond au signal de sortie du variateur qui varie le regime du moteur du 0 a 100 RPM .

### III.17.Reglage des paramètre PID

Pour augmenter la performance du notre regulation ,il faut bien regler les parametres PID du Gain,temps d'integration et temps de derivation .

Donc on a choisi la methode appliqué «PID mixte » du ziegler nichols illustré ci-dessous :

**Méthode de Ziegler-Nichols**

Type de correcteur	KP	Ti	Td
PID (mixte)	0,6 x Kc	To / 2	To / 8
PID « moins nerveux »	0,3 x Kc	To / 2	To / 6

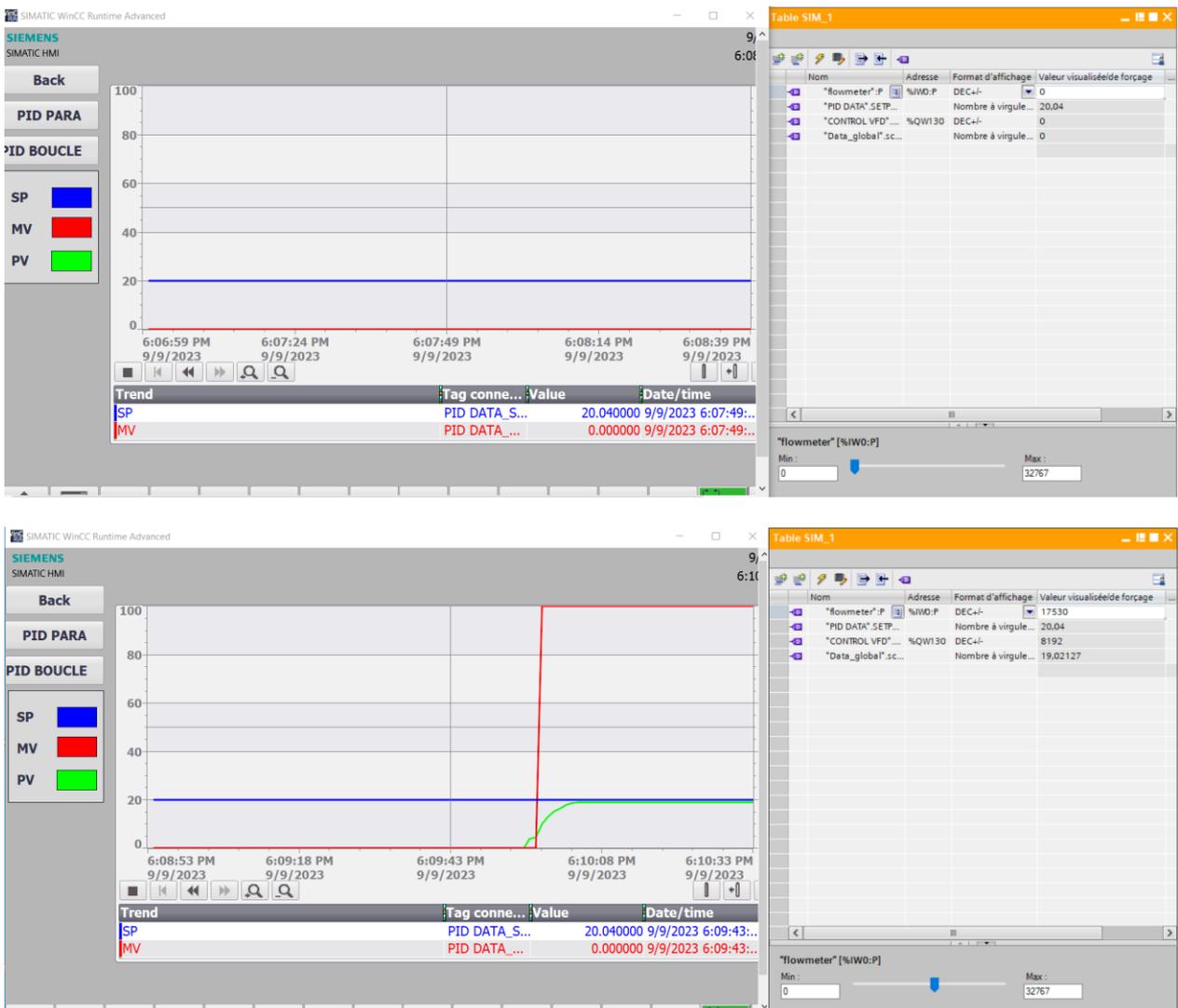
**Gain Kc = 15**  
**Période d'oscillation To = 5 s**

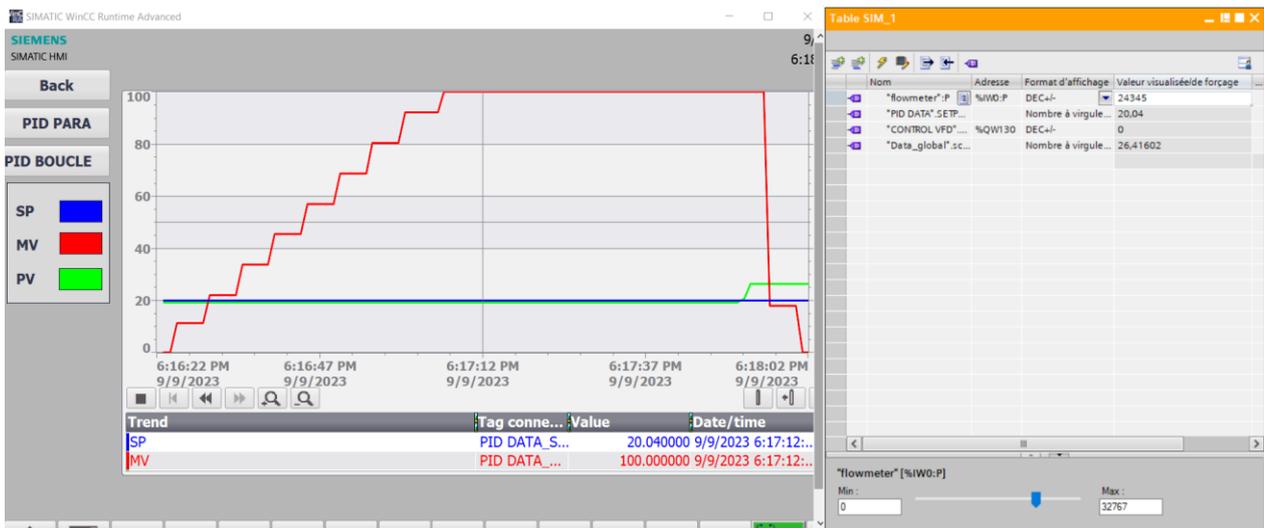
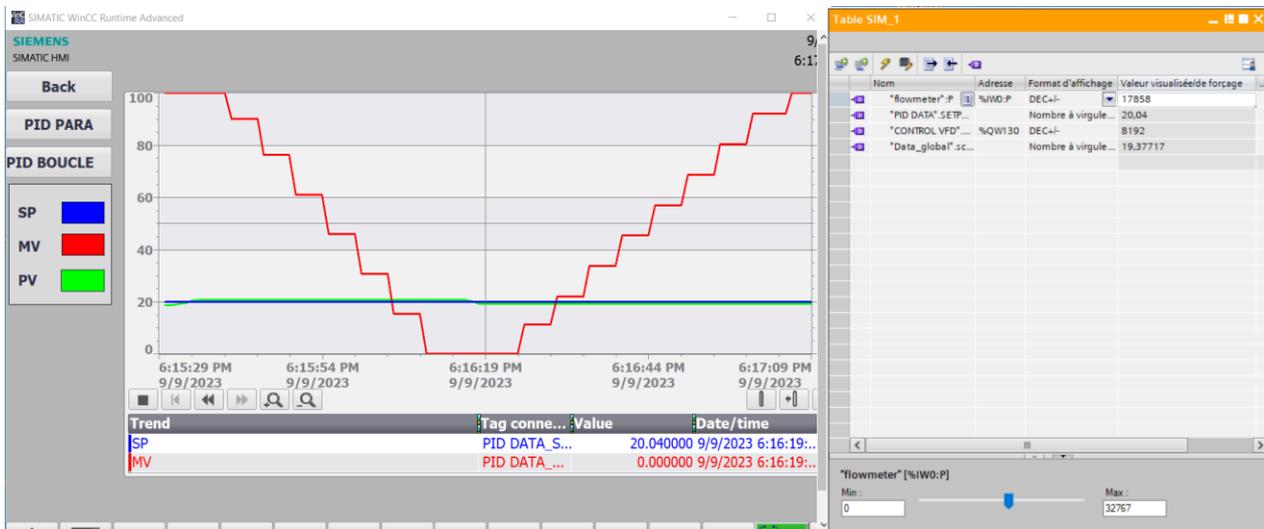
PID (mixte)	9	2,5	0,6
-------------	---	-----	-----

Figure III-45 : Méthode Ziegler Nichols

- On prend comme consigne : 20 L/Mn et on fait varié la valeur de mesure à partir de logiciel PLCSIM qui simule le fait de la régulation sur le de débitmètre.

On ouvre la vue du projet de PLCSIM simultanément avec la vue IHM de PID figuré ci-dessous :





### III.18. Analyse des résultats

En variant la mesure d'entrée on observe que si l'erreur est positif (l'input est sous le setpoint ) le variateur augmente la vitesse de la pompe au regime maximale ,et si elle est negatif le VFD fait baisser la vitesse .

Cependant , le temps de réponse du output est lié au taux d'erreur entre l'entrée et La consigne ,ca veut dire si l'erreur est relativement grande la sortie se varie brutalement ,au contraire si l'erreur est prète du consigne la sortie se varie progressivement pour reguler le signal d'entrée.

#### **Conclusion:**

Pour la programmation des systèmes d'automatisations, la firme Siemens offre une immense technologie des logiciels SIMATIC pour commander, superviser et maintenir les installations. Dans ce chapitre je me suis intéressé a la configuration et la commande de variateur de vitesse et la régulation de débit, la richesse de la bibliothèque de blocs fonction de logiciel de configuration TIA PORTAL ma permet facilement de programmer la commande.

Cette automatisation devrait permettre des tests mais le manque des matériels présent l'inconvénient majeur pour la réalisation pratique de cette application.

# Conclusion Générale

Pendant la réalisation de ce projet, j'ai élaboré un plan d'action visant à mettre en place un variateur de vitesse avec une régulation de débit dans notre conception intitulée "Maquette de station de pompage basée sur l'API SIEMENS". Cette maquette englobe la plupart des systèmes que l'on trouve dans une station de pompage réelle, que ce soit du point de vue matériel ou logiciel. Nous avons opté pour une solution basée sur un automate SIEMENS pour gérer le contrôle du variateur, la gestion des électrovannes et la régulation du débit. La communication entre les principaux éléments se fait via le réseau industriel PROFINET.

Pour mieux aborder la programmation des variateurs de vitesse, les premières étapes ont consisté à présenter la gamme des variateurs de vitesse SINAMICS G120C de SIEMENS. J'ai ainsi pu comprendre leur architecture, leur fonctionnement, ainsi que leur environnement avec les différentes fonctions associées. Ensuite, j'ai entrepris une étude du protocole de communication PROFINET PN, ce qui m'a permis de maîtriser les trames et les formats de communication. J'ai également décrit le logiciel TIA Portal V16.

Cette étude m'a offert la possibilité de paramétrer et de programmer le variateur à distance, tout en permettant une maintenance sur place. La capacité d'agir sur les différents bits de communication du TELEGRAMME-1 a facilité un contrôle précis de la chaîne de régulation. De plus, nous avons élaboré la programmation de l'API Siemens S7-1500 avec le logiciel TIA Portal.

Enfin, j'ai conçu une interface de supervision conviviale et facile à assimiler. Cette supervision de la maquette a été développée grâce au simulateur d'automate WINCC flexible.

La réalisation de ce projet m'a demandé de mobiliser toutes mes connaissances et compétences déjà acquises, tout en me confrontant aux défis du travail quotidien et en prenant des initiatives personnelles. Ce projet de fin d'études a été particulièrement enrichissant, car il m'a permis de découvrir le monde professionnel et de transformer les connaissances de base acquises à l'école en compétences professionnelles.

L'étude que j'ai menée n'est que préliminaire, nous recommandons que d'autres études soient menées dans l'axe et qui englobent les thèmes suivants :

- Intégration et mise en service d'un variateur de vitesse SINAMICS G120C avec logiciel STARTDRIVE de SIEMENS.
- Réalisation de cartes de communication de certain protocole ainsi que des interfaces passerelles entre différentes protocoles.
- Mettre en œuvre des commandes avancées au niveau des variateurs de vitesse, tel que la commande vectorielle, et au niveau de l'automate (commandes robustes, commande adaptative).

Encouragement à ouvrir des portes vers le monde industriel à travers des projets de terrain de différentes installations existantes dans des usines étatiques et privées.



## Reference bibliographique et Références web graphiques

- [1] Schneider Electric, «Chapitre 5, Départs moteur »
- [2] [www.wikipedia.com](http://www.wikipedia.com)
- [3] LOUIS J., MULTON B., BONNASSIEUX Y. et LAVABRE M. «
- [4] Schneider Electric, «Chapitre 5, Départs moteur »
- [5] Schneider Electric, «LA VARIATION DE FREQUENCE SUR LES INSTALLATIONS DES VENTILATIONS DU CHUV»
- [6] Schneider Electric, « Cahier technique n°204, Protections BT et variateurs de vitesse (convertisseurs de fréquence) »
- [7] A. OUALI, M.TAHRI, Commande scalaire d'un Moteur Asynchrone, Mémoire de Fin d'Etude de MASTER ACADEMIQUE, UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, 2018.
- [8] S.Fahima, M.Hamida MODELISATION ET CONTROLE SCALAIRE D'UNE MACHINE ASYNCHRONE TRIPHASE, Mémoire de Fin d'Etudes De MASTER PROFESSIONNEL,UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU,2014
- [9] : <https://www.elprocus.com/what-is-schrage-motor-circuit-diagram-advantages-its-applications/#:~:text=What%20is%20Schrage%20Motor%3F,and%20the%20wound%20rotor%20induction>
- [10] O.Ahmed , Mehdi, COMMANDE VECTORIELLE D'UNE MACHINE ASYNCHRONE ASSOCIÉE A UN OBSERVATEUR ADAPTATIF, Mémoire Master II,UNIVERSITÉ BADJI MOKHTAR- ANNABA,2019.
- [11] <https://www.eeca.eu/variateur-de-frequence/>
- [12] B.Mounir, G.Youcef, réalisation d'un onduleur de tension commandé par la carte Dspace 1104, Mémoire de master II, Université Abderrahmane Mira de Bejaia, 2015.
- [13] SINAMIC Variateur basse tension SINAMICS G120, Appareils CU 250 S-2 et exploitation de capteur, [www.siemens.com](http://www.siemens.com),2014.
- [14] [Siemens.com/Sinamics-G120C](http://Siemens.com/Sinamics-G120C)
- [15] SINAMICS G120 Inverter chassis units 0.37 kW to 132 kW (0.5 hp to 200 hp)
- [16] SINAMICS Intelligent Operating Panel 2 (IOP-2), [www.siemens.com/drives](http://www.siemens.com/drives)
- [17] B.Rachida, B.Milissa Etude, automatisation par un automate S7-300 et la

supervision de la rectifieuse VOUMARD 300 CNC, Mémoire Master II, Université Mouloud Mammeri De Tizi-Ouzou, 2018

- [18] <https://scietech.fr/architectures-des-automates-programmables-industriels>
- [19] Siemens, SIMATIC S7-1200, S7-1500, Régulation PID, Schéma fonctionnelle.
- [20] <https://www.emerson.com/fr-fr/automation/measurement-instrumentation/flow-measurement/about-liquid-turbine-flow-meters>
- [21] <https://www.plctutorialpoint.com/2017/06/siemens-s7-plc-1200-to-siemens-ac-drive.html>
- [22] O. Houria, S. Nora, Pilotage d'une Plateforme d'assemblage robotisée virtuelle, Apport des techniques Hardware -In-The Loop, Mémoire Master 2, UNIVERSITE MOULOUDE MAMMERI DE TIZI-OUZOU, 2018
- [23] <https://www.plctutorialpoint.com/2017/06/siemens-s7-plc-1200-to-siemens-ac-drive.html>
- [24] <https://electrotoile.eu/demarreur-variateur-vitesse.php>
- [25] TIA portale V16, Help
- [26] : <https://www.lebigdata.fr/interface-homme-machine-tout-savoir-sur-les-ihm>
- [27] **Siemens, Support d'apprentissage/de formation | Module 031-200 TIA Portal, édition 2018 | Digital Factory, DF FA.**
- [28] SIMATIC, Outils d'ingénierie, Aide en ligne de S7-PLCSIM V16.

# Annex

## ➤ Paramètre de régulateur PID Compact

Tableau 8-63 Réglages de base

Numéro	Nom	Réglage d'usine
r0046[0...31]	CO/BO : Déblocages manquants	-
r0052[0...15]	CO/BO : Mot d'état 1	-
r0056[0...15]	CO/BO : Mot d'état Régulation	-
r1084	CO : Limite de vitesse positive active	-
r1087	CO : Limite de vitesse négative active	- tr/min
p2200[C]	BI : Régulateur technologique Déblocage	0
p2252	Régulateur technologique Configuration	Voir la liste des paramètres
p2253[C]	CI : Régulateur technologique Consigne 1	0
p2254[C]	CI : Régulateur technologique Consigne 2	0
p2255	Régulateur technologique Consigne 1 Normalisation	100 %
p2256	Régulateur technologique Consigne 2 Normalisation	100 %
p2257	Régulateur technologique Temps de montée	1 s
p2258	Régulateur technologique Temps de descente	1 s
r2260	CO : Régulateur technologique Consigne en aval du générateur de rampe	- %
p2261	Régulateur technologique Filtre de consigne Constante de temps	0 s
r2262	CO : Régulateur technologique Consigne en aval du filtre	- %
p2263	Régulateur technologique Type	0
r2273	CO : Régulateur technologique Variable d'erreur	- %
p2274	Régulateur technologique Dérivée Constante de temps	0 s
p2280	Régulateur technologique Gain proportionnel	Voir la liste des paramètres
p2285	Régulateur technologique Temps d'intégration	Voir la liste des paramètres
p2286	BI : Régulateur technologique Arrêter l'intégrateur	56,13
p2289[C]	CI : Régulateur technologique Signal de commande anticipatrice	0
p2306	Régulateur technologique Variable d'erreur Inversion	0
p2339	Régulateur techn. Arrêter action I pour n occult. Valeur seuil	- s
r2344	CO : Régulateur technologique Dernière consigne de vitesse (lissée)	- %
p2345	Régulateur technologique Réaction au défaut	0
r2349[0...13]	CO/BO : Régulateur technologique Mot d'état	-
r3889[0...10]	CO/BO : ESM Mot d'état	-

Tableau 8-64 Limitation de la sortie du régulateur technologique

Numéro	Nom	Réglage d'usine
p2290[C]	BI : Régulateur technologique Limitation Déblocage	1
p2291	CO : Régulateur technologique Limite maximale	100 %
p2292	CO : Régulateur technologique Limite minimale	0 %
p2293	Régulateur technologique Temps de montée/descente	1 s
r2294	CO : Régulateur technologique Signal de sortie	- %
p2295	CO : Régulateur technologique Sortie Normalisation	100 %
p2296[C]	CI : Régulateur technologique Sortie Normalisation	2295
p2297[C]	CI : Régulateur technologique Limite maximale Source de signal	1084
p2298[C]	CI : Régulateur technologique Limite minimale Source de signal	1087
p2299[C]	CI : Régulateur technologique Limitation Offset	0
p2302	Régulateur technologique Signal de sortie Valeur de départ	0 %

Tableau 8-65 Adaptation de la mesure du régulateur technologique

Numéro	Nom	Réglage d'usine
p2264[C]	CI : Régulateur technologique Mesure	0
p2265	Régulateur technologique Filtre de mesure Constante de temps	0 s
p2266	CO : Régulateur technologique Mesure en aval du filtre	- %
p2267	Régulateur technologique Limite supérieure Mesure	100 %
p2268	Régulateur technologique Limite inférieure Mesure	-100 %
p2269	Régulateur technologique Gain Mesure	100 %
p2270	Régulateur technologique Mesure Fonction	0
p2271	Régulateur technologique Mesure Inversion (type de capteur)	0
r2272	CO : Régulateur technologique Mesure normalisée	- %

➤ Paramètre control U/f

Paramètre	Description	Réglage d'usine
r0025	CO : Tension de sortie lissée	- Veff
r0066	CO : Fréquence de sortie	- Hz
r0071	Tension de sortie maximum	- Veff
p0304[M]	Tension assignée du moteur	0 Veff
p0310[M]	Fréquence assignée du moteur	0 Hz
p1300[D]	Mode de commande/régulation	0
p1333[D]	Commande U/f Fréquence de démarrage FCC	0 Hz
p1334[D]	Commande U/f Compensation de glissement Fréquence de démarrage	0 Hz
p1335[D]	Compensation de glissement Normalisation	0 %
p1338[D]	Mode U/f Amortissement de la résonance Gain	0

➤ Paramètre control U/f après sélection Standard Drive control

Paramètre	Description	Réglage d'usine
r0025	CO : Tension de sortie lissée	- Veff
r0066	CO : Fréquence de sortie	- Hz
r0071	Tension de sortie maximum	- Veff
p0310[M]	Fréquence assignée du moteur	0 Hz
p501	Application technologique	0

➤ **Paramètre de régulation vectorielle**

Tableau 8-76 Régulation de vitesse sans capteur

Paramètre	Description
p0342	<b>Moment inertie Rapport total / moteur</b> (réglage d'usine : 1,0)
p1496	<b>Commande anticipatrice d'accélération Normalisation</b> (réglage d'usine : 0 %) Le variateur définit le paramètre sur 100 % lors de la mesure en rotation de l'identification des paramètres moteur.
p1452	<b>Régulateur de vitesse Mesure vitesse Temps lissage (ss capteur)</b> (réglage d'usine : 10 ms)
p1470	<b>Régulateur de vitesse Fonctionnement sans capteur Gain P</b> (réglage d'usine : 0,3)
p1472	<b>Régulateur vitesse Fonctionn. sans capteur Temps d'intégration</b> (réglage d'usine : 20 ms)

Tableau 8-77 Régulation de vitesse avec capteur

Paramètre	Description
p0342	<b>Moment inertie Rapport total / moteur</b> (réglage d'usine : 1,0)
p1496	<b>Commande anticipatrice d'accélération Normalisation</b> (réglage d'usine : 0 %) Le variateur définit le paramètre sur 100 % lors de la mesure en rotation de l'identification des paramètres moteur.
p1441	<b>Régulateur de vitesse Temps de lissage</b> (réglage d'usine : 0 ms)
p1442	<b>Régulateur de vitesse Mesure de vitesse Temps de lissage</b> (réglage d'usine : 4 ms)
p1460	<b>Régulateur de vitesse Fonctionnement sans capteur Gain P</b> (réglage d'usine : 0,3)
p1462	<b>Régulateur vitesse Fonctionnement sans capteur Temps d'intégration</b> (réglage d'usine : 20 ms)



