

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Automatique
Spécialité Automatique et Système

présenté par

KADDOURI Adel

&

DJIAR Mohcene

Réalisation d'un banc d'essai de la chaîne d'action « variateur de vitesse – moteur asynchrone » commandé par un API et supervisé par un écran tactile.

Proposé par : Mr. FAS Mohamed Lamine

Année Universitaire 2018-2019

REMERCIEMENT

*Nous tenons à remercier d'abord Dieu le tout puissant qui
Nous a donné la force et la patience pour accomplir ce
modeste travail.*

*Aussi, nous tenons à exprimer nos vifs remerciements à
notre promoteur Mr M.L.FAS pour ses conseils, sa
disponibilité et la confiance qu'il nous a accordé.*

*Nous tenons aussi à remercier les membres de jury d'avoir
accepté de juger notre travail.*

*Sans oublier de remercier tous les membres de nos familles
respectives pour leurs soutient et leurs encouragements,
particulièrement nos chères parents.*

*En fin nous tenons également à remercier toutes les
personnes qui ont participé de près ou de loin à la
réalisation de ce travail.*

Dédicace

Je dédie ce travail :

*A mes très chers parents Youcef et Khlifi Fatma qui n'ont
jamais cessés de m'encourager tout au long de mes études
A ma grand-mère Hayel Kheira qui m'ont appris droiture
et justesse*

A mon cher frère Ayoub et ma sœur Feriel

A toute les nombres de ma famille

*A mes amis, Abdelkader, Mounir, Ahmed, Hssen, Moussa,
Sadek, Tarek, Djallel, Amer*

A ma fiancée Amira

A mon binôme Mohssen.

KADDOURI Adel

Dédicace

Je dédie ce travail :

*A mes très chers parents Othman et Elaïchí Faíza qui n'ont
jamais cessés de m'encourager tout au long de mes études*

A mes grands-parents qui m'ont appris droiture et justesse

A la mémoire de mon frère Nadhír.

A mon cher frère Souhieb et mes sœurs

À mon neveu Nadhír et mes nièces Méléna, Ghízlane, Dalía

A toute les nombres de ma famille

A mes amis, Abdelkader, Mounír, Amíne, Billel, Abdenour,

Sadek, Amel, Isma, Fella, Yacer.

A mon bínôme Adel .

Djiair Mohcene

Résumé

Cette mémoire présente l'étude et la réalisation d'un banc d'essai de la chaîne d'action « moteur asynchrone-variateur de vitesse » commandé par un automate programmable et supervisé avec un écran tactile.

Nous commençons notre travail par une étude théorique sur les moteurs asynchrones et la présentation de capteur de vitesse utilisé (encodeur incrémentale) par donner sa définition et leur principe de fonctionnement. Puis nous allons étudier les variateurs de vitesse et on présente notre variateur de vitesse utilisé dans ce projet réaliser, leur mise en marche et le réglage des paramètres. Par la suite nous avons présenté les différentes parties des systèmes automatisés et on a parlé de la supervision par l'écran tactile L'IHM. Ensuite nous passerons à la partie réalisation du banc d'essai où l'objectif est de programmer l'automate et aussi l'écran tactile afin commander le variateur de vitesse via le bus de communication et lui envoyer des consignes via Modbus. Enfin nous allons récupérer la vitesse du moteur avec un codeur incrémentale et la comparer avec celle lu depuis le variateur de vitesse afin de vérifier si le moteur suit la consigne envoyée.

Mots clés : Variation de vitesse, banc d'essai, systèmes automatisés, API, Moteur asynchrone.

Abstract

This thesis presents the study and the realization of a test bench of the action chain "asynchronous motor- variator of speed " controlled by a PLC and supervised with a touch screen.

We start our work by a theoretical study on asynchronous motors and the presentation of speed sensor used (incremental encoder) to give that definition and their principle of operation. Then we will study the variable speed drives and we presented our speed variator to use in this project to realize, their starting and the setting of the parameters. Subsequently we presented the different parts of the automated systems and we talked about the touch screen supervision The HMI. We will then move to the implementation part of the test bench where the objective is to program the PLC and the touch screen to control the drive speed via the communication bus and send instructions via Modbus. Finally we will recover the speed of the motor with an incremental encoder and compare it with the one read from the variable speed drive to check if the motor follows the setpoint sent.

Key words: speed variation, test bench, automated systems, PLC, Asynchronous motor.

ملخص

هذه المذكرة تقدم دراسة و إنشاء منصة اختبارات للسلسلة العملية "محرك لا تزامني-مغير سرعات" و متحكم فيها بواسطة المتحكم المبرمج و مراقبة بواسطة شاشة لمسية .

نبدأ عملنا من خلال دراسة نظرية حول المحركات غير المتزامنة وعرض مستشعر السرعة المستخدم (التشفير الإضافي) بتعريفه وتقديم مبدأ تشغيله، ثم سنقوم بدراسة مغيرات السرعة ومن ثم قدمنا مغير السرعة المستخدم في مشروعنا المحقق، وكيفية بدأ تشغيله وتحديد المعلمات. ثم قدمنا الأجزاء المختلفة للأنظمة الآلية وناقشنا الإشراف على شاشة اللمس، سننتقل بعد ذلك إلى جزء التنفيذ من منصة الاختبار حيث يكون الهدف هو برمجة المتحكم المبرمج وشاشة اللمس للتحكم في سرعة القيادة عبر ناقل الاتصال وإرسال التعليمات عبر <<Modbus>>. أخيرًا ، سنقوم باستعادة سرعة المحرك من خلال برنامج مستشعر السرعة ومقارنته مع السرعة المسترجعة من مغير السرعات للتحقق مما إذا كان المحرك يتبع السرعة المعطاة.

الكلمات المفتاحية : تغيير السرعة، منصة اختبار، الأنظمة الآلية، متحكم مبرمج، محرك لا تزامني.

Listes des acronymes et abréviations

V : Tension simple aux bornes d'un enroulement.

U : Tension composée entre phase.

p : nombre de paires de pôles.

f : fréquence du courant d'alimentation .

Ω_s : Pulsation de synchronisme.

ns : Vitesse de synchronisme.

n : Vitesse de rotation du rotor.

Cos (φ) : Facteur de puissance.

g : Glissement.

P_a : La puissance absorbée.

I : Courant en ligne .

J : Courant dans un roulement .

P_{js} : Les pertes par effet Joule au stator.

P_{fs} : Les pertes fer au stator.

R : Résistance.

P_{tr} : La puissance transmise au rotor.

T_{em} : Moment du couple électromagnétique.

P_{mt} : Puissance mécanique totale .

P_{jr} : Pertes par effet Joule au rotor.

P_m : Perte mécaniques.

P_u : Puissance utile.

T_u : Couple utile.

η : Le rendement.

n : Vitesse de rotation du rotor .

Ω : Vitesse angulaire.

C : Le Couple.

MAS : Moteur asynchrone.

API : Automate programmable industriel.

PLC : Programmable Logic Controller.

IHM : Interface Homme-Machine.

CPU : Unité centrale de traitement.

DC : Courant continue.

AC : Courant alternative.

PC : Partie Commande .

PO : Partie Opérative .

MLI : Modulation de Largeur d'Impulsion.

TIA : Totally Integrated Automation.

AP : Macro d'application.

BOP : Basic Opérateur Panel.

Cn : macro de connexion.

IGBTS : Insulated Gate Field Effect Transistor.

DSP : Digital Signal Processor.

ROM : Mémoire morte.

HSC : Compteurs rapides.

Sommaire

Introduction générale.....	Error!
Bookmark not defined.1	
CHAPITRE I : Les moteurs asynchrones	02
I.1.Introduction.....	03
I.2. Définition.....	03
I.3. Symbole.....	03
I.4. Schéma de principe.....	04
I.5. Constitution de la machine asynchrone.....	04
I.5.1. Le Stator (inducteur).....	05
I.5.2. Le rotor (induit).....	05
I.5.2.1.Rotor à cage.....	06
I.5.2.2.Le rotor bobiné(rotor à bagues).....	07
I.5.3. Les paliers.....	08
I.5.4. Plaque a bornes.....	08
I.5.5. La plaque d'identification (ou plaque signalétique).....	09
I.5.6. Les flasques.....	10
I.5.7. L'arbre du rotor.....	10
I.5.8. Le ventilateur.....	10
I.5.9. Les roulements.....	10
I.6. Principe de fonctionnement d'un moteur asynchrone.....	10
I.6.1. Glissement d'une machine asynchrone.....	12
I.7. Bobinage et couplage d'un moteur asynchrone.....	12
I.7.1. Bobinage.....	12
I.7.2. Couplage.....	13
I.7.2.1. couplage en étoile.....	13

I.7.2.2. Couplage en triangle.....	13
I.7.2.3. Couplage étoile-triangle.....	14
I.8. Bilan de puissance.....	14
I.9. Les vitesses du moteur asynchrone.....	17
I.9.1. Vitesse de synchronisme.....	17
I.9.2. Vitesse de rotation du moteur.....	17
I.9.3. Vitesse angulaire.....	17
I.9.4. Le Couple.....	18
I.10. Avantages et inconvénients d'un moteur asynchrone et Domaine d'utilisation.....	18
I.10.1. Les Avantages.....	18
I.10.2. Les Inconvénients.....	18
I.10.3. Domaine d'utilisation.....	18
I.11. Capteur de vitesse (encodeur incrémental).....	19
II.11.1. Principe de fonctionnement.....	19
II.11.2. Description de l'encodeur utilisé.....	20
a. Présentation.....	20
c. Correspondance de câblage.....	21
I.12. Conclusion.....	21
CHAPITRE II : Les variateurs de vitesse.....	22
II.1. Introduction.....	23
II.2. Principe de variation de vitesse d'un moteur asynchrone.....	23
II.2.1. Action sur les paires de pôles.....	23
II.2.2. Action sur la fréquence d'alimentation du moteur.....	23
II.3. La variation de la fréquence.....	23
II.4. Variateur de vitesse.....	24
II.4.1. Symbole de variateur de vitesse.....	24
II.4.2. Constitution et principe de fonctionnement d'un variateur de vitesse.....	24
II.4.2.1. Circuit de commande.....	25

II.4.2.2. Redresseur.....	25
II.4.2.3. Etude du fonctionnement.....	26
II.4.2.4. Circuit intermédiaire (filtre).....	27
II.4.2.5 Onduleur.....	27
II.4.2.6 Onduleur de tension triphasé.....	28
II.4.2.7. La commande MLI.....	29
II.5. Fonction des variateurs de vitesse.....	30
II.6. Le variateur de vitesse SIEMENS SINAMICS V20.....	31
II.6.1. Mise en service.....	31
II.6.2. Fonctions des touches.....	32
II.6.3. Icônes d'état du variateur.....	33
II.6.4. Visualisation de l'état du variateur.....	34
II.6.5. Etats des LED.....	34
II.6.6. Démarrage du moteur pour un essai de fonctionnement.....	34
II.6.7. Mise en service rapide.....	35
II.6.7.1. Structure du menu de configuration.....	35
II.6.7.2. Réglage des paramètres moteur.....	36
II.6.7.3. Réglage des macros de connexion.....	37
II.6.7.4. Réglage des macros d'application.....	38
II.6.7.5. Réglage de paramètres communs.....	39
II.7. Avantages et inconvénients d'un variateur de vitesse.....	39
II.7.1. Avantages.....	39
II.7.2. Inconvénients.....	40
II.8. Conclusion.....	40
CHAPITRE III : L'automate programmable Et l'interface Homme-Machine.....	41
III.1. Introduction.....	42
III.2. Définition d'un système automatisé.....	42
III.3. Objectifs de l'automatisation.....	42

III.4. Structure d'un système automatisé.....	42
III.4.1. Partie commande.....	43
III.4.1.1. Unité de traitement d'informations.....	43
a. Automate programmable industriel.....	43
b. Structure matérielle des automates.....	43
b.1. Structure externe.....	44
b.2. Structure interne.....	45
c. Principe général de fonctionnement d'un API.....	46
d. Programmation d'API.....	47
e. Critères de choix de l'automate.....	47
f. Présentation de l'automate S7-1200.....	47
f.1. Principe de fonctionnement du S7-1200.....	48
f.2. Technologie intégrée de S7-1200.....	48
f.3. Choix de la CPU.....	49
g. Modes de fonctionnement de la CPU.....	50
h. Modules de communication.....	50
i. Module d'alimentation.....	52
III.4.2. Partie opérative.....	53
III.4.2.1. Pré-actionneur.....	53
III.4.2.2. Actionneurs.....	53
III.4.2.3. Capteurs.....	53
III.4.3. Partie de supervision.....	54
III.4.3.1. Définition de la supervision.....	54
III.4.3.2. Avantages de la supervision.....	54
III.4.3.3. Interface Homme-Machine.....	55
a. IHM SAMKOON SK-102HS.....	55
b. Différents ports de connexions et d'alimentation de SK-102HS.....	56
c. Caractéristiques techniques de SK-102HS.....	57

III.5. Logiciel de programmation.....	57
III.5.1. Présentation du logiciel TIA Portal V13.....	57
III.5.2. Création d'un nouveau projet.....	58
III.5.3. Configurer et ajouter un appareil.....	59
III.5.4. L'interface et l'adresse Ethernet de CPU.....	60
III.5.5. Ajoute module de communication.....	61
III.5.6. Activation de compteur rapide (HSC).....	61
III.5.7. Langages de programmation.....	62
III.5.8. Blocs de programme.....	64
III.5.9. Compilation et chargement de programme et la configuration matérielle.....	65
III.6. Conclusion.....	66
CHAPITRE IV : Banc d'essai de la variation de vitesse d'un moteur asynchrone.....	67
IV.1 Introduction.....	68
IV.2 Schéma structurelle du banc d'essai.....	68
IV.3 Raccordements des différents éléments du banc d'essai.....	68
IV.3.1 Raccordement du variateur de vitesse avec l'API.....	68
IV.3.2 Raccordement du variateur de vitesse avec le moteur asynchrone.....	71
IV.3.3 Raccordement du moteur asynchrone avec le codeur incrémentale.....	71
IV.3.4 Raccordement du l'encodeur incrémentale avec l'API.....	72
IV.3.5 la communication entre L'automate et l'IHM.....	72
IV.3.6 connexion entre le PC et l'IHM.....	73
IV.3.6.1 Logiciel « SKTOOL ».....	73
IV.4 réglage les paramètres variateur de vitesse (SINAMICS V20).....	75
IV.4.1 les modes de fonctionnement de variateur de vitesse v20.....	75
IV.4.2 le premier mode de fonctionnent Cn003 – Vitesses fixes.....	76
IV.4.2.1 Réglages des paramètres de macro de connexion Cn003.....	76
IV.4.2.2 Programmation de l'API dans TAI 13 pour le macro de connexion Cn003.....	77
a. Table de variable standard.....	77

b. Block d'organisation (programme cycle).....	77
c. Bloc fonctionnel.....	80
IV.4.2.3 Programme IHM dans SKTOOL pour le macro de connexion Cn003.....	82
IV.4.3 le deuxième mode de fonctionnement Cn010 – Régulation USS.....	85
IV.4.3.1 Réglages des paramètres de macro de connexion Cn010.....	85
IV.4.3.2 Programmation de l'API dans TAI 13 pour le macro de connexion Cn010.....	86
a. Table de variable standard.....	86
b. Block d'organisation (programme cycle).....	86
c. Block d'organisation (cyclic interrupt).....	89
d. Bloc fonctionnel.....	91
IV.4.3.3 Programme IHM pour le macro de connexion Cn010.....	92
IV.4.3.4 régulateur PID.....	95
a. Ajout de bloc PID control.....	95
b. Paramètres d'entrée et sortie du contrôleur PID_Compact.....	96
c. Description du bloc PID_Compact.....	96
d. Réglages de des paramètres du bloc PID_Compact	97
e. La mise en service du contrôleur PID_Compact.....	99
e.1.la transmission de la valeur de sortie PID.....	100
IV.5 Conclusion.....	101
Conclusion générale.....	102
Annex.....	104
Référence bibliographique.....	109

Liste des figures

CHAPITRE I

Figure I.01. Symbole d'un moteur asynchrone.....	03
Figure I.02. Schéma de principe d'un moteur asynchrone.....	04
Figure I.03. Constitution du moteur asynchrone.....	04
Figure I.04. Le stator.....	05
Figure I.05. Le rotor a cage et le rotor bobiné.....	06
Figure I.06. Eclaté d'un moteur à rotor à cage.....	07
Figure I.07. Eclaté d'un moteur à rotor à bagues.....	08
Figure I.08. Plaque à bornes.....	09
Figure I.09. Plaque signalétique de moteur asynchrone.....	10
Figure I.10. Principe de fonctionnement d'un moteur asynchrone.....	11
Figure I.11. La liaison des bornes aux bobines.....	13
Figure I.12. Couplage en étoile.....	13
Figure I.13. Couplage en triangle.....	14
Figure I.14. Bilan de puissance du moteur asynchrone.....	14
Figure I.15. Les composants fondamentaux d'un encodeur.....	19
Figure I.16. Signal généré par un encodeur rotatif incrémental.....	20
Figure I.17. L'encodeur incrémental LPD3806-400BM-G5-24C.....	20

CHAPITRE II

Figure II.01. Symbole fonctionnel d'un variateur de vitesse électronique deux sens.....	24
Figure II.02. Structure d'un variateur de vitesse.....	25
Figure II.03. Schéma de principe de redresseur.....	25
Figure II.04. Redresseur triphasé à diodes.....	26
Figure II.05. Tension redressée.....	26
Figure II.06. Schéma du principe de filtrage.....	27
Figure II.07. Schéma de principe de l'onduleur.....	28
Figure II.08. Circuit de puissance d'un onduleur de tension triphasé.....	28
Figure II.09. Schéma d'alimentation d'un MAS avec un onduleur de tension triphasé.....	29

Figure II.10. Modulation Sinus-Triangle.....	30
Figure II.11. Allure du courant pour $f = 50\text{HZ}$	30
Figure II.12. Courbe graphique de fonctionnement de variateur.....	31
Figure II.13. Le variateur de vitesse SINAMICS V20.....	31
Figure II.14. Basic Opérateur Panel (BOP) intégré.....	32
Figure II.15. Le menu d'affichage de variateur.....	34
Figure II.16. La Structure du menu de configuration.....	36
Figure II.17. La plaque signalétique du moteur utilisé.....	36

CHAPITRE III

Figure III.01. Structure d'un système automatisé.....	43
Figure III.02. Automate compact.....	44
Figure III.03. Automate modulaire.....	44
Figure III.04. Structure interne d'un API.....	45
Figure III.05. L'automate programmable S7-1200.....	48
Figure III.06. La CPU S7-1212C.....	50
Figure III.07. L'état des modes fonctionnement de S7-1200.....	50
Figure III.08. Module de communication RS 422/485	51
Figure III.09. Module d'alimentation PM 1207.....	52
Figure III.10. Principe d'un capteur.....	54
Figure III.11. Interface Homme-Machine.....	55
Figure III.12. Face avant de SK-102HS.....	56
Figure III.13. Schéma de connexion de l'écran tactile de la série SK.....	56
Figure III.14. Logo de TIA Portal V13.....	58
Figure III.15. Vue TIA Portal (Création d'un projet).....	58
Figure III.16. Configurer un appareil sur TIA Portal.....	59
Figure III.17. Ajouter un appareil sur TIA Portal.....	59
Figure III.18. Vue de l'interface de TIA Porta.....	60
Figure III.19. Adresse Ethernet de la CPU.....	60
Figure III.20. L'ajoute de module communication.....	61
Figure III.21. Présentation d'un schéma à contacte (CONT).....	63
Figure III.22. Présentation d'un langage structuré (SCL).....	64
Figure III.23. Les différents blocs de programmation.....	64
Figure III.24. La compilation et le chargement de configuration et de programme.....	65
Figure III.25. Mode de connexion.....	66

CHAPITRE IV

Figure IV.01. Schéma structurelle du banc d'essai.....	68
Figure IV.02. Communication variateur SINAMICS V20 avec l'API S7 1200 par l'interface RS-485.....	70
Figure IV.03. Présentation de Communication USS.....	70
Figure IV.04. Raccordement du variateur de vitesse avec le moteur.....	71
Figure IV.05. Raccordement l'arbre du moteur avec l'encodeur incrémental	71
Figure IV.06. Raccordement des sorties du l'encodeur avec l'API.....	72
Figure IV.07. Connexion entre l'API et l'IHM.....	73
Figure IV.08. vu de l'interface de logiciel SKTOOL.....	74
Figure IV.09. macro connexion Cn003.....	76
Figure IV.10. table des variables standards de macro cn003.....	77
Figure IV.11. réseau pour mise en marche de variateur et visualisé leur l'état.....	78
Figure IV.12. réseau pour la command des trois vitesses fixe.....	78
Figure IV.13. le bloc fonctionnel pour afficher la fréquence commandé et Calculer la consigne de vitesse.....	79
Figure IV.14. ce réseau pour calculer la vitesse réelle de moteur.....	79
Figure IV.15. Réseau pour calculer l'erreur entre la consigne de vitesse et la vitesse réelle.....	80
Figure IV.16. la tabla des variables pour bloc "Bloc_1_DB".....	80
Figure IV.17. Le programme pour le block "Bloc_1_DB".....	80
Figure IV.18. La tabla des variables du bloc "Bloc_2_DB".....	81
Figure IV.19. Le programme du bloc "Bloc_2_DB".....	81
Figure IV.20. la première fenêtre dans l'IHM.....	82
Figure IV.21. La deuxième fenêtre dans l'IHM.....	83
Figure IV.22. La courbe de l'erreur de vitesse et leur valeur.....	83
Figure IV.23. La courbe de vitesse réelle de moteur.....	84
Figure IV.24. La courbe variation de fréquence et leur valeur.....	84
Figure IV.25. le macro de connexion cn010.....	85
Figure IV.26. Table des variables standards de macro cn010.....	86
Figure IV.27. Bloc fonctionnel intégrer "USS_DRIVE".....	87
Figure IV.28. Bloc fonctionnel "Bloc_3_DB".....	89
Figure IV.29 réseau pour calculer l'erreur entre la consigne de vitesse et la	

Vitesse réelle.....	89
Figure IV.30. Bloc fonctionnel intégrer "USS_PORT".....	90
Figure IV.31. La tabla des variables pour bloc "Bloc_3_DB".....	91
Figure III.32. Le programme du bloc "Bloc_3_DB".....	92
Figure IV.33. la premier fenêtre de l'IHM pour macro cn010.....	92
Figure IV.34. La deuxième fenêtre de l'IHM pour macro cn010.....	93
Figure IV.35. la courbe de l'erreur de vitesse et leur valeur.....	93
Figure IV.36. La courbe de vitesse réelle de moteur.....	94
Figure IV.37. La courbe variation de fréquence et leur valeur.....	94
Figure IV.38. Ajout de bloc PID control.....	96
Figure IV.39. Représentions bloc PID_Compact.....	97
Figure III.40. configuration de type de régulation, paramètres d'entrees sorties le sens de rotation.....	98
Figure IV.41. configuration des paramètres de mesure.....	98
Figure IV.42. les paramètres de contrôleur PID.....	99
Figure IV.44. Représentation de la fenêtre mise en service de contrôleur PID_Compact et leur régulation.....	100
Figure IV.30 réseau de transmettre de pourcentage de fréquence de la sortie PID à l'entré Speed_sp bloc "USS_DRIVE" par l'instruction de transfert "MOVE".....	100
Figure IV.45. La courbe de l'erreur de vitesse et leur valeur d'après (régulation PID).....	101

Liste des tableaux

CHAPITRE I

Tableau I.1. Caractéristiques de l'encodeur incrémental LPD3806-400BM-G5-24C.....	21
Tableau I.2. Correspondance de câblage.....	21

CHAPITRE II

Tableau II.1. Les Fonctions des touches.....	32
Tableau II.2. Les Icônes d'état du variateur.....	33
Tableau II.3. Les états des LED.....	34
Tableau II.4. La Structure du menu de configuration.....	35
Tableau II.5. Réglage des paramètres moteur.....	37
Tableau II.6. La recréation des macros de connexion.....	37
Tableau II.7. Réglage des macros d'application.....	38
Tableau II.8. Les paramètres Macro d'application AP010.....	39
Tableau II.9. Réglage de paramètres communs.....	39

CHAPITRE III

Tableau III.1. Tableau de comparaison des CPU S7-1200.....	49
Tableau III.2. Tableau d'information générale sur RS 422/485	51
Tableau III.3. Tableau de caractéristique de SK 102HS.....	57
Tableau III.4. Activation et fonctionnement et adresse E/S d'un HSC.....	62

CHAPITRE IV

Tableau IV.1. paramètres Macro de connexion Cn003.....	76
Tableau IV.2. paramètres Macro de connexion Cn010.....	85
Tableau IV.3. paramètres de l'instruction "USS_DRIVE".....	87
Tableau IV.4. les paramètres de l'instruction "USS_PORT".....	90
Tableau IV.5. Paramètres PID_Compact.....	96

Introduction générale

Les machines électriques ou machines tournantes sont des dispositifs électriques qui permettent de générer un mouvement ou une énergie mécanique à partir d'une énergie électrique. Elles occupent une place prépondérante dans tous les secteurs industriels. Ces dernières sont classées en trois catégories qui sont les machines à courant continu, les machines synchrones et les machines asynchrones.

Parmi tous les types de moteurs existants, les moteurs asynchrones jouent un rôle très important dans les systèmes industriels ces derniers avons une conception simple, robuste et peu couteux à l'achat et à l'entretien. Il peut aussi fonctionner dans des milieux critiques (poussières, température, humidité,...), la vitesse des moteurs asynchrone dépend principalement de sa fréquence et pour la varier on utilise les variateurs de vitesse.

L'objectif principal de notre travail est de réaliser et commander un banc d'essai de la variation de vitesse d'un moteur asynchrone commandé par un automate programmable industriel via le bus de communication Modbus avec le protocole USS. Nous avons aussi supervisé notre système avec un écran tactile afin d'avoir un suivi en temps réel.

Pour atteindre notre objectif, nous avons organisé notre mémoire en quatre chapitres :

Le premier chapitre consacré sur le moteur asynchrone et le capteur de vitesse utilisé (encodeur incrémental).

Le troisième chapitre aborde les systèmes automatisés, comme l'automate programmable siemens S7-1200, les modules communication et d'alimentation. Dans le la suite de ce chapitre on présente l'écran tactile utilisé (IHM) de marque SAMKOON SK-102HS.

Le quatrième chapitre, qui est la partie réalisation de notre projet. Cette réalisation est basée sur la commande d'un banc d'essai de la chaîne d'action (moteurs asynchrones et variateurs de vitesse) commandé par un API et supervisé avec un écran tactile. A la fin on utilise le régulateur PID pour corriger l'erreur la vitesse réelle et la consigne de la vitesse.

Enfin, nous clôturons notre document avec une conclusion générale qui est dédiée à une synthèse du travail présenté.

CHAPITRE I

Les moteurs asynchrones

I.1. Introduction

Les moteurs asynchrones triphasés représentent plus de 80% du parc de machines électriques, sa simplicité de construction en fait un matériel très fiable et qui demande peu d'entretien. C'est une machine robuste économique à l'achat et ne nécessitant que peu de maintenance .de plus la vitesse de rotation est presque constante sur une large plage de puissance .Certaines applications industrielles nécessitant la variation de vitesse et les asservissements mécaniques lui ont été par conséquent refusés malgré la recherche de solutions technologiques parfois innovantes. C'était vrai jusqu'à l'apparition des convertisseurs de fréquence.

Dans ce chapitre, nous présenterons la variation de vitesse d'un moteur asynchrone. Après avoir donné la constitution d'un moteur asynchrone, nous expliquerons le principe de fonctionnement et les caractéristiques de ce dernier. Enfin nous expliquerons le principe de la variation de vitesse des moteurs asynchrones ainsi qu'on va expliquer le capteur de vitesse utilisé dans notre projet réalisé.

I.2. Définition

Le moteur asynchrone est une machine qui transforme de l'énergie électrique en énergie mécanique. Le fonctionnement est basé sur la production d'un champ tournant. La machine asynchrone est une machine à *courant alternatif* sans connexion entre le stator et le rotor. Le terme asynchrone provient du fait que la vitesse de ces machines n'est pas forcément proportionnelle à la fréquence des courants qui la traversent. Les moteurs asynchrones sont uniquement alimentés par des systèmes de courant triphasé [1].

I.3. Symbole

Il y a deux symboles pour le moteur asynchrone triphasé, un symbole pour le moteur à cage d'écureuil et l'autre pour le moteur à rotor bobiné.

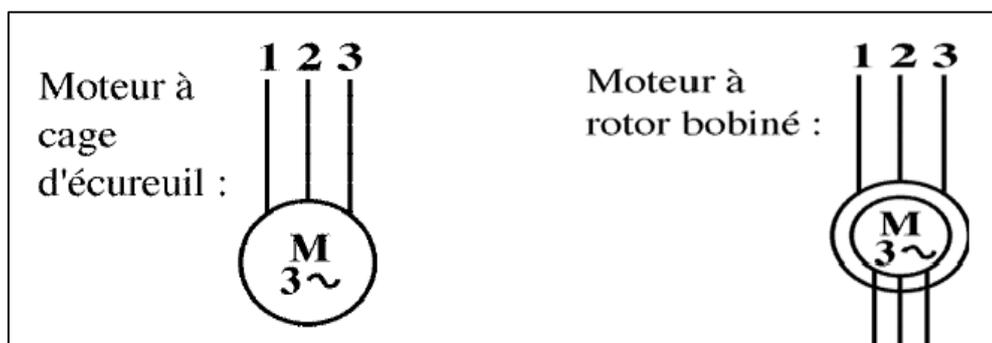


Figure I.1. Symbole d'un moteur asynchrone.

I.4. Schéma de principe

Le rôle du moteur asynchrone est de convertir l'énergie électrique en énergie mécanique.

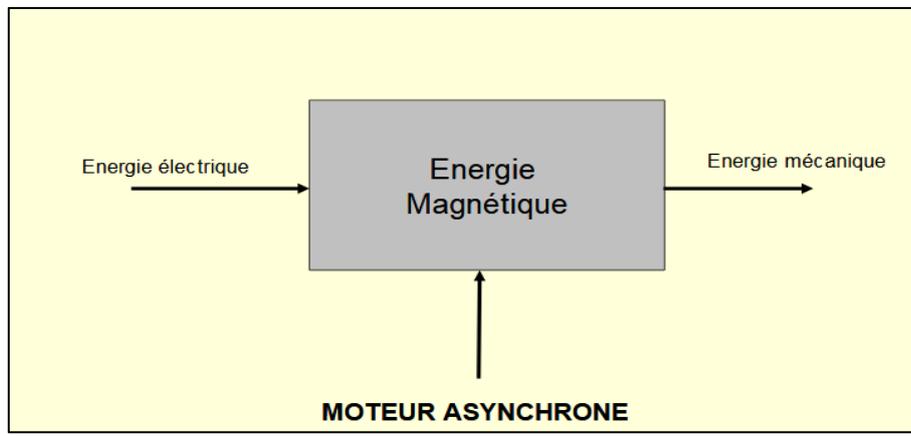


Figure I.2. Schéma de principe d'un moteur asynchrone.

I.5. Constitution de la machine asynchrone

La machine asynchrone est la machine la plus utilisée du faite qu'elle nécessite peu d'entretien, moins coûteuse, et se présente avec une construction assez simple, elle est aussi très connue par sa standardisation et sa robustesse [2].

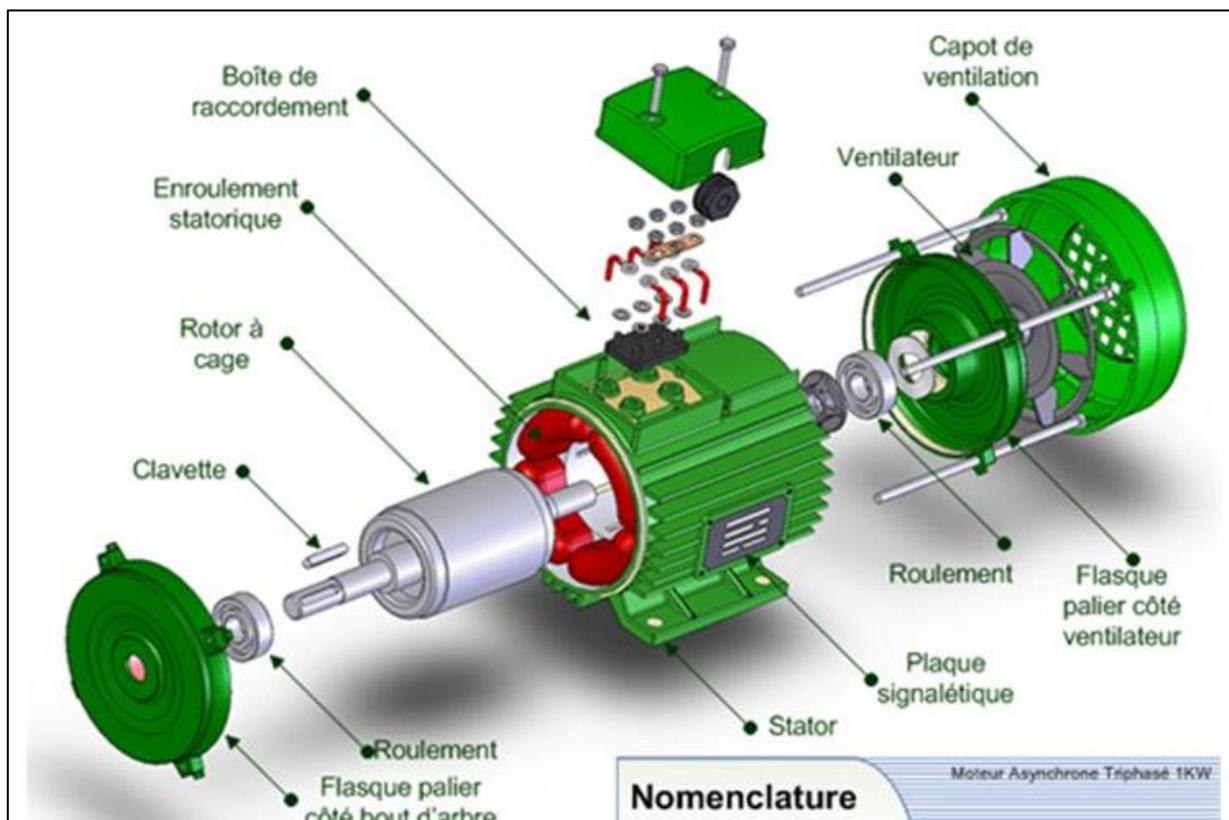


Figure I.3. Constitution du moteur asynchrone.

Le moteur asynchrone est constitués par :

I.5.1. Le Stator (inducteur)

C'est la partie fixe du moteur, il est constitué de trois enroulements (bobines) et possède p paires de pôles. Il est identique à celui d'une machine synchrone alimentée par des courants triphasés de fréquence f et produit un champ magnétique tournant à la pulsation de synchronisme Ω_s qui se répartit sinusoïdalement dans l'entrefer (espace entre le rotor et le stator) de la machine. Les enroulements du stator sont prévus pour être couplés, soit en étoile, soit en triangle. Le couplage des enroulements dépendra de la tension nominale par phase prévue pour le moteur et du réseau dont on dispose.



Figure I.4. Le stator.

I.5.2. Le rotor (induit)

C'est l'élément mobile du moteur. Le rotor du moteur asynchrone triphasé peut-être à cage d'écurueil ou bobiné. Le rotor ne possède aucune liaison électrique avec le stator. Le rotor constitue un circuit électrique fermé où se crée des courants induits qui entraînent la mise en rotation du rotor. Le rotor tourne à la vitesse de rotation n qui est inférieure à la vitesse synchrone n_s . On dit que le rotor glisse par rapport au champ magnétique tournant; on parle alors de glissement qui dépend de la charge.

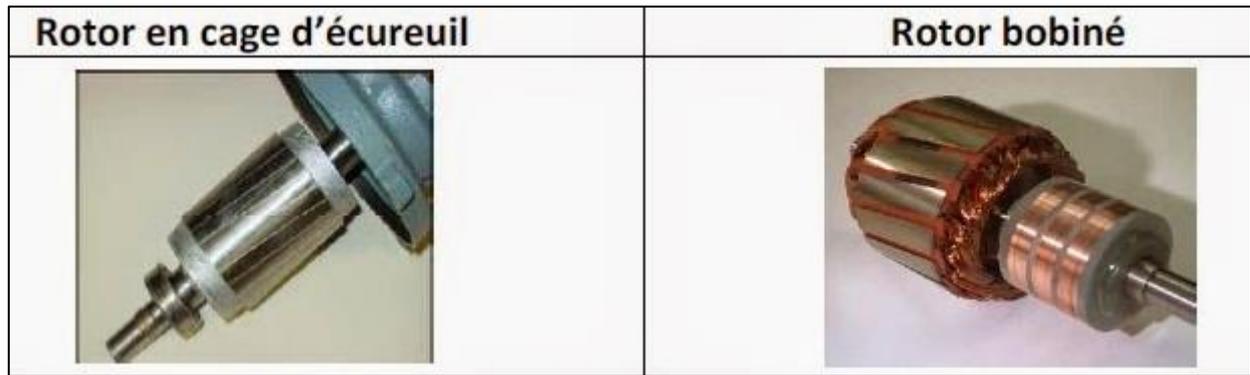


Figure I.5. Le rotor a cage et le rotor bobiné.

I.5.2.1. Rotor à cage

Le circuit du rotor est constitué de barres conductrices régulièrement réparties entre deux couronnes métalliques formant les extrémités, le tout rappelant la forme d'une cage d'écureuil. Bien entendu, cette cage est insérée à l'intérieur d'un circuit magnétique analogue à celui du moteur à rotor bobiné. Les barres sont faites en civre, en bronze ou en aluminium, suivant les caractéristiques mécaniques et électriques recherchées par le constructeur. Dans certaines constructions, notamment pour des moteurs à basse tension (par exemple 230/400 V), la cage est réalisée par coulée et centrifugation d'aluminium. On démontre que, si le nombre de barres est suffisamment grand (soit en pratique, la cage se transforme automatiquement en un circuit polyphasé de polarité adéquate). Ce type de moteur, beaucoup plus aisé à construire que le moteur à rotor bobiné est par conséquent d'un prix de revient inférieur et a une robustesse intrinsèquement plus grande. Il n'est donc pas étonnant qu'il constitue la plus grande partie du parc des moteurs asynchrones actuellement en service. Son inconvénient majeur est qu'il a, au démarrage, de mauvaises performances (courant élevé et faible couple). C'est pour remédier à cette situation qu'ont été développés deux autres types de cages (rotor à double cage et rotor à encoches profondes) [3].

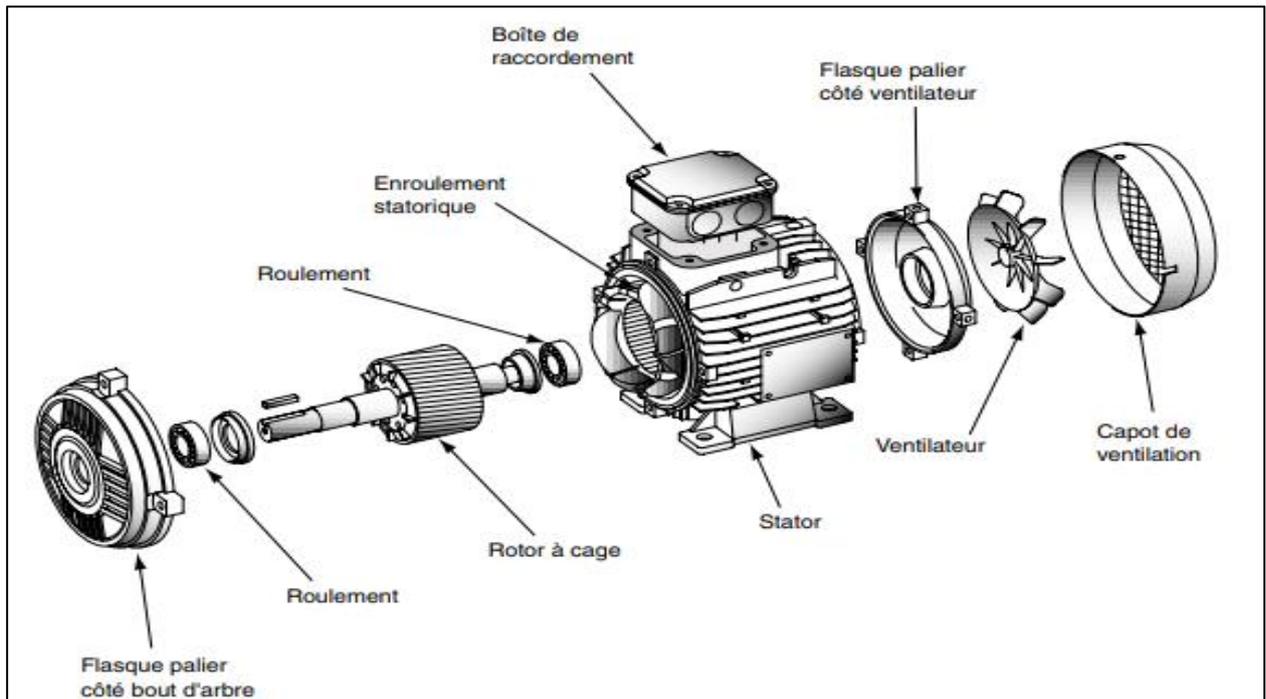


Figure I.6. Eclaté d'un moteur à rotor à cage.

Il existe différents types de rotor à cage [4]:

- ✓ Rotor à simple cage
- ✓ Rotor à double cage
- ✓ Rotor à cage résistante
- ✓ Rotor à encoches profondes

I.5.2.2. Le rotor bobiné (rotor à bagues)

Dans des encoches pratiquées à la périphérie du rotor sont logés des enroulements identiques à ceux du stator. Généralement le rotor est triphasé. Une extrémité de chacun des enroulements est reliée à un point commun (couplage étoile). Les extrémités libres peuvent être raccordées sur un coupleur centrifuge ou sur trois bagues en cuivre, isolées et solidaires du rotor. Sur ces bagues viennent frotter des balais à base de graphite raccordés au dispositif de démarrage.

En fonction de la valeur des résistances insérées dans le circuit rotorique, ce type de moteur peut développer un couple de démarrage s'élevant jusqu'à 2,5 fois le couple nominal. Cette solution est de plus en plus abandonnée au profit de solutions électroniques associées à un moteur à cage standard. En effet ces dernières permettent de résoudre des problèmes de maintenance (remplacement des balais d'alimentation du rotor usés, entretien des résistances de réglage), de réduire l'énergie dissipée dans ces résistances et aussi d'améliorer de façon importante le rendement de l'installation.

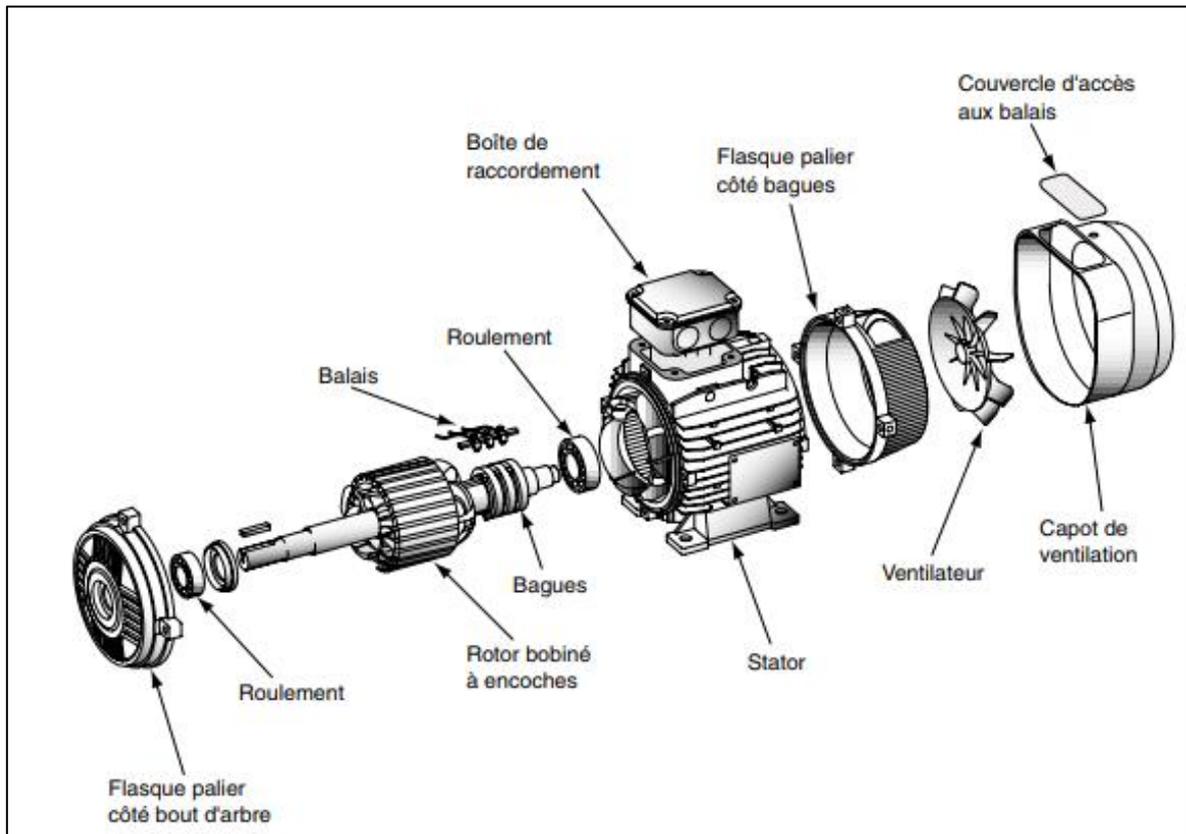


Figure I.7. Eclaté d'un moteur à rotor à bagues.

I.5.3. Les paliers

Les paliers, qui permettent de supporter et de mettre en rotation l'arbre rotorique, sont constitués de flasques et de roulements à billes insérés à chaud sur l'arbre. Les flasques, moulés en fonte, sont fixés sur le carter statorique grâce à des boulons ou des tiges de serrage.

I.5.4. Plaque a bornes

C'est sur la plaque à bornes située dans la boîte à bornes que sont raccordées les extrémités des enroulements du stator. C'est également sur cette plaque que vient se raccorder l'alimentation du moteur. Par convention, si un moteur est alimenté de la façon suivante **L1** sur **U1**, **L2** sur **V1** et **L3** sur **W1**, il tourne dans le sens horaire. Il est donc facile de définir son branchement quand le sens de rotation est imposé pour entraîner une charge. Pour s'assurer de l'ordre des phases, on effectue un contrôle d'ordre des phases. Pour inverser le sens de rotation d'un moteur, il suffit d'inverser 2 phases au niveau de l'alimentation du moteur. C'est aussi sur cette plaque que seront placées les barrettes de couplage, lorsque le couplage du moteur sera fixe.

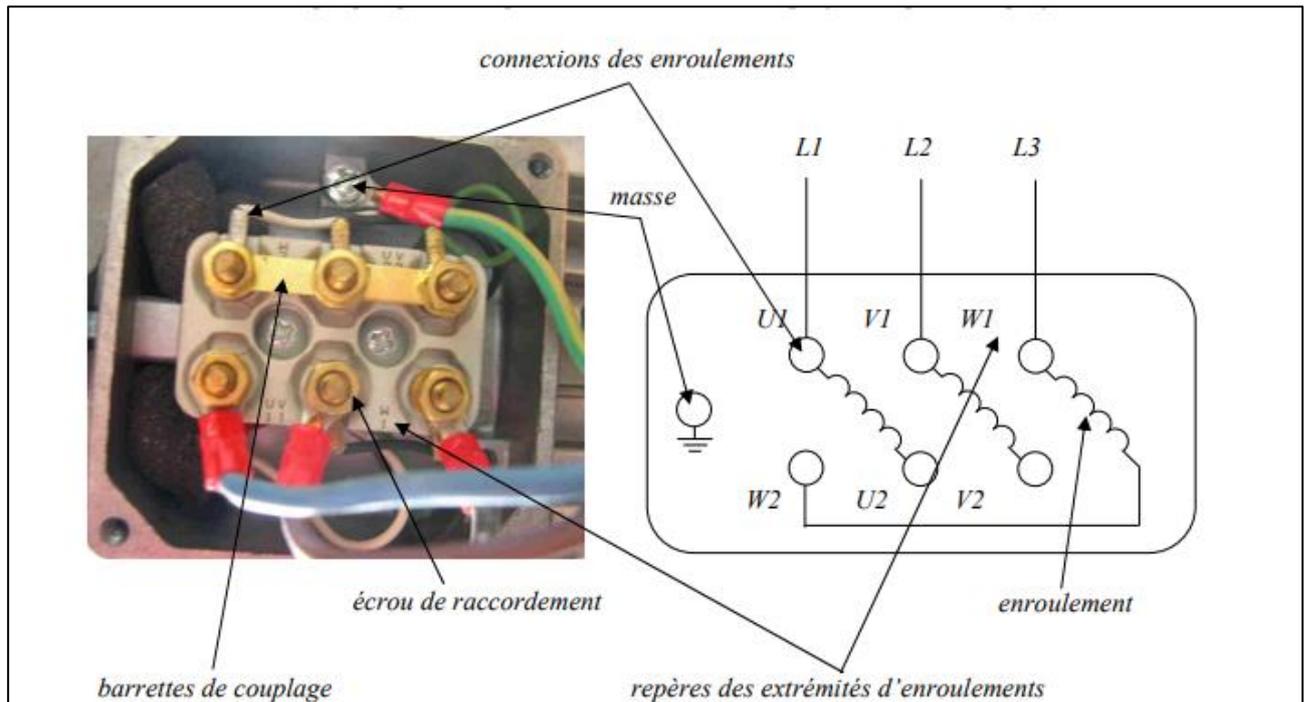


Figure I.8. Plaque à bornes.

I.5.5. La plaque d'identification (ou plaque signalétique)

La plaque signalétique est fixée sur la carcasse. C'est la carte d'identité du moteur, Sur la plaque signalétique, le constructeur indique :

- ✓ **Type** : référence propre au constructeur.
- ✓ **Puissance** : puissance utile délivrée sur l'arbre du moteur.
- ✓ **Facteur de puissance** ou $\cos \varphi$: permet le calcul de la puissance réactive consommée.
- ✓ **Rendement** : permet de connaître la puissance électrique consommée ou absorbée.
- ✓ **Tensions** : (230V/400V) la première indique la valeur nominale de la tension aux bornes d'un enroulement. Elle détermine le couplage (étoile ou triangle) à effectuer en fonction de la tension du réseau d'alimentation.
- ✓ **Intensités** : Elles représentent l'intensité en ligne (dans chaque phase) pour chacun des couplages.
- ✓ **Vitesse** : Indique la vitesse nominale du rotor. On dit aussi vitesse réelle. On connaît alors la vitesse de synchronisme n_s du moteur.
- ✓ Température ambiante utilisation recommandée maximum
- ✓ **Fréquence** : fréquence du réseau d'alimentation.
- ✓ **Nombre de phases** : (Ph 3) moteur triphasé.

- ✓ **Indice de protection IP** : défini par trois chiffres le degré de protection du moteur à la poussière, à l'eau et aux chocs mécaniques.

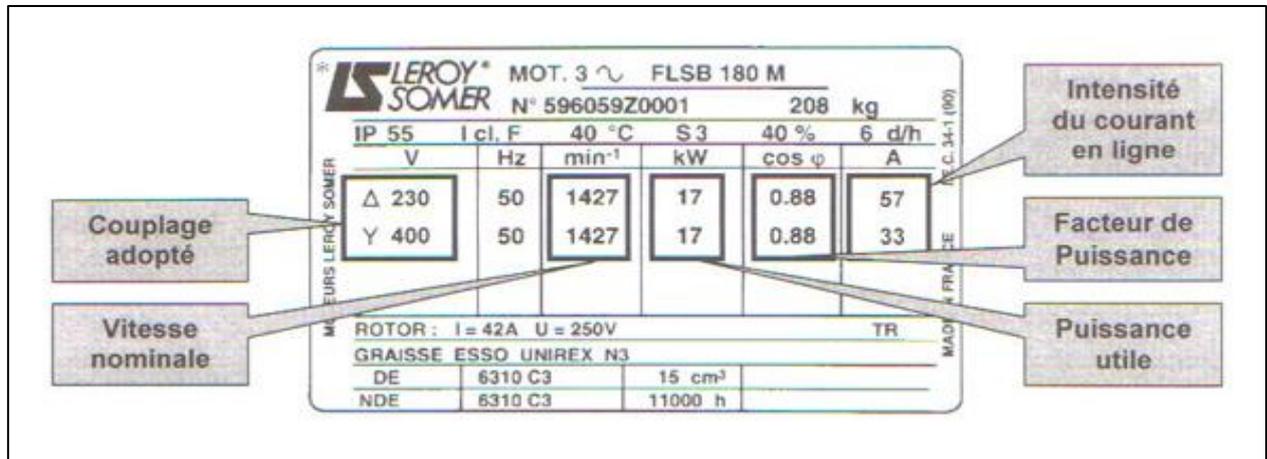


Figure I.9. Plaque signalétique de moteur asynchrone.

I.5.6. Les flasques

Ils ferment le carter moteur aux deux extrémités et sont réalisés en font grise ou en aluminium injecté. Les flasques sont centrés sur le carter et réunis entre eux par des tirants ou tiges d'assemblage.

I.5.7. L'arbre du rotor

Le rotor est monté sur un arbre en acier. A une de ses extrémité est monté le ventilateur et éventuellement le frein, à l'autre côté on trouve l'arbre de sortie avec une rainure de clavetage pour le montage du pignon d'entraînement.

I.5.8. Le ventilateur

Le ventilateur est placé à l'arrière du moteur, il permet le refroidissement du moteur. Le capot oriente le flux d'air vers les ailettes du carter.

I.5.9. Les roulements

Sur les MAS, le guidage en rotation de l'arbre se fait par deux roulements à billes montés dans les flasques.

I.6. Principe de fonctionnement d'un moteur asynchrone

Le fonctionnement d'une machine asynchrone est basé sur le principe de l'interaction électromagnétique du champ tournant créée par le courant triphasé fourni à l'enroulement

statorique par le réseau, et des courants induits dans l'enroulement rotorique lorsque les conducteurs de ce dernier sont coupés par le champ tournant. De cette façon le fonctionnement d'une machine asynchrone est analogue à celui d'un transformateur : le stator étant comparable à l'enroulement primaire et le rotor à l'enroulement secondaire qui, dans le cas général, peut tourner à la vitesse de rotation donnée par le rapport suivant:

$$n_s(\text{tr/s}) = \frac{f(\text{Hz})}{p} \quad (\text{I.1})$$

f : la fréquence en [Hz].

p : nombre de paires de pôles.

n_s : la vitesse de synchronisme en [tr/s] .

les moteurs asynchrones est basé sur la production d'un champ tournant qui produit par trois bobinages fixes géométriquement décalés de 120° , et parcourus par des courants statoriques alternatifs présentant le même décalage électrique, les trois champs alternatifs qu'ils produisent se composent pour former un champ tournant d'amplitude constante [5].

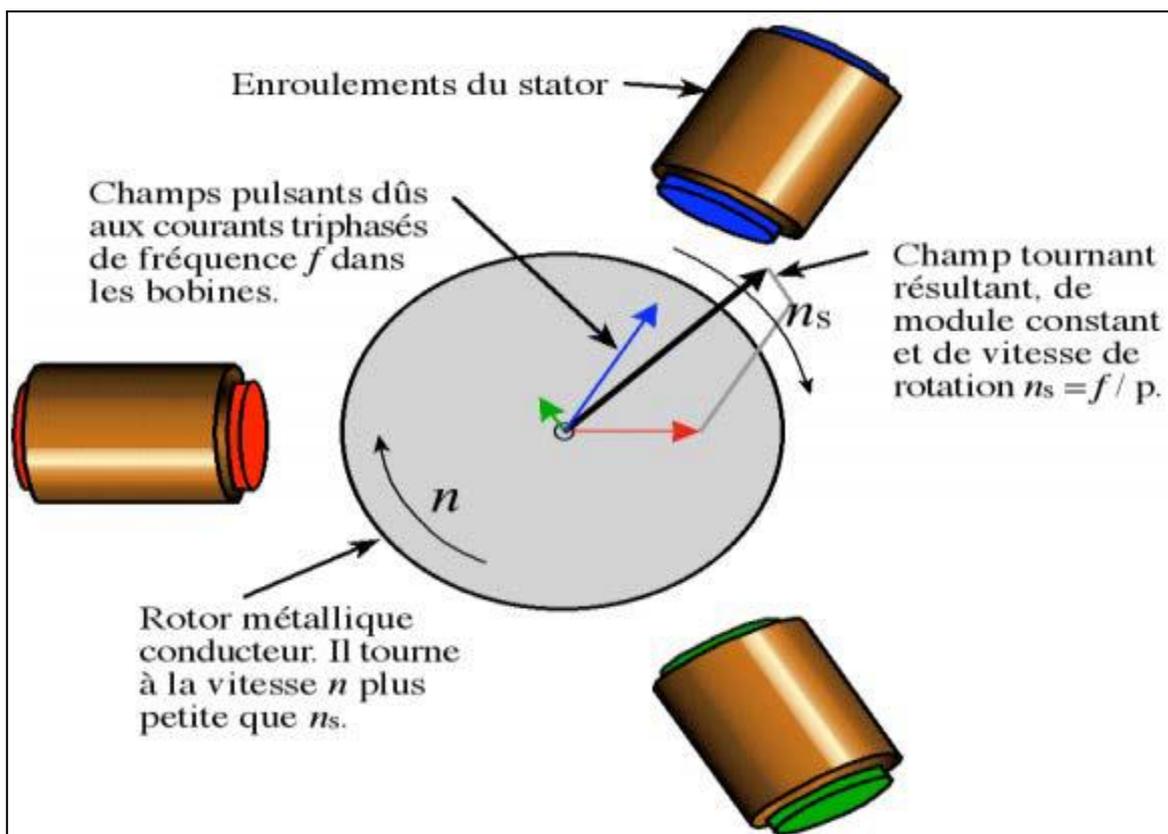


Figure I.10. Principe de fonctionnement d'un moteur asynchrone.

I.6.1. Glissement d'une machine asynchrone

On appelle glissement l'écart des vitesses angulaires synchrone Ω_s et réelle rapporté à la vitesse angulaire synchrone Ω_r . On peut comparer les pulsations correspondant à ces vitesses

$$\omega_s = p \Omega_s \quad (\text{I.2})$$

$$\omega_r = p \Omega_r \quad (\text{I.3})$$

Ou les vitesses en tours par seconde

$$N_s = \frac{\Omega_s}{2\pi} \quad (\text{I.4})$$

$$N_r = \frac{\Omega_r}{2\pi} \quad (\text{I.5})$$

D'où les diverses définitions du glissement g :

$$g = \frac{\Omega_s - \Omega_r}{\Omega_s} = \frac{N_s - N_r}{N_s} \quad (\text{I.6})$$

Examinons les différents cas possibles :

$g = 0$ pour $N_s = N_r$ il n'y a pas de couple.

$g < 0$ pour $N_s > N_r$ la machine fonctionne en génératrice.

$g > 0$ pour $N_s < N_r$ la machine fonctionne en moteur.

$g = 1$ pour $N_r = 0$ la machine est à l'arrêt ou à l'instant du début du démarrage.

I.7. Bobinage et couplage d'un moteur asynchrone

I.7.1. Bobinage

Les bobines sont logées dans les encoches du stator. S'il y'a une paire de pôles magnétique pour chacune des trois phases, la fréquence de synchronisme est alors de 3000 tr/min. Si on augmente le nombre de paires de pôles, il est possible d'obtenir des moteurs avec des fréquences de rotations différentes.

- 1 paire de pôles $\Rightarrow N_s = 3000$ tr/min
- 2 paires de pôles $\Rightarrow N_s = 1500$ tr/min
- 3 paires de pôles $\Rightarrow N_s = 750$ tr/min

Le branchement des bobines sur le réseau se fait au niveau de la plaque à borne située sur le dessus du moteur. On dispose ainsi de 6 connexions, une pour chacune des extrémités des trois bobines. Les bornes sont reliées aux bobines selon le schéma suivant :

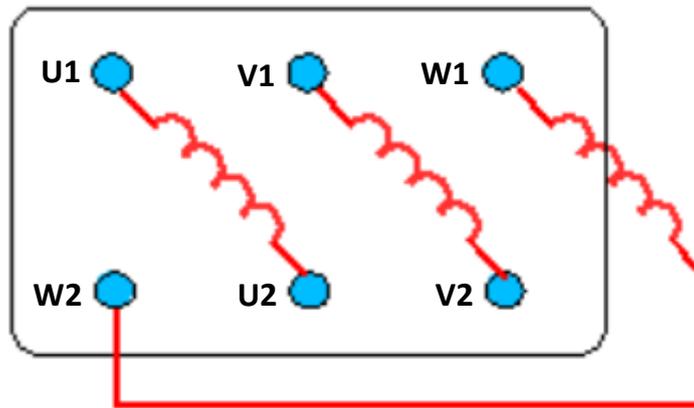


Figure I.11. La liaison des bornes aux bobines.

I.7.2. Couplage

Le couplage pour les moteurs asynchrone, il possible de coupler soit en **triangle**, en **étoile**, soit en **étoile-triangle**.

I.7.2.1. couplage en étoile

Avec un branchement en étoile, la tension aux bornes de chacune des bobines est d'environ 230V. On utilise le montage étoile si un moteur de 230V doit être relié sur un réseau de 400V ou pour démarrer un moteur à puissance réduite dans le cas d'une charge avec une forte inertie mécanique. Ce branchement se fait comme il est montré dans la figure I.12.

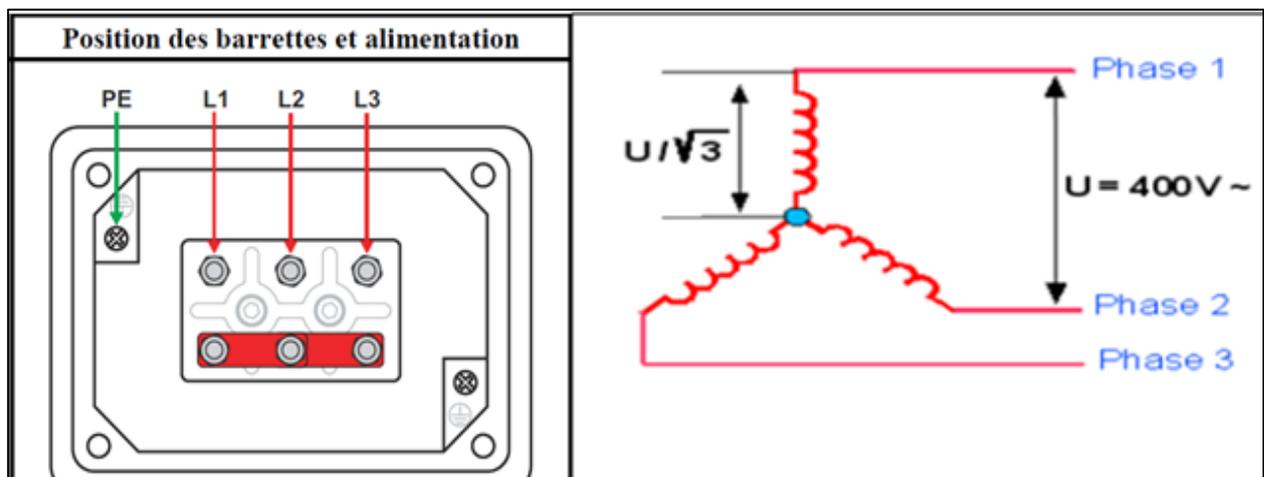


Figure I.12. Couplage en étoile.

I.7.2.2. Couplage en triangle

Dans le montage en triangle, chacune des bobines est alimentée avec la tension nominale (400V) comme il est montré dans la figure I.13.

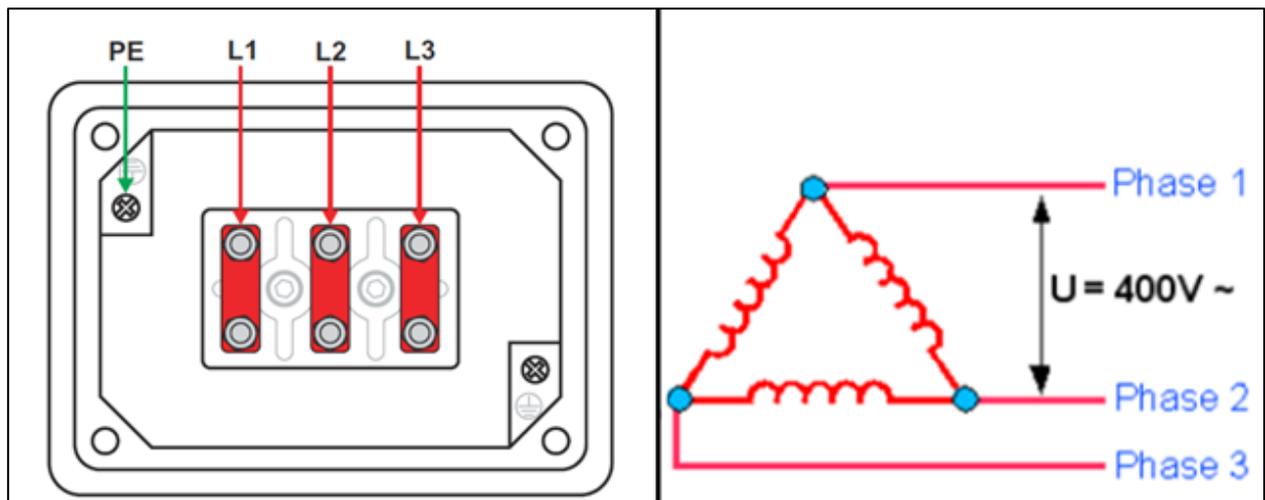


Figure I.13. Couplage en triangle.

I.7.2.3. Couplage étoile-triangle

Le démarrage étoile triangle est très utilisé en électrotechnique pour la mise en route des moteurs électriques asynchrones triphasés. Ce dispositif est employé afin de diminuer les risques du démarrage direct. En effet, l'intensité du courant au démarrage (en direct) est très importante vis à vis du courant nominal du moteur (environ 5 à 7 fois l'intensité nominale). Sur les gros moteurs ces courants importants entraînent des surcharges sur les lignes d'alimentations (fils, câbles, bornes) et sur les appareils de protection et de commande (fusible, sectionneur, contacteur, relais thermique...) d'où une usure, voir une destruction, prématurée des composants du démarreur.

I.8. Bilan de puissance

Le bilan des puissances et des pertes dans un moteur asynchrone est reporté sur la Figure I.14.

Le moteur asynchrone absorbe sur le réseau une puissance électrique active triphasée P_a .

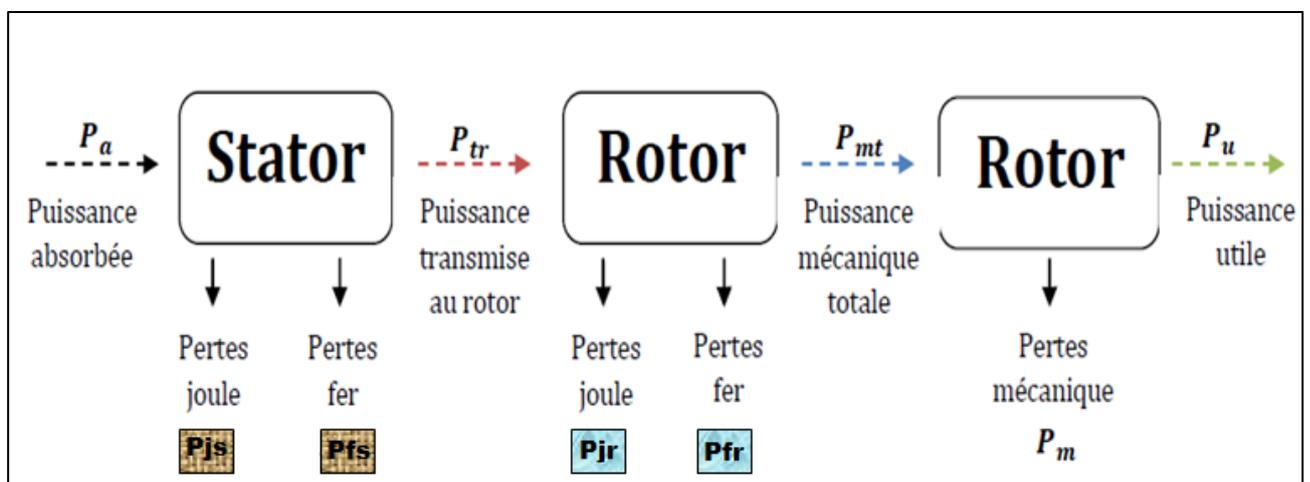


Figure I.14. Bilan de puissance du moteur asynchrone.

❖ Expression de la puissance absorbée P_a

La puissance absorbée par un moteur asynchrone est une puissance active électrique. Lorsque le moteur est connecté à un réseau triphasé, P_a vérifie la relation :

$$P_a = U \cdot I \cdot \sqrt{3} \cdot \cos(\varphi) = 3 \cdot V \cdot J \cdot \cos(\varphi) \quad (\text{I.7})$$

Avec :

U : tension composée donc entre phases (V).

V : tension simple donc aux bornes d'un enroulement (V).

I : courant en ligne (A).

J : courant dans un enroulement (A).

$\cos(\varphi)$: facteur de puissance sans unité.

❖ Les pertes par effet Joule au stator P_{js}

Le stator peut être câblé en étoile ou en triangle. Puisque le moteur constitue un récepteur triphasé équilibré, alors :

$$P_{js} = \frac{3}{2} R I^2 \quad (\text{I.8})$$

Avec R : est la résistance entre deux bornes.

✚ Dans le montage étoile, la résistance entre deux bornes R est reliée à la résistance de chaque enroulement r par la relation :

$$R = 2 r \quad (\text{I.9})$$

✚ Dans le montage triangle, cette relation devient :

$$R = \frac{2}{3} r \quad (\text{I.10})$$

❖ Les pertes fer au stator P_{fs}

Notés P_{fs} elles regroupent les pertes par hystérésis et par courant de Foucault dans les masses métalliques du stator.

❖ La puissance transmise au rotor P_{tr}

Une partie de la puissance absorbée 1% à 2% est perdue dans le stator sous forme de pertes dans le fer et de pertes dans le cuivre dues à l'effet joule P_{js} . La puissance restante la puissance transmise au rotor.

$$P_{tr} = P_a - (P_{js} + P_{fs}) \quad (\text{I.11})$$

❖ Moment du couple électromagnétique T_{em}

Les forces qui s'exercent sur les conducteurs du rotor tournent à la vitesse Ω_s : elles glissent sur le rotor qui, ne tourne qu'à la vitesse Ω .

L'action de l'ensemble des forces électromagnétiques se réduit à un couple électromagnétique résultant du moment T_{em} .

$$T_{em} = \frac{P_{tr}}{\Omega_s} \quad (\text{I.12})$$

❖ Puissance mécanique totale P_{mt}

Le couple électromagnétique du moment T_{em} entraîne le rotor à la vitesse Ω . Il lui transmet donc la puissance mécanique totale P_{mt} .

$$P_{mt} = T_{em} \Omega \quad (\text{I.13})$$

Soit :

$$P_{mt} = T_{em} \Omega = P_{tr} \frac{\Omega}{\Omega_s} = P_{tr} (1 - g) \quad (\text{I.14})$$

$$P_{mt} = P_{tr} (1 - g) \quad (\text{I.15})$$

❖ Pertes par effet Joule au rotor P_{jr}

Les pertes Joule au rotor sont fonctions de la puissance transmise à celui-ci. On peut par ailleurs montrer qu'elles sont aussi fonctions du glissement. En effet, les grandeurs P_{jr} et P_{tr} sont liées par la relation :

$$P_{jr} = g P_{tr} \quad (\text{I.16})$$

Les pertes fer du rotor sont négligeables.

❖ Perte mécaniques

Elles correspondent aux frottements sur les paliers, à la ventilation, etc. Elles sont négligées dans la plus part des cas.

$$P_m = P_U - P_{mt} \quad (\text{I.17})$$

❖ Puissance utile P_U

C'est la puissance mécanique diminuée des pertes mécaniques. La puissance indiquée sur la plaque signalétique d'un moteur asynchrone et la puissance utile nominale. Elle se calcule comme suit :

$$P_U = P_{mt} - P_m \quad (\text{I.18})$$

Le couple utile se calcul alors avec l'équation suivante :

$$T_U = \frac{P_U}{\Omega} \quad (\text{I.19})$$

❖ Le Rendement η

Le rendement η du moteur asynchrone est le rapport entre la puissance utile mécanique qu'il fournit et la puissance électrique qu'il absorbe, de sorte que :

$$\eta = \frac{P_U}{P_a} < 1 \quad (\text{I.20})$$

À l'aide de la chaîne des pertes, on peut aussi écrire :

$$\eta = \frac{P_U - \text{pertes}}{P_a} \quad (\text{I.21})$$

I.9. Les vitesses du moteur asynchrone

I.9.1. Vitesse de synchronisme

C'est la vitesse de rotation du champ tournant.

$$n_s = \frac{60 f}{p} \quad (\text{I.22})$$

Avec :

n_s : vitesse de synchronisme (vitesse du champ tournant) (*tr/min*).

f : la fréquence du réseau (**Hz**).

p : nombre de paires de pôles.

I.9.2. Vitesse de rotation du moteur

C'est la vitesse de rotation de la machine.

$$n = n_s(1 - g) \quad (\text{I.23})$$

avec :

n_s : vitesse de synchronisme (*tr/s* ou *tr/min*).

n : vitesse de rotation du rotor (l'arbre) dans la même unité que n_s .

g : glissement généralement exprimé en pourcentage (%).

I.9.3. Vitesse angulaire

La vitesse angulaire du moteur asynchrone est exprimée par :

$$\Omega = 2 \pi n \quad (\text{I.24})$$

avec :

n : vitesse ou fréquence de rotation (tr/s ... parfois Hz)

Ω : vitesse angulaire (rad/s).

I.9.4. Le Couple

Le couple « C » d'un moteur asynchrone est fonction de la puissance utile P_U et de la vitesse de rotation N du moteur. Il s'exprime par la relation suivante :

$$C = \frac{P_U \times 9550}{N} \quad (I.25)$$

Avec :

P_U : Puissance utile du moteur (Puissance du moteur) (W).

N : la vitesse de rotation du moteur (tr/min).

I.10. Avantages et inconvénients d'un moteur asynchrone et Domaine d'utilisation

I.10.1. Les Avantages

Il ya plusieurs avantages dans le moteur asynchrone on peut citer :

1. Structure simple.
2. Robuste et facile à construire.
3. Utilisé dans la puissance moyenne et élevés.
4. Relié directement aux réseaux industriels à tension et fréquence.
5. Il tourne à vitesse variable différente de la vitesse synchrone.
6. Il est utilisé pour la réalisation de la quasi-totalité de l'entraînement à vitesse.

I.10.2. Les Inconvénients

Parmi les inconvénients de la machine asynchrone :

1. Le couple de démarrage très élevé que les couples nominales.
2. La vitesse dépend de la charge.
3. Variation de vitesse (nécessité d'un variateur de vitesse).
4. La structure dynamique est fortement non linéaire et l'existence d'un fort couplage

I.10.3. Domaine d'utilisation

Les machines asynchrones sont les machines alternatives les plus répandues en les utilise dans nombreux dispositifs :

1. Les stations de pompage.
2. Les appareils de levage.
3. Les ascenseurs.
4. Les machines outils de base pour le travail des métaux ou du bois.
5. Les dispositifs de manutention comme les tapis roulants.
6. La ventilation des locaux climatiques.

I.11. Capteur de vitesse (encodeur incrémental)

Le capteur de vitesse utilisé dans notre banc d'essai est un encodeur incrémental optique. Avant de présenter notre encodeur, nous allons citer ci-dessous sa définition et son principe de fonctionnement. L'encodeur incrémental est destiné à des applications où l'information de position ou de vitesse est obtenue par mesure du déplacement de l'objet. Le codeur délivre un train d'impulsions dont le nombre permet de déduire la valeur du déplacement ainsi que la vitesse car ce dernier est proportionnel à la fréquence des impulsions. Il est constitué d'un disque comportant deux à trois pistes : A, B et Z. La figure I.30 montre les composants fondamentaux d'un encodeur.

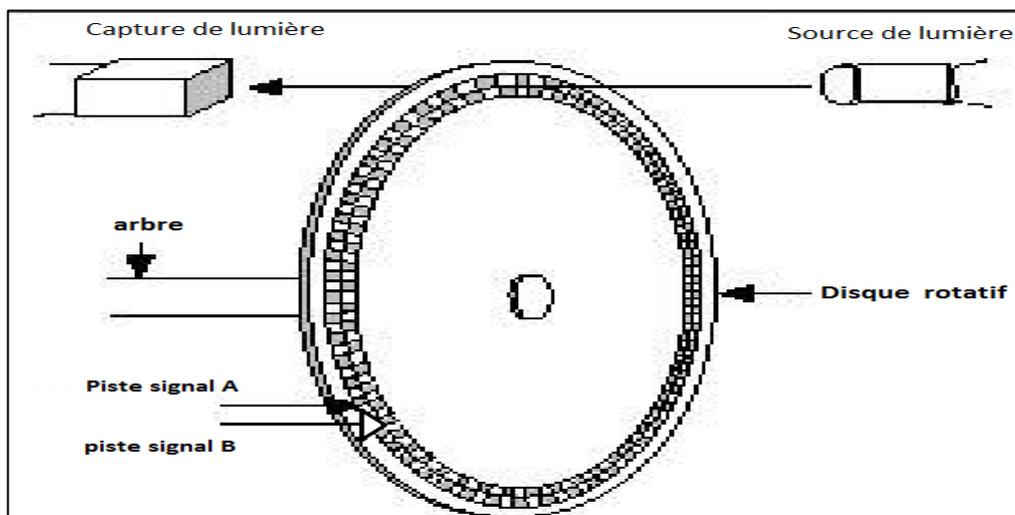


Figure I.15. Les composants fondamentaux d'un encodeur.

II.11.1. Principe de fonctionnement

Un encodeur est un appareil électromécanique capable de mesurer un mouvement ou une position ou de vitesse. Un encodeur rotatif composé d'une diode électroluminescente (LED), d'un disque et d'un détecteur de lumière situé du côté opposé du disque. Le disque contient plusieurs trous, qui est monté sur l'arbre en rotation, Les encodeurs rotatifs sont utilisés pour mesurer le mouvement de rotation d'un arbre Quand le disque commence à tourner L'émetteur

optique envoie des ondes lumineuses qui traversent des trous situés sur le disque pour être reçues par le récepteur. La plupart des encodeurs utilisent des capteurs optiques pour fournir des signaux électriques sous forme de trains d'impulsions, qui peuvent à leur tour être traduits en mouvement ou, direction ou position ou de vitesse. Le rôle des pistes A et B c'est de déterminer le sens de rotation de l'arbre de la machine avec lequel est lié l'encodeur de telle sorte que si A est en avance de phase par rapport à B, la machine tourne dans le sens 1 et dans le cas contraire c'est le sens 2.

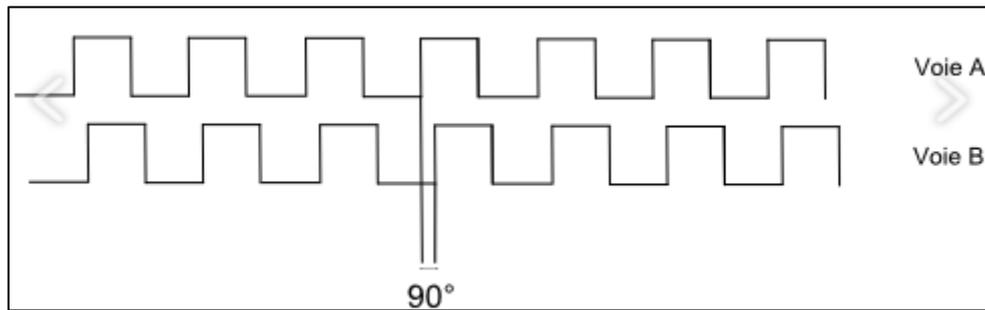


Figure I.16. Signal généré par un encodeur rotatif incrémental.

II.11.2. Description de l'encodeur utilisé

a. Présentation

L'encodeur incrémental optique utilisé dans notre banc d'essai est LPD3806-400BM-G5-24C. Type biphase phase A / phase B, sens de rotation basé sur la phase des deux signaux, Il génère un signal d'impulsion orthogonal biphase AB, 400 impulsions par tour pour chaque phase ce dernier délivre 1600 impulsions par tour pour une sortie biphase 4 fois, Output Sortie collecteur ouvert et peut être alimenté par une large plage de tensions d'alimentation qui est comprise entre 5 et 24 V.



Figure I.17. L'encodeur incrémental LPD3806-400BM-G5-24C.

b. Les Caractéristiques

Le tableau suivant montre les caractéristiques principales de l'encodeur incrémental LPD3806-400BM-G5-24C

Caractéristiques électriques	Caractéristiques mécanique
<ul style="list-style-type: none"> ➤ Tension d'alimentation 5V....24V. ➤ Résolution numérique sortie 400 imp. ➤ Capacité de charge des sorties 27V/40Ma. 	<ul style="list-style-type: none"> ➤ Vitesse de rotation maximale de 5000 tr/min ➤ Diamètre de l'axe 6,0 mm ➤ Dimensions globales du capteur d38×54 mm ➤ Longueur de câble 140 cm ➤ Matière : Aluminium

Tableau I.1. Caractéristiques de l'encodeur incrémental LPD3806-400BM-G5-24C.

c. Correspondance de câblage

Couleur de fil	Fonction
Rouge	Entrée d'alimentation (5V à 24V)
Noire	GND
Vert	Sortie de signal de phase A
Blanc	Sortie de signal de phase B

Tableau I.2. Correspondance de câblage.

I.12. Conclusion

Dans ce chapitre nous avons présenté le moteur asynchrone par donner sa définition, la description de constitution de machine asynchrone, nous a permis d'avoir une idée claire sur les différents éléments qui la compose, on a expliqué leur principe de fonctionnement par la suite on a expliqué le bobinage d'un moteur asynchrone et leur couplage (étoile ,triangle) ainsi que nous avons donner leur bilan de puissance et leur vitesse.

Avant de finir ce chapitre nous avons étudié les avantages et les inconvénients du moteur asynchrone et leur domaine d'utilisation.

A la fin de ce chapitre nous avons vu le capteur de vitesse utilisé dans notre projet réalisé est un encodeur incrémental optique, nous avons donné sa définition et son principe de fonctionnement.

CHAPITRE II

Les variateurs de vitesse

II.1. Introduction

Un variateur de vitesse est un appareil électronique permettant la régulation ou la commande de la vitesse des moteurs électriques, il converti en grandeur variable, la tension, le courant, la fréquence du réseau.

Dans ce chapitre on va présenter le variateur de vitesse SINAMIC V20 utilisé dans notre banc d'essai, leur mise en marche et leur réglage de ces paramètres.

II.2. Principe de variation de vitesse d'un moteur asynchrone

II.2.1. Action sur les paires de pôles

D'après l'équation $n_s = \frac{60f}{p}$ on constate qu'on peut varier la vitesse en agissant sur les paramètres f (la fréquence) ou p (nombre de paires de pôles).

On s'aperçoit que d'après les formules précédentes la fréquence de rotation d'un moteur asynchrone triphasé est lié au nombre de paires de pôles p du bobinage.

- Si le nombre de paires de pôles augmente, la fréquence de rotation d'un moteur diminue.
- Si le nombre de paires de pôles diminue, la fréquence de rotation d'un moteur augmente.

II.2.2. Action sur la fréquence d'alimentation du moteur

Toujours d'après les formules précédentes, on s'aperçoit que la fréquence de rotation d'un moteur asynchrone triphasé est directement liée à la fréquence f du réseau l'alimentant.

- Si la fréquence **augmente**, la fréquence de rotation d'un moteur **augmente**.
- Si la fréquence **diminue**, la fréquence de rotation d'un moteur **diminue**.

Dans notre projet, nous nous intéressons à la variation de fréquence.

II.3. La variation de la fréquence

La solution passe par l'utilisation de variateur de fréquence, plus communément appelé variateur de vitesse. L'utilisation d'un tel convertisseur permet d'obtenir une plage de variation de vitesse allant de 0 ($f=0\text{Hz}$) à la vitesse nominale du moteur ($f=50\text{Hz}$). On peut même faire fonctionner le moteur en survitesse si la fréquence dépasse les 50 Hz .

La vitesse du champ magnétique et donc la vitesse de rotation d'un moteur asynchrone dépend directement de la fréquence de la tension d'alimentation c'est sur ce paramètre que le variateur va agir. Le principe général étant de fournir un courant à amplitude et à fréquence variable tout en maintenant une tension constante.

II.4. Variateur de vitesse

Un variateur de vitesse est un équipement électrotechnique alimentant un moteur électrique de façon à pouvoir faire varier sa vitesse de manière continue, de l'arrêt jusqu'à sa vitesse nominale. La vitesse peut être proportionnelle à une valeur analogique fournie par un potentiomètre, ou par une commande externe : un signal de commande analogique ou numérique, issue d'une unité de contrôle [8].

Le variateur de vitesse est constitué de quatre composantes principales :

1. Redresseur (diodes et/ou thyristors).
2. Circuit intermédiaire (filtre).
3. Onduleur.
4. Un circuit de commande (Interface utilisateur).

II.4.1. Symbole de variateur de vitesse

Le symbole fonctionnel d'un variateur de vitesse électronique deux sens de rotation pour moteur asynchrone triphasé est le suivant :

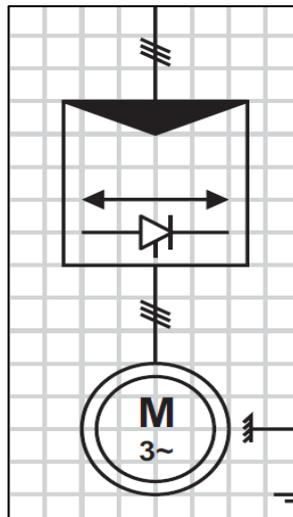


Figure II.1. Symbole fonctionnel d'un variateur de vitesse électronique deux sens.

II.4.2. Constitution et principe de fonctionnement d'un variateur de vitesse

Un variateur de vitesse est constitué d'un redresseur, d'un filtre, d'un onduleur et un circuit de commande.

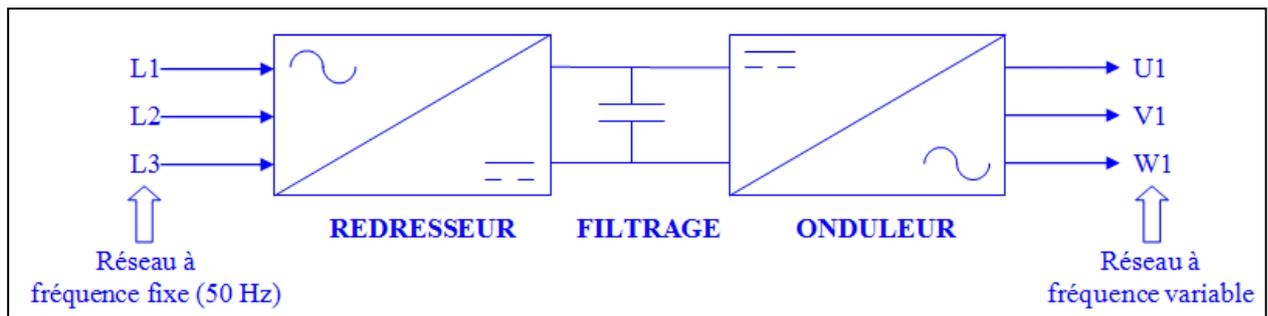


Figure II.2. Structure d'un variateur de vitesse.

II.4.2.1. Circuit de commande

L'interface utilisateur lui permet de démarrer et d'arrêter le moteur ainsi que d'en ajuster la vitesse. Elle peut aussi inclure le fonctionnement du moteur en sens opposé, un sélecteur pour choisir entre commande manuelle ou automatique, respectivement locale ou à distance, de la vitesse. L'interface dispose d'un affichage donnant des informations sur le statut du moteur. Des touches permettent à l'utilisateur de communiquer avec l'interface. Des portes d'entrée ou sortie, sont souvent fournis pour connecter d'autres périphériques, signaux... Un port, par exemple série, sert à configurer le variateur de vitesse à parti d'un ordinateur [9].

II.4.2.2. Redresseur

Ce sont des convertisseurs alternatif-continu (AC/DC) qui permettent de convertir une tension alternative en une tension continue unidirectionnelle, s'ils sont commandés la valeur moyenne de la tension obtenue est alors réglable [10].

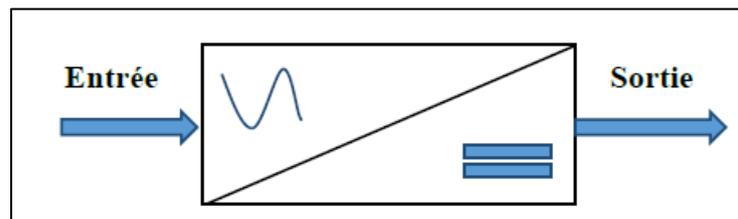


Figure II.3. Schéma de principe de redresseur.

Nous utilisons le pont triphasé à diodes alimenté par un système de tensions sinusoïdales triphasées, schématisé dans la Figure II.4. Les tensions simples des réseaux triphasés équilibrés directs sont :

$$V_a = V_m \sin(\omega t - \varphi) \quad (\text{II.1})$$

$$V_b = V_m \sin\left(\omega t - \varphi - \frac{2\pi}{3}\right) \quad (\text{II.2})$$

$$V_c = V_m \sin\left(\omega t - \varphi - \frac{4\pi}{3}\right) \quad (\text{II.3})$$

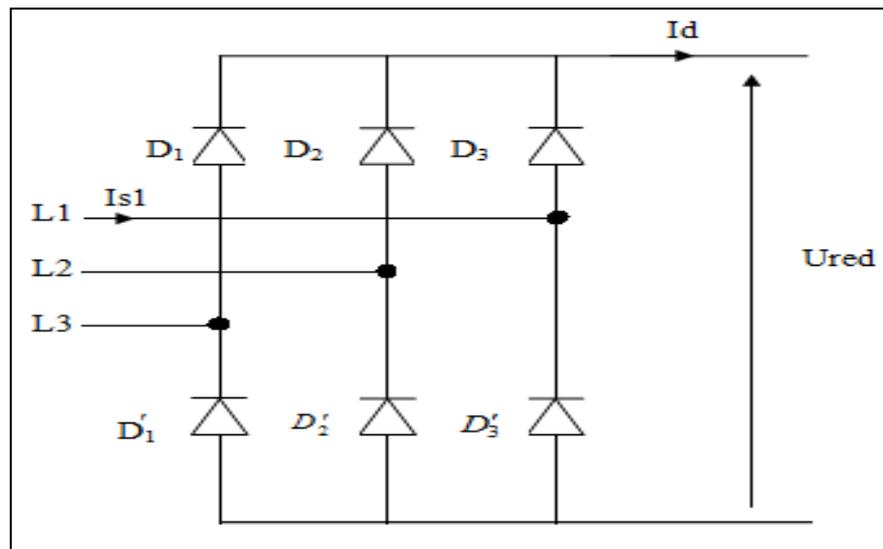


Figure II.4. Redresseur triphasé à diodes.

II.4.2.3. Etude du fonctionnement

Les trois diodes D_1 , D_2 , D_3 forment un commutateur plus positif, qui laisse passer à tout instant la plus positive des tensions, et les diodes D'_1 , D'_2 , D'_3 forment un commutateur plus négatif, qui laisse passer la plus négative des tensions. La tension redressée est à tout instant la différence entre ces deux tensions, soit :

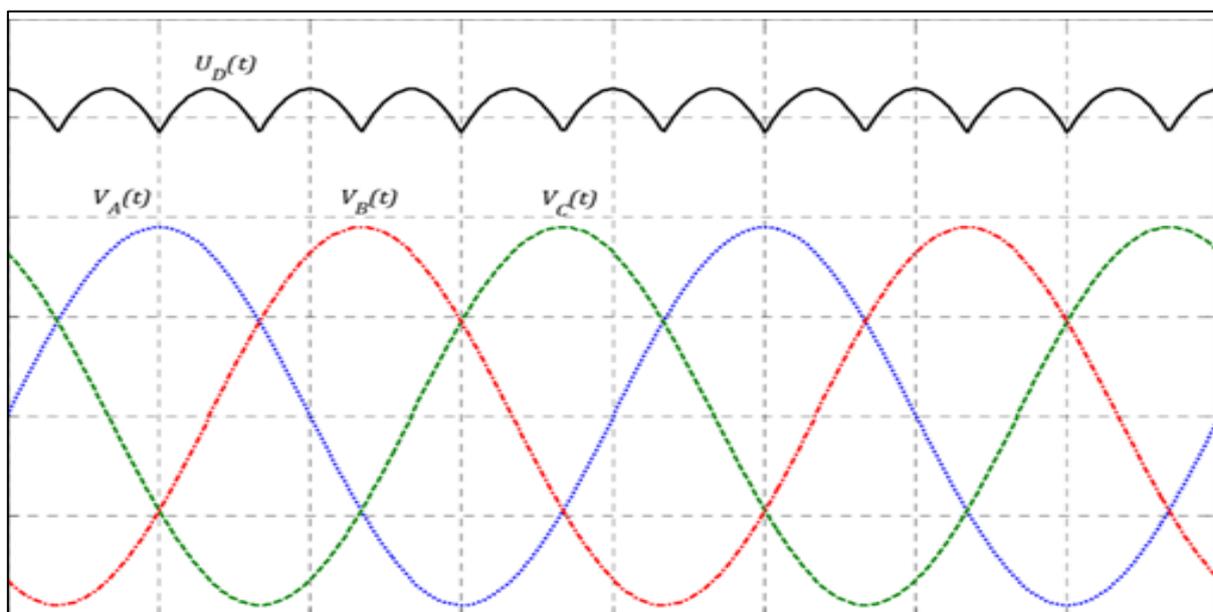


Figure II.5. Tension redressée.

Le rôle du redresseur est de transformer et de lisser la tension alternative en tension continue ou plus exactement en tension pseudo continue. En effet la tension continue obtenue n'est pas parfaite car elle comporte des ondulations résiduelles.

II.4.2.4. Circuit intermédiaire (filtre)

Pour filtrer la tension continue délivrée par le pont redresseur, on utilise le circuit RC comme indiqué sur la figure II.6.

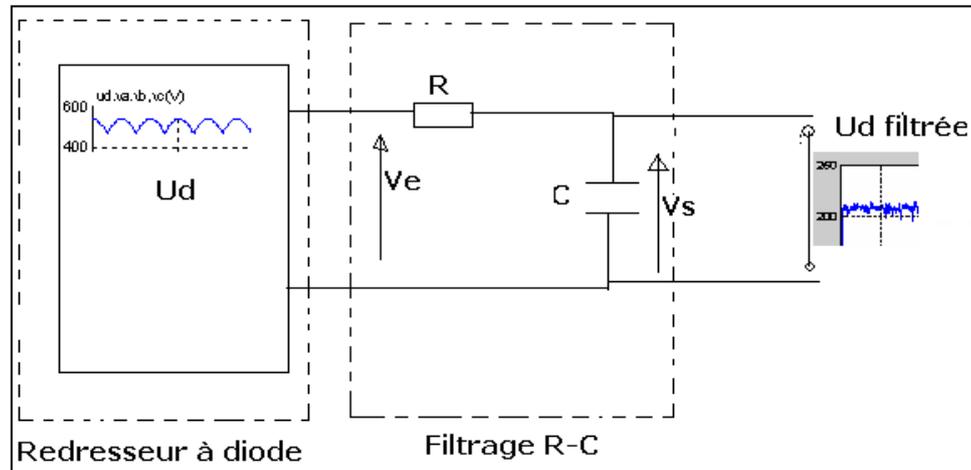


Figure II.6. Schéma du principe de filtrage.

Le rôle essentiel de ce filtre peut être résumé comme suit :

- Il dérive les harmoniques des courants créés par l'onduleur et empêche qu'ils ne soient injectés dans le réseau d'alimentation.
- Il évite les fluctuations importantes de la tension continue d'alimentation, que ces fluctuations viennent du redresseur (filtrage des harmoniques de la tension redressée) ou du débit instantané variable de l'onduleur, le maintien de cette tension continue apparaît comme un facteur favorisant la stabilité des montages variateurs de vitesse.
- Il limite la vitesse de croissance des courants de court-circuit, en freinant les grandeurs qui interviennent.

Pour le dimensionnement du filtre on prend en considération les paramètres suivant :

- ❖ la tension U_f d'alimentation de la charge.
- ❖ la composante continue du courant exigé I_f .
- ❖ l'ondulation maximale de ce courant et celle tolérée sur la tension U_f dans les plus mauvaises conditions [11].

II.4.2.5 Onduleur

Un onduleur est un convertisseur statique assurant la conversion d'énergie électrique de la forme continue (DC) à la forme alternative (AC). En fait, cette conversion d'énergie est satisfaite au moyen d'un dispositif de commande (semi-conducteurs). Il permet d'obtenir aux bornes du

récepteur une tension alternative réglable en fréquence et en valeur efficace, en utilisant ainsi une séquence adéquate de commande.

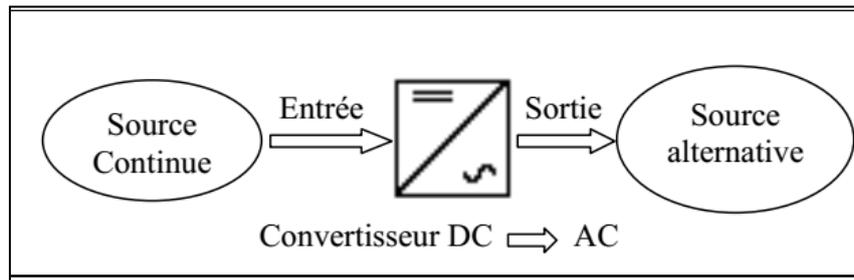


Figure II.7. Schéma de principe de l'onduleur.

L'onduleur est la dernière partie du variateur de vitesse située avant le moteur. C'est ici qu'a lieu le dernier réglage de la tension.

II.4.2.6 Onduleur de tension triphasé

On peut réaliser un onduleur triphasé en regroupant, en parallèle, trois onduleurs monophasés (en pont ou en demi-pont) et commander les interrupteurs de chacun pour obtenir à la sortie trois phases décalées de 120° . En fait, en regroupant trois demi ponts monophasés, on obtient un onduleur en pont triphasé à six interrupteurs représenté par la figure II.8 dont les interrupteurs du même bras de l'onduleur doivent être complémentaires pour que la tension continue U ne soit jamais en court-circuit et pour que les circuits des courants i_a , i_b et i_c ne soient jamais ouverts. Afin que les interrupteurs puissent imposer les tensions de sortie, quelques soient les courants de charge, il faut que ces derniers soient bidirectionnels en courant. Chacun d'eux est formé d'un semi-conducteur à ouverture et à fermeture commandés et d'une diode montée en antiparallèle. Ces interrupteurs peuvent être réalisées, suivant la puissance à contrôler, avec des transistors MOS ou bipolaires, des IGBTs ou des GTOS associés à une diode en antiparallèle pour obtenir la réversibilité en courant.

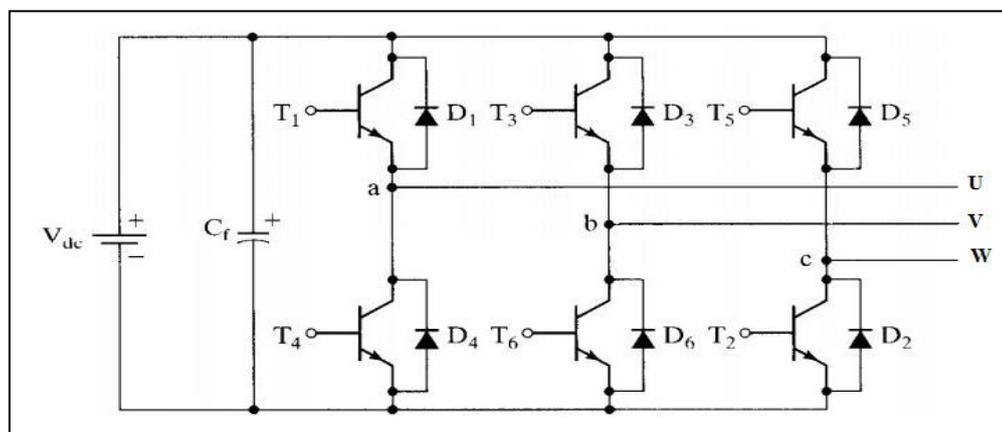


Figure II.8. Circuit de puissance d'un onduleur de tension triphasé.

La forme de la tension obtenue introduit des harmoniques importants qui perturbent le fonctionnement du récepteur, dans notre cas, elles perturbent le fonctionnement du moteur asynchrone.

Pour faire face à ce problème, il faut faire **fonctionner l'onduleur en MLI**.

II.4.2.7. La commande MLI

Le terme MLI signifie : Module de Largeur d'Impulsion. Son principe est que la tension continue d'entrée de l'onduleur est hachée de façon à créer une succession de créneaux d'amplitude égale à celle-ci, mais de largeur variable. La figure II.9. Présente un schéma d'alimentation pour le moteur asynchrone avec un onduleur de tension alimenté à partir d'un réseau triphasé.

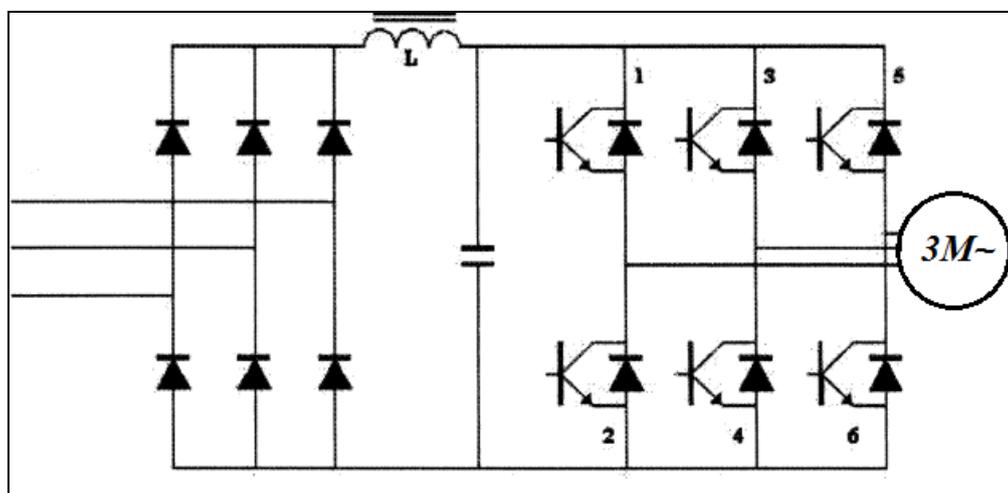


Figure II.9. Schéma d'alimentation d'un MAS avec un onduleur de tension triphasé.

Le filtre LC, associé au pont redresseur à diodes continue une source de tension réversible au courant. L'énergie ne peut donc pas transiter du moteur asynchrone au réseau. L'ensemble de transistors constitue l'onduleur triphasé à MLI, qui impose la fréquence du champ tournant et l'amplitude du courant dans la machine. Le moteur, inductif par nature, lisse le courant. Ce dernier est pratiquement sinusoïdal. La rotation du moteur est donc régulière et sans à coups.

La Modulation de Largeur d'Impulsion (en anglais PWM) peut utiliser plusieurs algorithmes, la méthode intersective aussi appelé « MLI sinus-triangle » est la plus ancienne et la plus utilisée. La figure ci-dessous illustre cette dernière méthode. Elle consiste à comparer une onde modulatrice (le signal à synthétiser) à une onde triangulaire. Le signal de sortie vaut **1** si la modulante est plus grande que l'onde triangulaire, sinon c'est **0**. Ce dernier permet, par l'intermédiaire de transistors de puissance, le pilotage d'une phase de la machine. Les autres phases sont pilotées par des ensembles identiques, déphasés à **120°**.

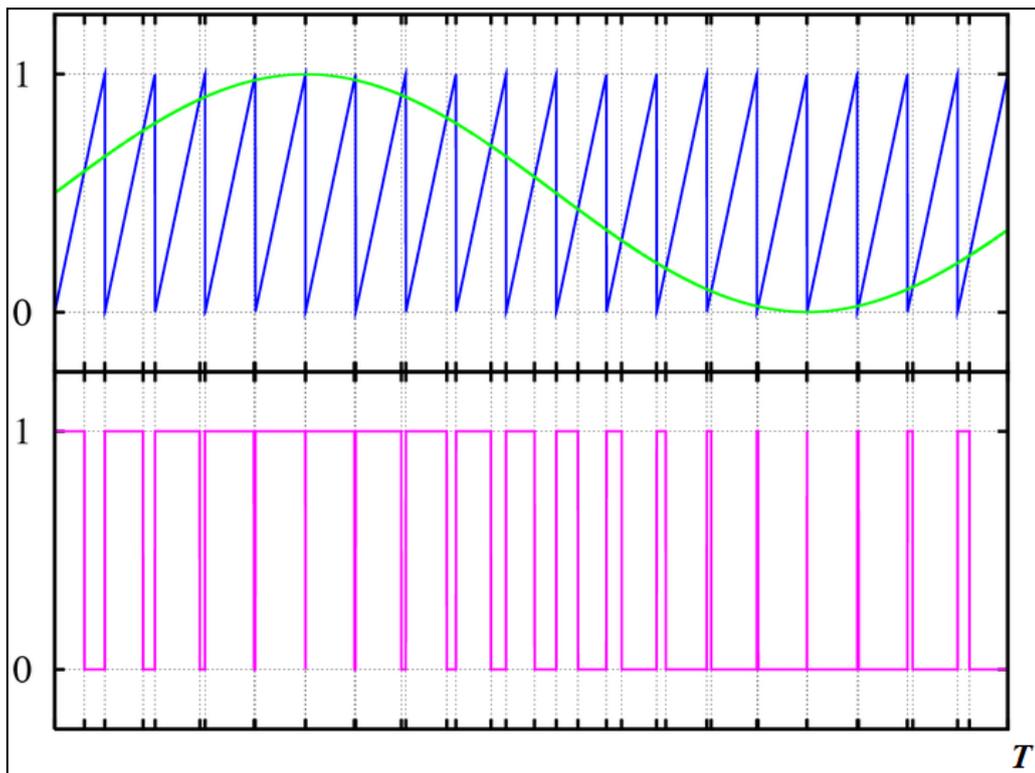


Figure II.10. Modulation Sinus-Triangle.

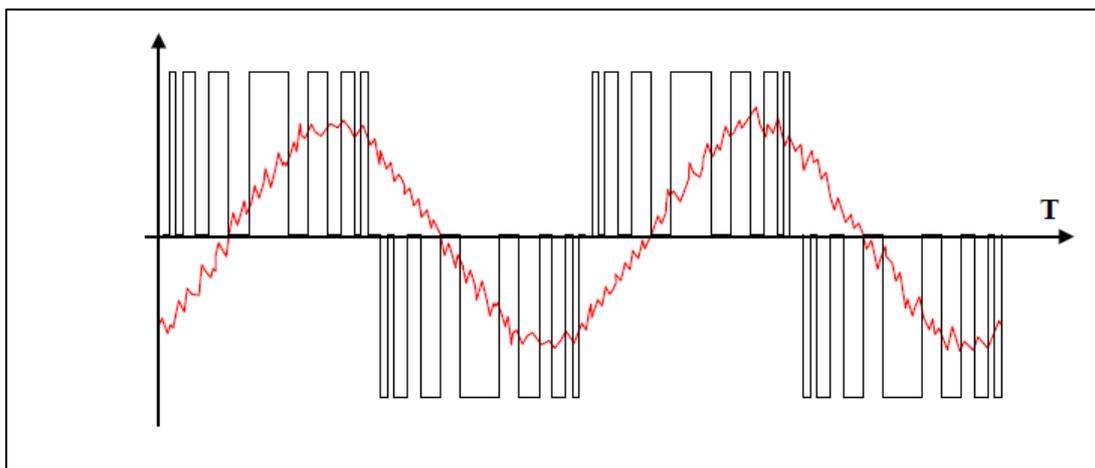


Figure II.11. Allure du courant pour $f = 50\text{Hz}$.

II.5. Fonction des variateurs de vitesse

Parmi la multitude de possibilités de fonctions qu'offrent les variateurs de vitesse actuels, on épinglera [12]:

- L'accélération contrôlée.
- La décélération contrôlée.
- La variation et la régulation de vitesse.

- L'inversion du sens de marche.
- Le freinage d'arrêt.
- La Protection.

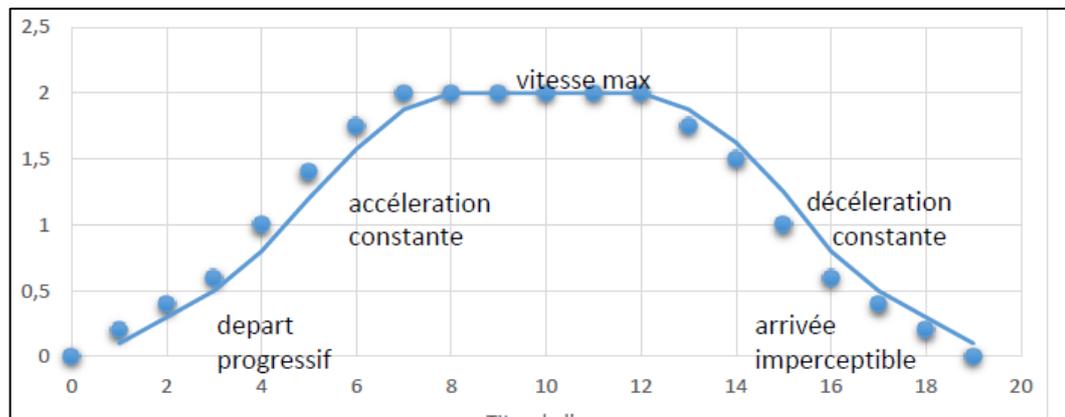


Figure II.12. Courbe graphique de fonctionnement de variateur.

II.6. Le variateur de vitesse SIEMENS SINAMICS V20

Le variateur de vitesse de marque **SIEMENS** sont trop utilisés dans l'industriel, parmi ces variateurs il y a la gamme **SINAMICS V20**. La figure II.13 représente le variateur de vitesse de type SIEMENS SINAMICS V20.



Figure II.13. Le variateur de vitesse SINAMICS V20.

II.6.1. Mise en service

La figure II.14 représente le Basic Opérateur Panel (BOP) intégré dans le variateur V20 [14].

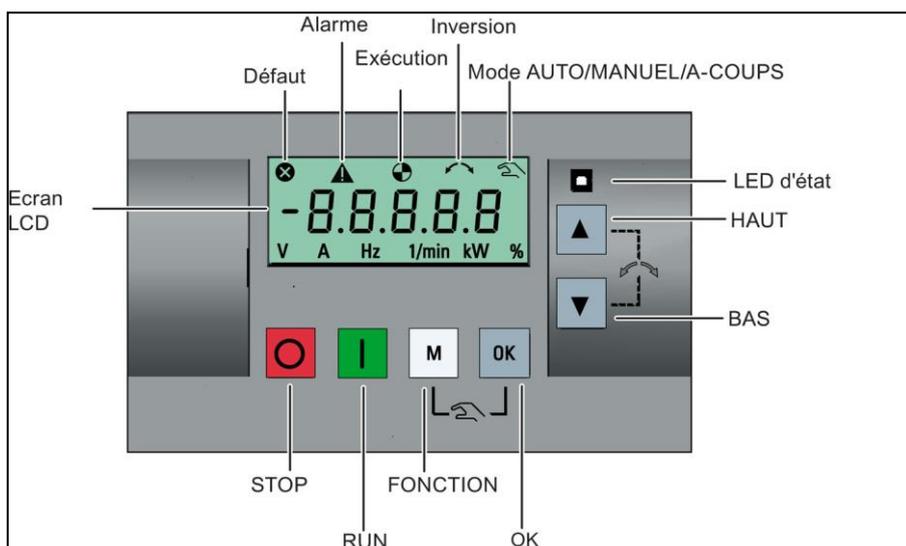


Figure II.14. Basic Opérateur Panel (BOP) intégré.

II.6.2. Fonctions des touches [14]

	Arrête le variateur	Réaction sur stop ARRET1 : le variateur immobilise le moteur en l'espace du temps de descente	
	Démarré le variateur	Si le variateur est démarré en mode Manuel / Marche par à-coups, l'icône de fonctionnement du variateur () s'affiche.	
	Bouton multifonction	Pression brève (< 2 s)	<ul style="list-style-type: none"> ➤ Accède au menu de paramétrage ou passe à l'écran suivant. ➤ Relance l'édition chiffre par chiffre pour l'élément sélectionné.
		Pression longue (> 2 s)	<ul style="list-style-type: none"> ➤ Retour à l'écran d'état. ➤ Accède au menu de configuration.
		Pression brève (< 2 s)	<ul style="list-style-type: none"> ➤ Commute entre les valeurs d'état. ➤ Accède au mode d'édition de valeur ou passe au chiffre suivant. ➤ Efface les défauts.
		Pression longue (> 2 s)	<ul style="list-style-type: none"> ➤ Edition rapide du numéro ou de la valeur de paramètre.
	+	<ul style="list-style-type: none"> ➤ Basculer entre les différents modes <div style="text-align: center;"> <pre> graph LR A[Mode auto (Aucune icône)] -- "M + OK" --> B[Mode Manuel (Avec icône de main)] B -- "M + OK" --> C[Mode marche par à-coups (Avec icône de main clignotante)] C -- "M + OK" --> A </pre> </div>	

	<ul style="list-style-type: none"> ➤ Lors de la navigation dans un menu, ce bouton permet de déplacer la sélection vers le haut parmi les écrans disponibles. ➤ Lors de l'édition de la valeur d'un paramètre, il augmente la valeur affichée. ➤ Lorsque le variateur est en mode RUN, il augmente la vitesse. ➤ Une pression longue (> 2 s) sur cette touche fait défiler rapidement les numéros des paramètres, indices ou valeurs vers le haut.
	<ul style="list-style-type: none"> ➤ Lors de la navigation dans un menu, ce bouton permet de déplacer la sélection vers le bas parmi les écrans disponibles. ➤ Lors de l'édition de la valeur d'un paramètre, il diminue la valeur affichée. ➤ Lorsque le variateur est en mode RUN, il réduit la vitesse. ➤ Une pression longue (> 2 s) sur cette touche fait défiler rapidement les numéros des paramètres, indices ou valeurs vers le bas.
	<ul style="list-style-type: none"> ➤ Inverse le sens de rotation du moteur. Appuyer une fois sur les deux touches pour activer la rotation du moteur dans le sens inverse. Appuyer une nouvelle fois sur ces deux touches pour désactiver la rotation du moteur dans le sens inverse.

Tableau II.1. Les Fonctions des touches.

II.6.3. Icônes d'état du variateur [14]

	<ul style="list-style-type: none"> ➤ Le variateur a au moins un défaut en instance. 	
	<ul style="list-style-type: none"> ➤ Le variateur a au moins une alarme en instance. 	
		<ul style="list-style-type: none"> ➤ Le variateur est en marche (la vitesse du moteur peut être de 0 tr/min).
	 (clignotant)	<ul style="list-style-type: none"> ➤ Le variateur peut être traversé par le courant de manière inattendu.
	<ul style="list-style-type: none"> ➤ Le moteur tourne dans le sens inverse. 	
		<ul style="list-style-type: none"> ➤ Le variateur est en mode MANUEL.
	 clignotant	<ul style="list-style-type: none"> ➤ Le variateur est en mode marche par à-coups.

Tableau II.2. Les Icônes d'état du variateur.

II.6.4. Visualisation de l'état du variateur

Le menu d'affichage fournit une vue de base pour la surveillance de certains paramètres clés tels que la fréquence, la tension, le courant, etc [14].

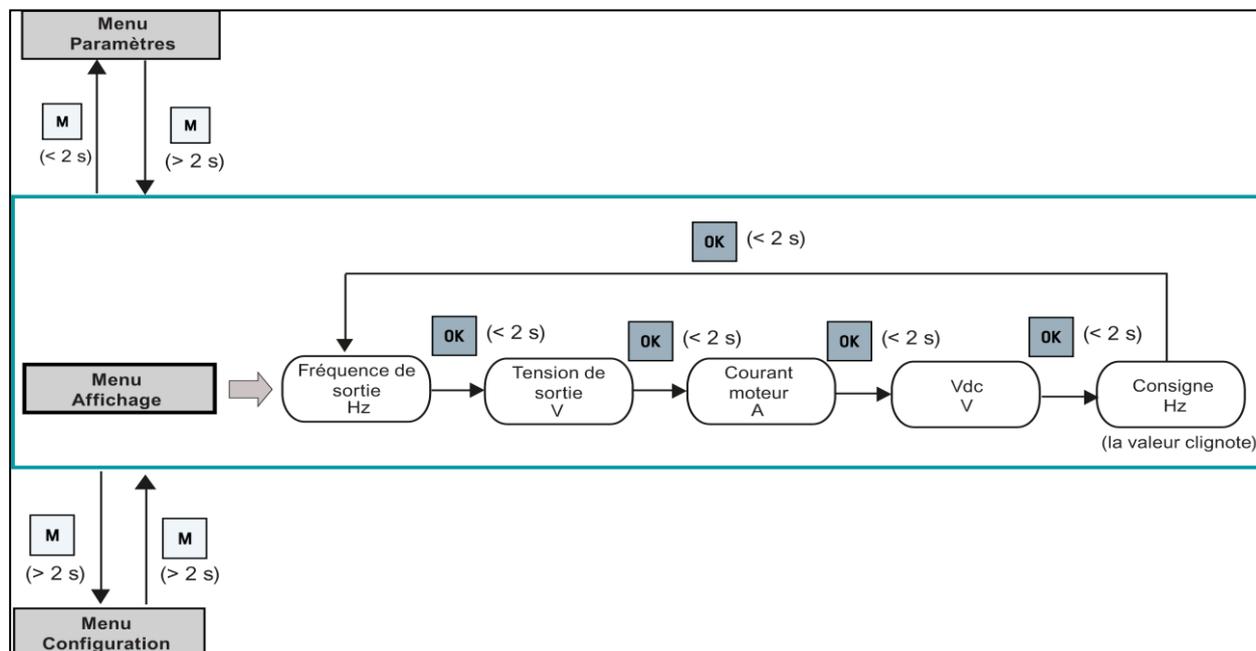


Figure II.15. Le menu d'affichage de variateur.

II.6.5. Etats des LED

Le SINAMICS V20 ne dispose que d'une seule LED pour les indications d'état. Cette LED peut être orange, verte ou rouge [14].

Etat du variateur	Couleur de LED	
Mise sous tension	Orange	
Prêt (aucun défaut)	Vert	
Mode de mise en service	Clignotement vert lent à 0,5 Hz	
Tous les défauts	Clignotement rouge rapide à 2 Hz	
Clonage de paramètres	Clignotement orange à 1 Hz	

Tableau II.3. Les états des LED.

II.6.6. Démarrage du moteur pour un essai de fonctionnement

Cette section explique comment démarrer le moteur pour un essai de fonctionnement afin de vérifier que la vitesse et le sens de rotation du moteur sont corrects. Le moteur peut être démarré en mode Manuel ou Marche par à-coups [14].

❖ Démarrage du moteur en mode MANUEL

1. Appuyer sur pour démarrer le moteur.
2. Appuyer sur pour arrêter le moteur.

❖ Démarrage du moteur en mode marche par à-coups

1. Appuyer sur + pour passer du mode Manuel au mode Marche par à-coups (l'icône clignote).
2. Appuyer sur pour démarrer le moteur. Relâcher pour arrêter le moteur.

II.6.7. Mise en service rapide

II.6.7.1. Structure du menu de configuration

Le menu de configuration vous guide à travers les étapes requises pour réaliser une mise en service rapide du système de variateur. Il se compose des quatre sous-menus suivants [14]:

	Sous-menu	Fonctionnalité
1	Paramètres moteur	Définit les paramètres nominaux du moteur pour une mise en service rapide.
2	Sélection des macros de connexion	Définit les macros requises pour les configurations de câblage standard.
3	Sélection des macros d'application	Définit les macros requises pour certaines applications Communes
4	Sélection des paramètres communs	Définit les paramètres requis pour optimiser les performances du variateur

Tableau II.4. La Structure du menu de configuration.

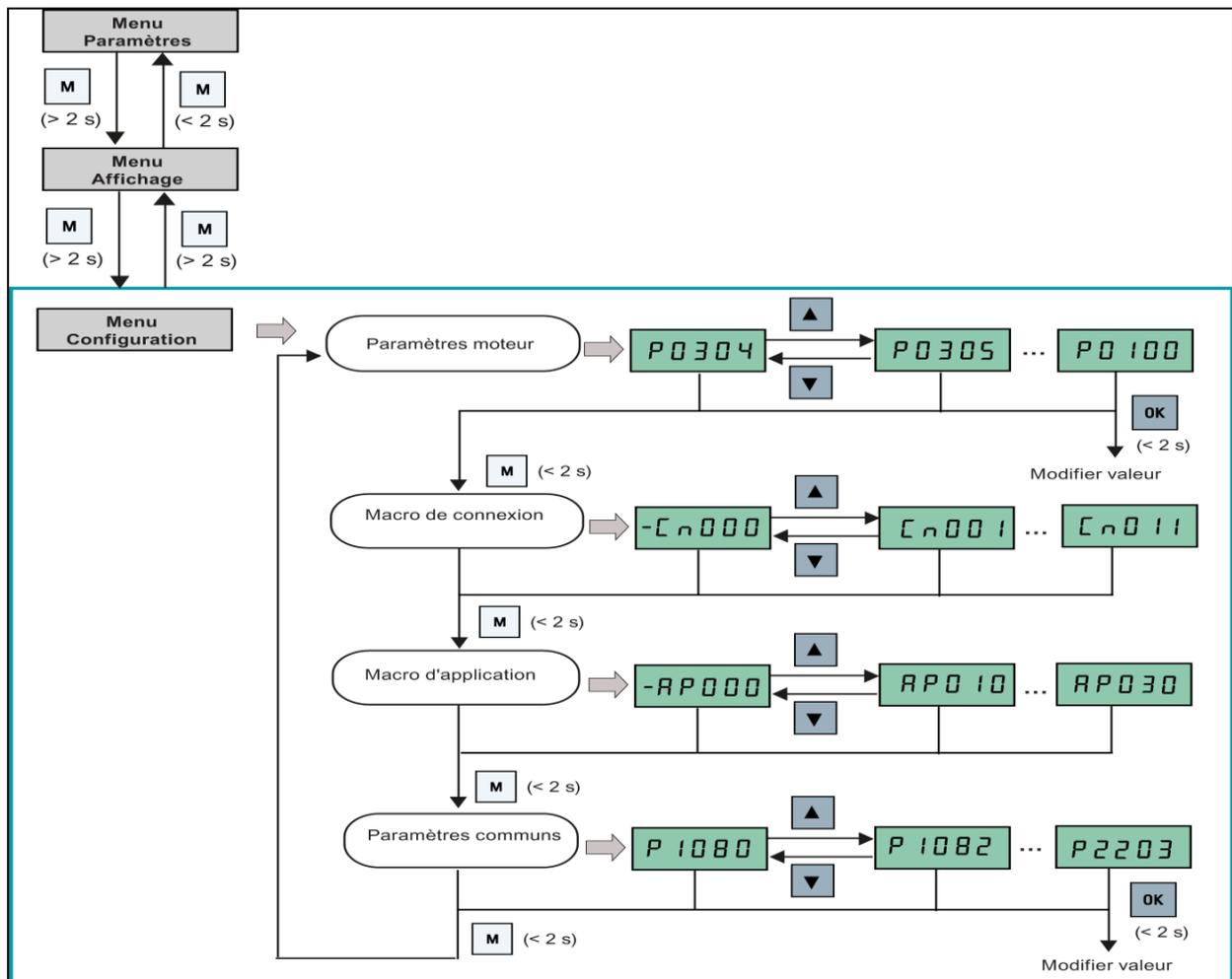


Figure II.16. La Structure du menu de configuration.

II.6.7.2. Réglage des paramètres moteur

Ce menu est conçu pour configurer facilement les caractéristiques nominales du moteur issues de la plaque signalétique [14].

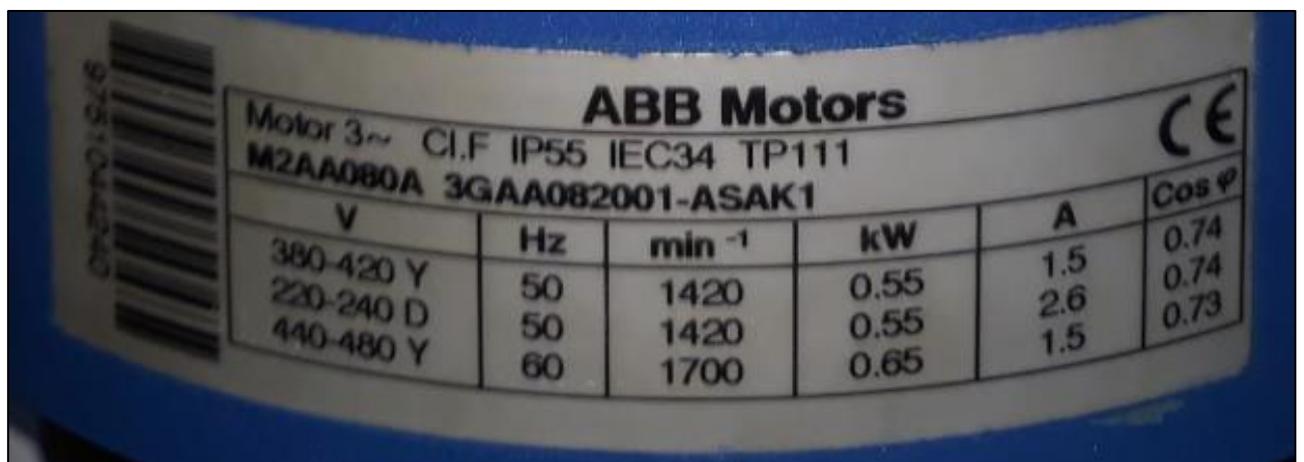


Figure II.17. La plaque signalétique du moteur utilisé.

Le tableau suivant représenté les paramètres moteurs

Paramètre	Fonction	Paramètres du moteur utilisé
p0100	Sélection 50/60 Hz = 0: Europe [kW], 50 Hz (réglages d'usine) = 1: Amérique du Nord [hp], 60 Hz = 2: Amérique du Nord [kW], 60 Hz	0
p0304	Tension assignée du moteur [V] Noter que les données de la plaque signalétique entrées doivent Correspondre au câblage du moteur (étoile/triangle).	220 [V]
p0305	Courant assigné du moteur [A] Noter que les données de la plaque signalétique entrées doivent correspondre au câblage du moteur (étoile/triangle).	2.6 [A]
p0307	Puissance assignée du moteur [kW/hp] Si p0100 = 0 ou 2, unité de puissance du moteur = [kW] Si p0100 = 1, unité de puissance du moteur = [hp]	0.55 [kW]
p0308	Facteur de puissance assigné du moteur (cosφ) Visible uniquement lorsque p0100 = 0 ou 2	0.74 (cosφ)
p0310	Fréquence assignée du moteur [Hz]	50 [Hz]
p0311	Vitesse assignée du moteur [tr/min]	1420 [tr/min]
p1900	Sélection de l'identification des paramètres moteurs = 0: Désactivé = 2: Identification de tous les paramètres à l'arrêt	0

Tableau II.5. Réglage des paramètres moteur.

II.6.7.3. Réglage des macros de connexion

Ce menu sélectionne quelle macro est requise pour les configurations de câblage standard. La macro par défaut est "Cn000" (macro de connexion 0). Il y a **11** macros nous les représentons dans le tableau suivant [14].

Macro de connexion	Description	Exemple d'affichage
Cn000	Réglage d'usine. Ne modifie aucun paramètre.	
Cn001	BOP comme unique source de commande	

Cn002	Commande via les bornes (PNP/NPN)	  Le signe moins indique que cette macro est actuellement sélectionnée.
Cn003	Vitesses fixes	
Cn004	Vitesse fixe en mode binaire	
Cn005	Entrée analogique et fréquence fixe	
Cn006	Commande par bouton-poussoir externe	
Cn007	Bouton-poussoir externe avec consigne analogique	
Cn008	Régulation PID avec référence par entrée analogique	
Cn009	Régulation PID avec valeur de consigne fixe	
Cn010	Régulation USS	
Cn011	Régulation MODBUS RTU	

Tableau II.6. La recréation des macros de connexion.

Après l'étude de ces tableaux qui représente les différentes macros connexion Nous avons choisi deux macros de connexion Cn003 et Cn010.

II.6.7.4. Réglage des macros d'application

Ce menu définit certaines applications communes. Chaque macro d'application fournit un jeu de paramètres pour une application spécifique. Après avoir sélectionné une macro d'application, les réglages correspondants sont appliqués au variateur pour simplifier le processus de mise en service. Il y a Cinq macros d'application macros nous les représentons dans le tableau suivant.

Macro d'application	Description	Exemple d'affichage
AP000	Réglage d'usine. Ne modifie aucun paramètre.	  Le signe moins indique que cette macro est actuellement sélectionnée.
AP010	Applications de pompe simples	
AP020	Applications de ventilateur simples	
AP021	Applications de compresseur	
AP030	Applications de convoyeur	

Tableau II.7. Réglage des macros d'application.

Après l'étude de ce tableaux qui représente les différents macros d'application Nous avons choisi le macro d'application AP010.

❖ Macro d'application AP010 – Applications de pompe simples [14].

Paramètre	Description	Valeur par défaut pour AP010	Commentaires
p1080	Fréquence minimale	15	Fonctionnement du variateur à une vitesse inférieure bloqué
p1110	Bloquer la consigne de fréquence négative	1	Rotation de pompe dans le sens inverse bloquée
p1120	Temps de montée	10	Temps de montée de zéro à la fréquence maximale
p1121	Temps de descente	10	Temps de descente de la fréquence maximale à zéro

Tableau II.8. Les paramètres Macro d'application AP010.

II.6.7.5. Réglage de paramètres communs

Ce menu fournit des paramètres communs pour optimiser les performances du variateur. Nous les représentons dans le tableau suivant [14].

Paramètre	fonction	Paramètre	fonction
p1080	Fréquence minimale du moteur	p1001	Consigne de fréquence fixe 1
p1081	Fréquence maximale du moteur	p1002	Consigne de fréquence fixe 2
p1120	Temps de montée	p1003	Consigne de fréquence fixe 3
p1121	Temps de descente	p2201	Consigne de fréquence PID fixe 1
p1058	Fréquence marche par à-coups	p2202	Consigne de fréquence PID fixe 2
p1060	Temps de montée de la marche par à-coups	p2203	Consigne de fréquence PID fixe 3

Tableau II.9. Réglage de paramètres communs.

II.7. Avantages et inconvénients d'un variateur de vitesse

II.7.1. Avantages

Le recours aux variateurs de vitesse offre plusieurs avantages [13]:

- ✓ Démarrage progressif des moteurs réduisant les chutes de tension dans le réseau et limitant les courants de démarrage.
- ✓ Amélioration du facteur de puissance.
- ✓ Précision accrue de la régulation de vitesse.
- ✓ Prolongement de la durée de service du matériel entraîné.
- ✓ Diminution de la consommation d'électricité.
- ✓ Économies d'énergie.

II.7.2. Inconvénients

Tous les variateurs de vitesse intégrant des dispositifs de commutation (diodes, thyristors, IGBT, etc.) forment une charge non linéaire qui engendre des courants harmoniques, sources de distorsion de l'onde (chute ou perturbation de la tension) dans le réseau électrique. Cette dégradation de l'onde peut perturber tant que les équipements électriques du client que ceux du réseau électrique si aucune mesure d'immunité n'est prise. Par ailleurs, des résonances harmoniques peuvent également apparaître entre les variateurs de vitesse et les batteries des condensateurs [13].

II.8. Conclusion

Après avoir terminé ce chapitre nous avons constaté que la vitesse d'un moteur asynchrone dépend principalement de sa fréquence et pour la varier on utilise les variateurs de vitesse. Un variateur de vitesse est composé principalement d'un redresseur et un filtre (qui converti un signal alternatif en signal continu) et d'un onduleur (qui converti un signal continu en un signal alternatif de fréquence variable). Le principe de la MLI utilisée dans l'onduleur consiste à appliquer aux enroulements du moteur une suite d'impulsions de tension, d'amplitude égale à la tension continue fournie par le redresseur. Les impulsions sont modulées en largeur de manière à créer une tension alternative d'amplitude variable.

Nous avons également parlé de le variateur de vitesse de la marque SIEMENS SINAMICS V20 utilisé dans notre banc d'essai et comment configuré les paramètres pour réaliser une mise en service rapide du système de variateur. Nous avons présenté les macros de connexion et le macro d'application Sur lequel nous avons travaillé.

CHAPITRE III

L'automate programmable Et l'interface Homme-Machine

III.1. Introduction

En tant qu'automaticien on a tendance à avoir une vue systémique sur les machines et les processus. Notre objectif consiste à automatiser et à asservir ces systèmes afin d'accroître la productivité, améliorer la qualité et la sécurité. Le plus important c'est de remplacer l'homme dans ces actions pénibles, délicates et répétitives. Dans ce chapitre, nous allons définir les systèmes automatisés et les parties qui constituent ces derniers.

III.2. Définition d'un système automatisé

Un système automatisé est toujours composé d'une Partie Commande et d'une Partie Opérative. Pour faire fonctionner ce système, l'opérateur (personne qui va faire fonctionner le système) va donner des consignes à la Partie Commande. Celle-ci va traduire ces consignes en ordres qui vont être exécutés par la Partie Opérative. Une fois les ordres accomplis, la Partie Opérative va le signaler à la Partie Commande (elle fait un compte-rendu) qui va à son tour le signaler à l'Opérateur. Ce dernier pourra donc dire que le travail a bien été réalisé.

III.3. Objectifs de l'automatisation

La compétitive de l'entreprise et des produits. Cette compétitivité passe par la qualité, la maîtrise des coûts. On cherche donc à améliorer la productivité. L'amélioration des conditions de travail, et surtout la sécurité, fait partie des objectifs de l'automatisation.

Les objectifs sont comme suite:

- Eliminer les tâches répétitives.
- Simplifier le travail de l'humain.
- Augmenter la sécurité (responsabilité).
- Accroître la productivité.
- Economiser les matières premières et l'énergie.
- S'adapter à des contextes particuliers: flexibilité.
- fournir les quantités nécessaires.
- produire à qualité constante.
- Améliorer la qualité.
- augmenter la productivité.
- améliorer les conditions de travail [17].

III.4. Structure d'un système automatisé

Un système automatisé peut se décomposer selon le schéma ci-dessous :

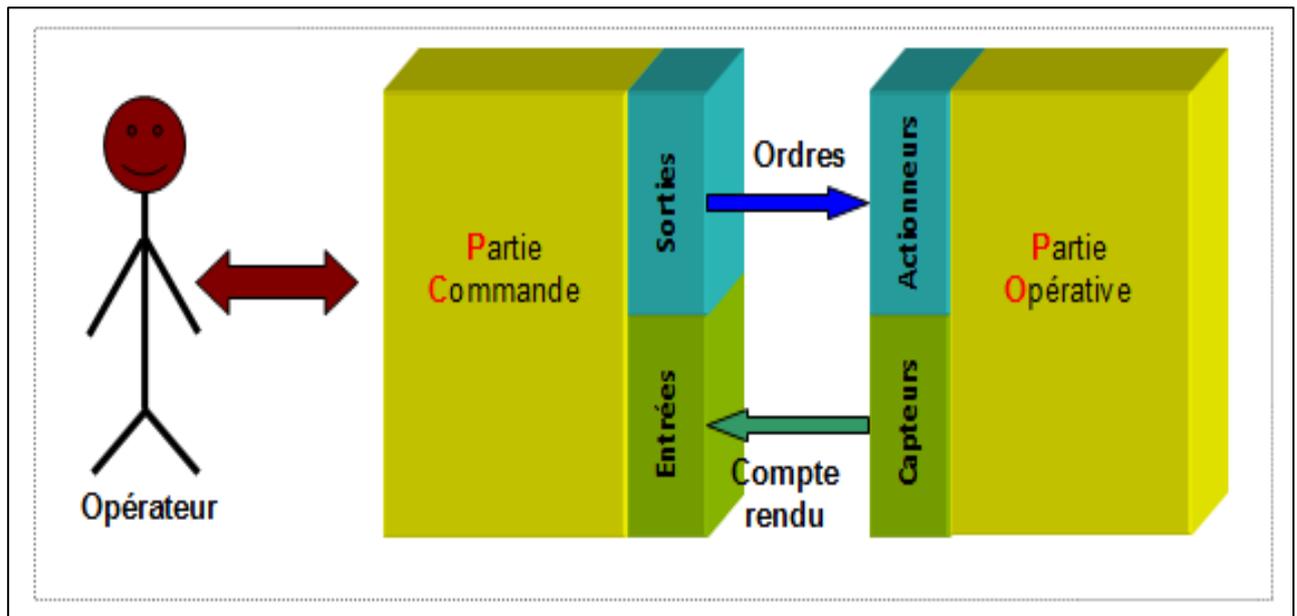


Figure III.1. Structure d'un système automatisé.

III.4.1. Partie commande

C'est la partie qui gère le fonctionnement du Système Automatisé. Elle est en général composée d'un ordinateur qui contient dans ses mémoires un programme. Elle transmet les ordres aux actionneurs de la partie opérative à partir :

- du programme qu'elle contient,
- des informations reçues par les capteurs,
- des consignes données par l'utilisateur ou par l'opérateur [18].

III.4.1.1. Unité de traitement d'informations

L'unité de traitement d'informations est un ensemble d'appareils (électriques, mécaniques ou biologiques) permettant de traiter automatiquement des informations. Cette opération peut être gérée par un **API**, une carte **Arduino** ou **DSP**... Dans notre système nous avons utilisé un API comme unité de traitement d'informations.

a. Automate programmable industriel

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique [19].

b. Structure matérielle des automates

L'automate programmable industriel possède deux structures **externe** et **interne**.

b.1 Structure externe

L'API peut être de type **compact** ou **modulaire**.

- ❖ **Automate compact** : Il intègre dans un seul boîtier le processeur, l'alimentation, et les entrées sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes [19].



Figure III.2. Automate compact.

- ❖ **Automate modulaire** : Le processeur, l'alimentation et les interfaces d'entrées sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires, possibilité d'ajouter une grande variété de modules spécialisés (solution adaptée aux besoins), aussi il est doté de grand nombre d'E/S.



Figure III.3. Automate modulaire.

b.2. Structure interne

La structure interne d'un API est assez voisine de celle d'un système informatique simple, Les API comportent quatre parties principales :

- ✓ Une alimentation 230 V, 50/60 Hz (AC) ; 24 V (DC).
- ✓ Unité centrale (Processeur).
- ✓ Interfaces d'entrées / sorties.
- ✓ Mémoires.

Ces quatre parties sont reliées entre elles par des bus (ensemble câble autorisant le passage de l'information entre ces 4 secteurs de l'API). Ces quatre parties réunies forment un ensemble compact appelé automate.

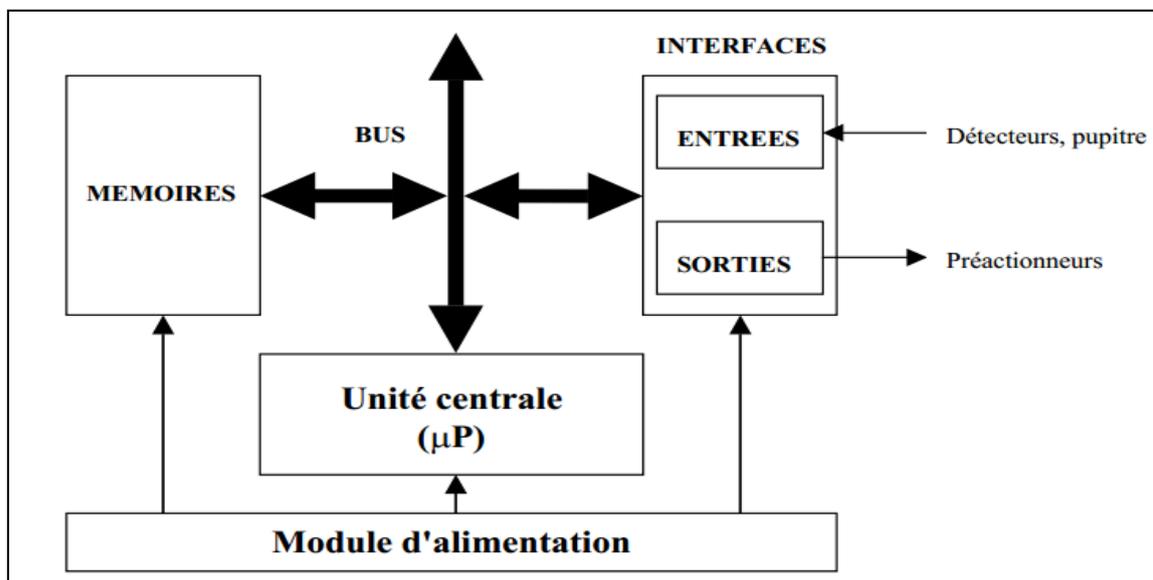


Figure III.4. Structure interne d'un API.

❖ Description des éléments d'un API

➤ L'alimentation :

L'unité d'alimentation est indispensable puisqu'elle converti une tension alternative en une base tension continue (24V, 5V...) nécessaire au processeur et aux modules d'entrées sorties.

➤ Unité centrale (Processeur)

L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Exemple : Si deux actions doivent être simultanées, l'API les traite successivement.

➤ Interfaces d'entrées / sorties

L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon des adresses de sortie. Chaque pré-actionneur est relié à une de ces adresses. Le nombre de ces entrées et sorties varient en fonction du type d'automate. Les cartes d'entrées et de sorties sont modulaires, la modularité varie entre 8, 16 et 32 voies.

- ☑ **Les entrées TOR** : L'information ne peut prendre que deux états (vrai/faux, 0 ou 1). Elles permettent de raccorder à l'automate les différents capteurs logiques tels que : boutons poussoirs, thermostats, fins de course, capteur de proximité....
- ☑ **Les sorties TOR** : Elles permettent de raccorder à l'automate les différents pré-actionneurs tels que : vannes, contacteurs, voyants, afficheurs.
- ☑ **Les entrées sorties numériques** : Utilisés pour les API haute ou moyenne gamme effectuant des traitements numériques. La longueur définit par la taille du mot mémoire de l'API (ex : 16 bits).
- ☑ **Les Entrées/Sorties analogiques** : Transforment une grandeur analogique en une valeur numérique et vice versa. La précision dépend du nombre de bits utilisés. Technologiquement, les EA/SA sont caractérisées par l'amplitude du signal analogique (typiquement 0/10V ou -10/+10V) et par le courant correspondant.

➤ La mémoire

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système, qui sont le terminal de programmation (PC ou console) et le processeur. Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates deux types de mémoires qui remplissent des fonctions différentes :

- **La mémoire Langage** où est stocké le langage de programmation. Elle est en général en lecture seulement (ROM : mémoire morte).
- **La mémoire Travail** utilisable en lecture-écriture pendant le fonctionnement, c'est la RAM (mémoire vive). Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).

c. Principe général de fonctionnement d'un API

L'automate lit en permanence et à grande vitesse des instructions du programme dans la mémoire. Selon la modification des entrées, il réalise les opérations logique entre les informations d'entrées et de sorties. Le temps de lecture d'un programme est pratiquement inférieur à 10ms. Ce temps est très inférieur au temps d'évolution d'une séquence.

d. Programmation d'API

La programmation d'un API consiste à traduire dans le langage spécialisé de l'automate, les équations de fonctionnement du système à automatiser. Parmi les langages normalisés, on cite quelques-uns des plus connus et plus utilisés

- Langage à contacts (LADDER) ;
- Langage List d'instructions (Instruction List) ;
- Langage GRAFCET (Sequential Function Chart : SFC) ;
- Langage littéral structuré.

Généralement, les constructeurs d'API proposent des environnements logiciels graphiques pour la programmation [19].

e. Critères de choix de l'automate

Il revient à nous d'établir le cahier des charges de notre système et de chercher sur le marché l'automate le mieux adapté à nos besoins. Cela est fait en considérant un certain nombre de critères importants [20]:

- ✓ Avoir les compétences et l'expérience nécessaire pour programmer la gamme d'automate.
- ✓ Le nombre et le type d'entrées et de sorties nécessaires.
- ✓ La communication envisagée avec les autres systèmes.
- ✓ Les capacités de traitement de la CPU.
- ✓ Les moyens de sauvegarde.
- ✓ La fiabilité et la robustesse.
- ✓ Le cout d'investissement, de fonctionnement, de maintenance de l'équipement.
- ✓ La qualité du service après-vente.

En tenant compte des points mentionnés ci-dessus, et pour résoudre les problèmes rencontrés, on choisit l'automate **S7-1200** qui est la nouvelle gamme de **SIEMNS**.

f. Présentation de l'automate S7-1200

L'API **S7-1200** offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre à vos besoins en matière d'automatisation. Sa forme *compacte*, sa

configuration souple et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées.

La CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant.

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, GPRS, RS 485 ou RS 232 [21].



Figure III.5. L'automate programmable S7-1200.

f.1 Principe de fonctionnement du S7-1200

Une fois le programme chargé, la CPU contient la logique nécessaire au contrôle et à la commande des appareils dans l'application. Il surveille les entrées et modifie-les sorties conformément à la logique du programme utilisateur, qui peut contenir des instructions booléennes, de comptage, de temporisation, ou mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents.

f.2 Technologie intégrée de S7-1200

➤ Entrées rapides

Le nouvel automate SIMATIC S7-1200 comporte jusqu'à six compteurs high-speed. Trois entrées à 100 kHz et trois autres entrées à 30 kHz, sont intégrées en continu pour des fonctions de comptage et de mesure.

➤ Sorties rapides

Deux sorties rapides pour des trains d'impulsions de 100 kHz sont également intégrées et permettent de piloter la vitesse et la position d'un moteur pas à pas ou d'un actionneur.

f.3 Choix de la CPU

Les CPU du système SIMATIC S7-1200 se déclinent en cinq classes de performances : CPU 1211C, CPU 1212C et CPU 1214C, CPU 1215C et CPU 1217C, chacune d'elles pouvant être étendue en fonction des besoins de la station. Les CPU's de l'automate S7-1200 sont données dans le tableau suivant :

CPU	CPU 1211C	CPU 1212C	CPU1214C	CPU 1215C	CPU 1217C
Mémoire de travail	50 ko	75 ko	100 ko	125 ko	150 ko
E/S TOR	6 entrées 4 sorties	8 entrées 6 sorties	14 entrées 10 sorties	14 entrées 10 sorties	14 entrées 10 sorties
E/S analogiques	2 entrées	2 entrées	2 entrées	2 entrées 2 sorties	2 entrées 2 sorties
Modules E/S extensible	Aucune	2 modules	8 modules	8 modules	8 modules
Module de communication (CM) (extension vers la gauche)	3 modules	3 modules	3 modules	3 modules	3 modules
Port de communication Ethernet PROFINET	1 porte	1 porte	1 porte	2 portes	2 portes

Tableau III.1. Tableau de comparaison des CPU S7-1200.

Après avoir étudié notre système quand doit réalisé et après la comparaison entre les CPU disponibles, on a choisit la CPU 1212C de référence **6ES7212-1HE31-0XB0** qui réponde a nous besoins (Figure II.5).

- Le nombre d'entries sortie selon nous besoins.
- Il contient Les compteurs rapides (HSC) qui nous avons utilisé pour calculer la vitesse d'après les impulsions qui délivre par l'encodeur.
- Il a la capacité d'ajouter trois modules de communication Parmi eux le module utilisé dans notre projet (RS422/485).

- le cout



Figure III.6. La CPU S7-1212C.

g. Modes de fonctionnement de la CPU : La CPU a les modes de fonctionnement suivants :

- En mode « **STOP** », la CPU n'exécute pas le programme, et on ne peut pas charger un projet.
- En mode « **STARTUP** », la CPU entame une procédure de démarrage.
- En mode « **RUN** », le programme est exécuté de façon cyclique, certaines parties d'un projet peuvent être chargées dans la CPU en mode RUN.



Figure III.7. L'état des modes fonctionnement de S7-1200.

h. Modules de communication

Chaque CPU SIMATIC S7-1200 peut être complétée par trois modules de communication maximum. Les modules de communication **RS 422/485** et **RS 232** conviennent pour des liaisons série point à point à base de caractères. Les fonctions de bibliothèque **USS Drive Protocol** et **Modbus RTU** Maître et Esclave sont déjà contenues dans le système d'ingénierie SIMATIC STEP 7 Basic.

Dans notre banc d'essai nous avons choisis l'interface **RS 422/485** par ce que le variateur de vitesse qu'on a utilisé (SINAMIC V20) prend en charge la communication avec les API SIEMENS via USS sur RS 422/485.

➤ **Modules de communication RS 422/485**

La figure ci-dessous représente le Module de communication CM 1241, RS 422/485, Sub-D9 pôles, prise en charge de Freeport. Pour faire la communication entre l'API et le variateur de vitesse.



Figure III.8. Module de communication **RS 422/485**.

Le tableau ci-dessus représente les caractéristiques du module de communication **RS 422/485** :

Tension d'alimentation (Valeur nominale (CC))	24 V cc
Courant d'entrée (Consommation, maxi)	220 mA; sur bus interne 5V CC
Puissance dissipée	1.1W
Nombre d'interfaces	1
Couplage point à point	1 000 m

(Longueur de câble, maxi)	
---------------------------	--

Tableau III.2. Tableau d'information générale sur **RS 422/485**.

i. Module d'alimentation

Il est important de choisir une alimentation protégée contre les courts-circuits, les surcharges, les surchauffes ou surtensions. Dans d'autres applications, une alimentation courant continué fixe peut être essentielle. Dans notre système réalisé on a utilisé l'alimentation **PM 1207** de référence **6EP1332-1SH71**. C'est l'alimentation idéale pour les automates S7-1200.

➤ **Le module d'alimentation PM 1207**

L'alimentation monophasée SIMATIC PM 1207 avec sélection automatique de la plage de tension d'entrée correspond parfaitement à l'automate SIMATIC S7-1200. Il fournit l'alimentation aux CPU avec une entrée 24V ainsi qu'aux modules de signalisation et aux charges 24V connectées aux modules [22].



Figure III.9. Module d'alimentation PM 1207.

➤ **Caractéristiques et avantages de PM 1207 6EP1332-1SH71**

- ✓ Connexion à tous les réseaux monophasés (120V CA / 230 V CA) grâce à la commutation automatique de plage sauvegarde à court terme en cas de coupure de courant.
- ✓ Indicateur d'état à LED verte pour 24V OK.
- ✓ Deux bornes de sortie 24 Vcc pour la connexion de consommateurs 24V.
- ✓ Puissance active fournie / typique 60 W.

✓ COURANT 2.5A.

III.4.2. Partie opérative

Appelée parfois partie puissance, Cette partie exécute les ordres reçus de la partie commande, elle transforme les signaux de commande en énergie électrique, pneumatique ou hydraulique pour réaliser le fonctionnement du système. En même temps, elle transmet l'état du système à la partie commande à travers les capteurs.

Elle comporte en général :

III.4.2.1. Pré-actionneur

L'énergie issue de la chaîne d'information est faible, insuffisante pour être utilisable directement par les actionneurs. Le rôle du pré-actionneur est de **distribuer**, en le modulant si besoin, et sur ordre de la partie commande, **l'énergie utile et importante** aux actionneurs. Si l'actionneur qui suit dans la chaîne fonctionnelle est électrique, le pré-actionneur sera aussi électrique (relais, contacteur, variateur, hacheur, carte de puissance). Dans notre projet nous avons utilisé un variateur de vitesse V20 de marque **SIEMENS** pour varier (commander) la vitesse du moteur asynchrone.

III.4.2.2. Actionneurs

Les actionneurs convertissent l'énergie d'entrée disponible sous une certaine forme (électrique, Pneumatique, hydraulique) en une énergie utilisable sous une autre forme, par exemple le moteur électrique transforme une énergie électrique à une énergie mécanique.

Il y'a plusieurs types d'actionneurs, par exemple les :

- **Actionneurs pneumatiques** (vérin et moteur pneumatique).
- **Actionneurs hydraulique** (vérin hydraulique).
- **Actionneurs électriques** (moteur asynchrone).

III.4.2.3. Capteurs

Le capteur est un organe qui prélève des informations sur le comportement de la partie opérative et les transforme en une information exploitable par la partie commande.

Pour résumer. Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable.

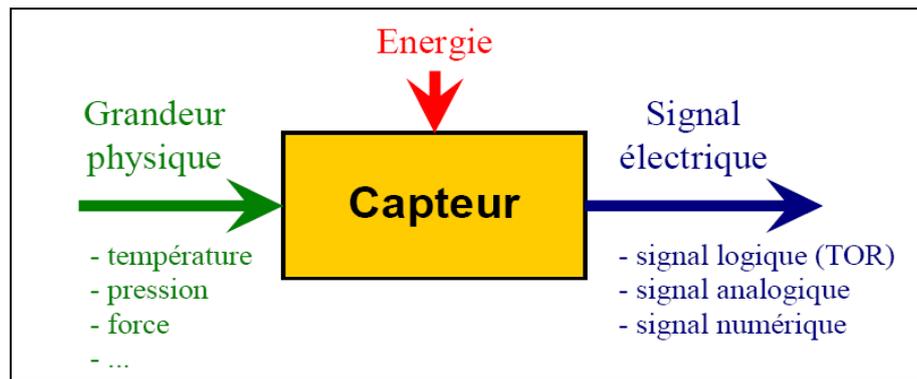


Figure III.10. Principe d'un capteur.

Dans l'industrie, les capteurs sont classés en trois types selon le signal délivré.

- **Les capteurs T.O.R.** (capteur de fin de course et de proximité).
- **Les capteurs numériques** (encodeur incrémentale).
- **Les capteurs analogiques** (capteur de température).

III.4.3. Partie de supervision

C'est la partie qui effectue **la supervision** du système. Aussi appelé l'interface homme machine. Elle permet d'effectuer des réglages d'afficher des messages et de gérer les défauts.

III.4.3.1. Définition de la supervision

La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés. La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des automates programmables.

En informatique industrielle, la supervision des procédés peut être une application de surveillance, de contrôle-commande ou de diagnostic ou l'ensemble de ces dernières réunies.

Elle se fait à travers un logiciel fonctionnant sur un ordinateur en communication, via un réseau local ou distant industriel, avec un ou plusieurs équipements.

III.4.3.2. Avantages de la supervision

Un système de supervision donne de l'aide à l'opérateur dans la conduite du processus, son but est de présenter à l'opérateur des résultats expliqués et interprétés et son avantage principal est :

- Surveiller le processus à distance.
- La détection des défauts.
- Le diagnostic et le traitement des alarmes.

- Traitement des données.

III.4.3.3. Interface Homme-Machine

La programmation et la mise en marche d'une installation industrielle automatisée ne sont pas suffisantes, il est donc nécessaire de visualiser l'état et le mode de fonctionnement de l'installation.

Il existe plusieurs configurations d'interface de contrôle / commande. La configuration la plus simple est de rassembler toutes les informations sur une **Interface Homme-Machine**, pour faciliter la tâche de l'opérateur.

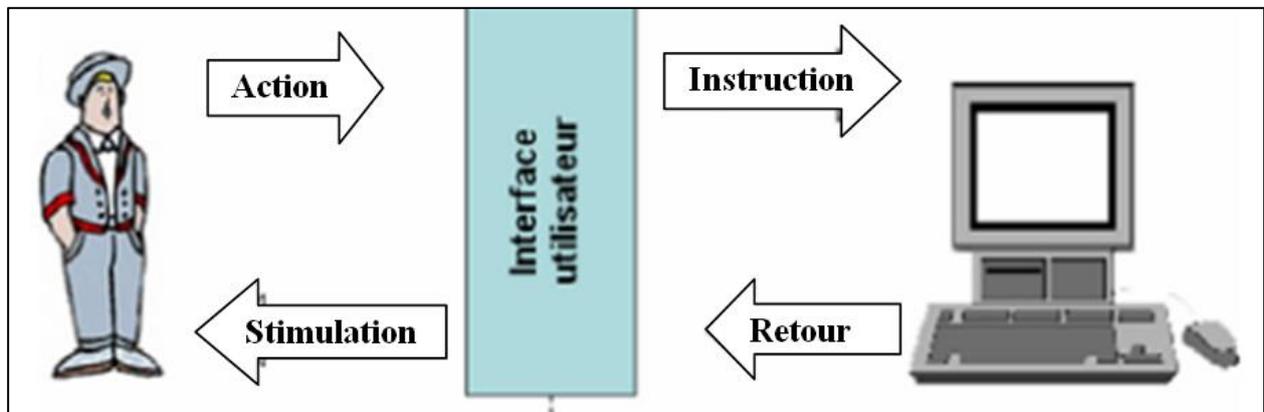


Figure III.11. Interface Homme-Machine.

a. IHM SAMKOON SK-102HS

La série SK IHM de SAMKOON est un produit d'interface Homme-Machine de haute qualité, il intégrant entièrement une unité centrale, une unité d'entrée et de sortie, un écran, une mémoire et d'autres unités de module, il peut être largement appliqué dans les systèmes de contrôle industriels de tous les secteurs de l'industrie.

Avec une conception optimale via matériel et logiciel, il est conforme aux exigences de contrôle de la machine pour la précision et la précision du toucher, ainsi que pour les couleurs d'écran.

Sur la base de la version précédente, l'IHM de la série SK apporte de grandes améliorations avec un temps de démarrage plus rapide, vitesse de communication supérieure et une sensibilité plus élevée qu'auparavant. Il fournit une interface série standard pouvant être connectée à d'autres périphériques. Principalement compatible avec la communication série, il peut communiquer avec les principaux API du marché, tels que SAMKOON, MITSUBISHI, SIEMENS, OMRON, MODICON [23].



Figure III.12. Face avant de SK-102HS.

b. Différents ports de connexions et d'alimentation de SK-102HS

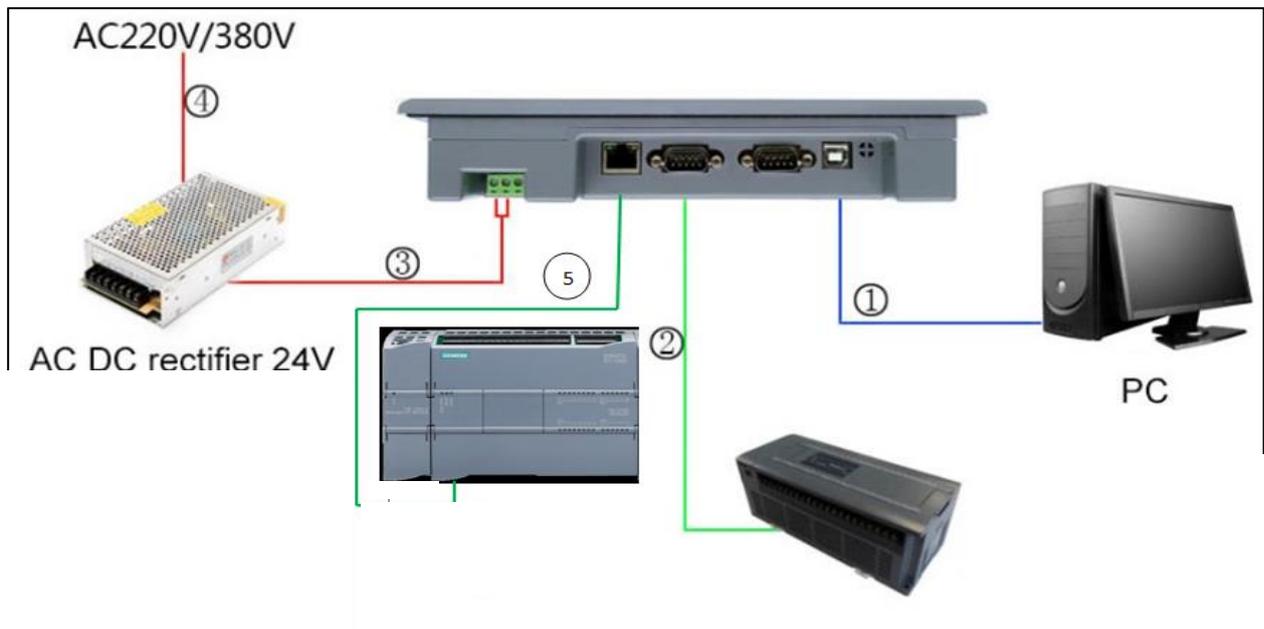


Figure III.13. Schéma de connexion de l'écran tactile de la série SK.

1. Fil de programmation (USB).
2. Fil de programmation (COM).
3. Fil d'alimentation (24V DC).
4. Fil d'alimentation (220V -380V AC).
5. Fil de programmation (Ethernet RG45).

c. Caractéristiques techniques de SK-102HS

Le tableau suivant représente les caractéristiques techniques de SK-102HS :

Tension d'alimentation	24V DC (+/-15%)
Consommation d'énergie	6W
Portes séries	COM1 et COM2 pour liaison série RS 232/422/485
Port USB-A	Support pour une vitesse de transmission de 10M/S
Port USB-B	Support pour flash disc, imprimante, souris et autres périphériques USB
Porte Ethernet	Réaliser la communication entre les IHM ou IHM et le dispositif de commande via Ethernet (S7-1200).
Mémoire	128M FLASH + 128M DDR2
Processeur	processeur hautes performances de qualité industrielle : cortex A8, jusqu'à 600 MHz
Logiciel de programmation	SKTOOL 6.2

Tableau III.3. Tableau de caractéristique de SK 102HS.

III.5. Logiciel de programmation

Pour la programmation de notre banc d'essai, on a utilisé le logiciel de Siemens TIA Portal V13.

III.5.1. Présentation du logiciel TIA Portal V13

La plate-forme TIA (Totally Integrated Automation) Portal est la dernière évolution des logiciels de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré, dans un seul logiciel cette plate-forme regroupe la programmation des différents dispositifs d'une installation. On peut donc programmer et configurer, en plus de l'automate, les dispositifs HMI les variateurs...etc.

Le logiciel STEP 7 Professional (TIA Portal V13) est l'outil de programmation des nouveaux automates comme :

- ✓ SIMATIC S7-1500
- ✓ SIMATIC S7-1200
- ✓ SIMATIC S7-300
- ✓ SIMATIC S7-400

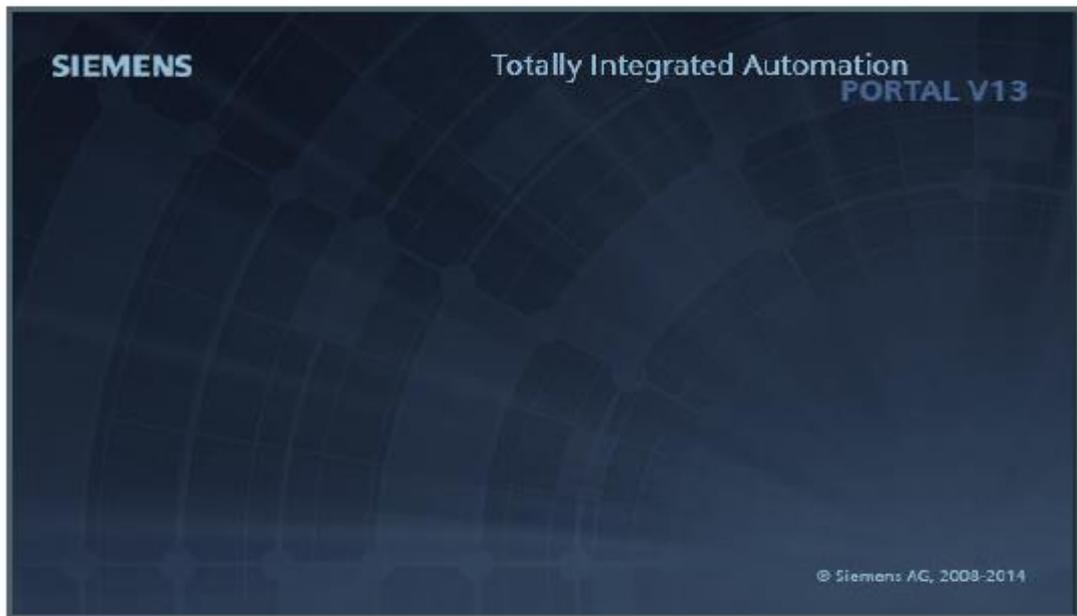


Figure III.14. Logo de TIA Portal V13.

III.5.2. Création d'un nouveau projet

- Cliquez sur nouveau projet
- Complétez les données nécessaires
 - Nom du projet : projet 1
 - Chemin : C:\Users\pc click\Desktop\projet tia
- Cliquez sur le bouton Créer

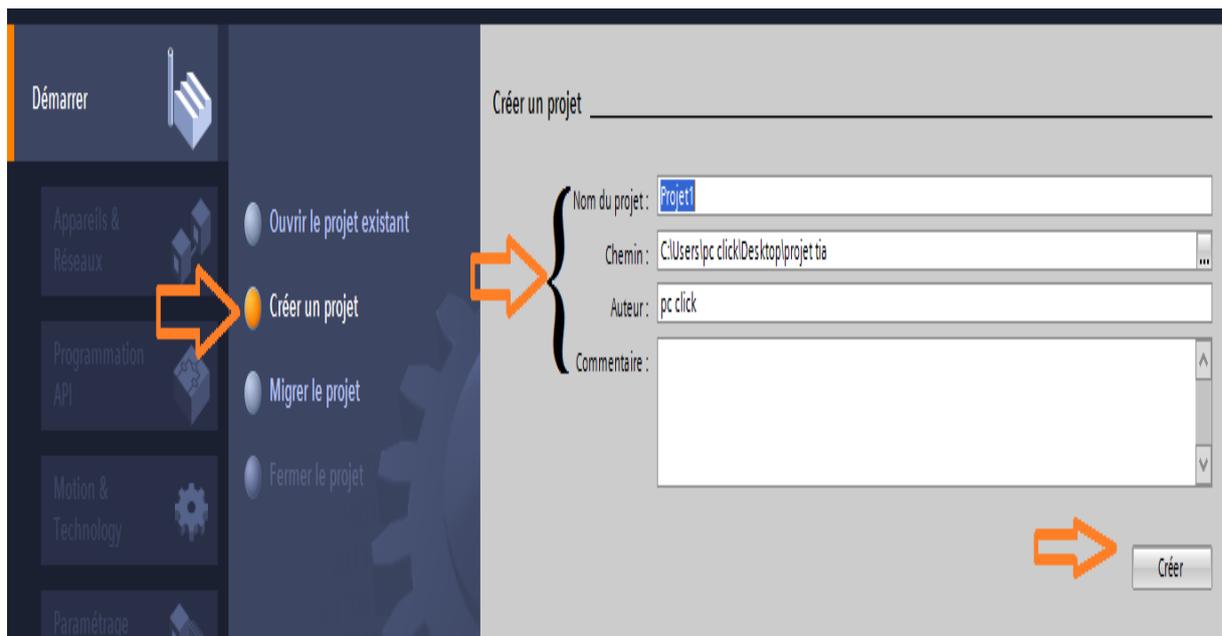


Figure III.15. Vue TIA Portal (Création d'un projet).

III.5.3. Configurer et ajouter un appareil

Après la création de notre projet on doit configurer l'appareil qu'on doit programmer.

- clique sur configurer un appareil

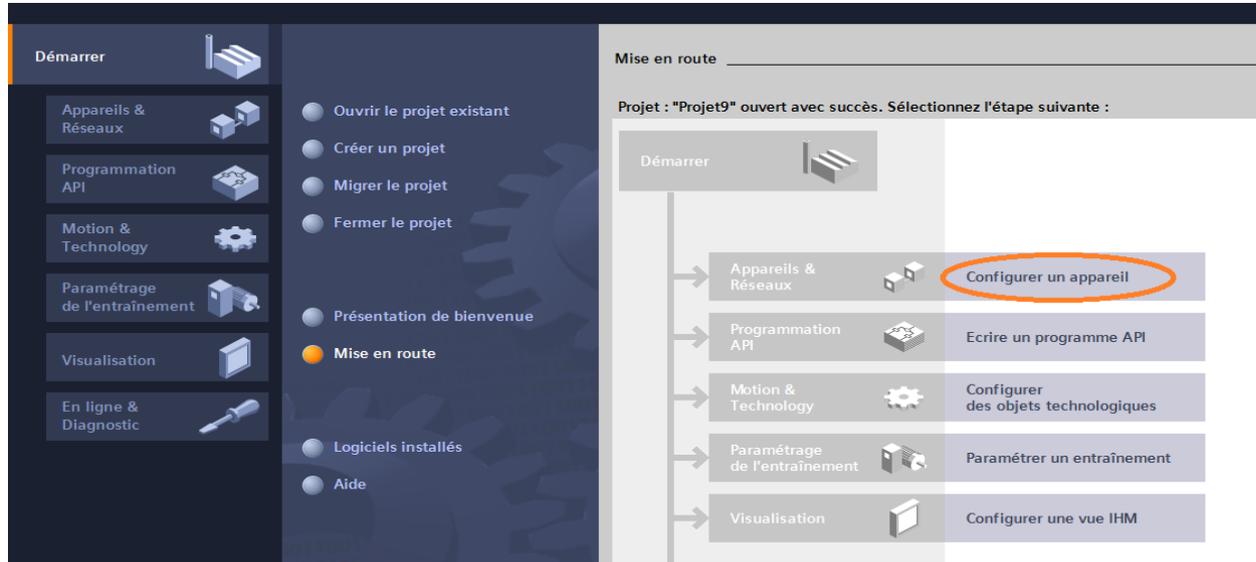


Figure III.16. Configurer un appareil sur TIA Portal.

- clique sur ajouter un appareil.
- clique sur SIMATIC S7-1200 (on va travailler par S7-1200).
- clique sur CPU.
- clique sur l'automate choisi (dans notre projet on a utilisé le CPU 1212c DC/DC/Rly).
- clique sur le numéro de série de CPU concerné.
- clique sur ajouter.

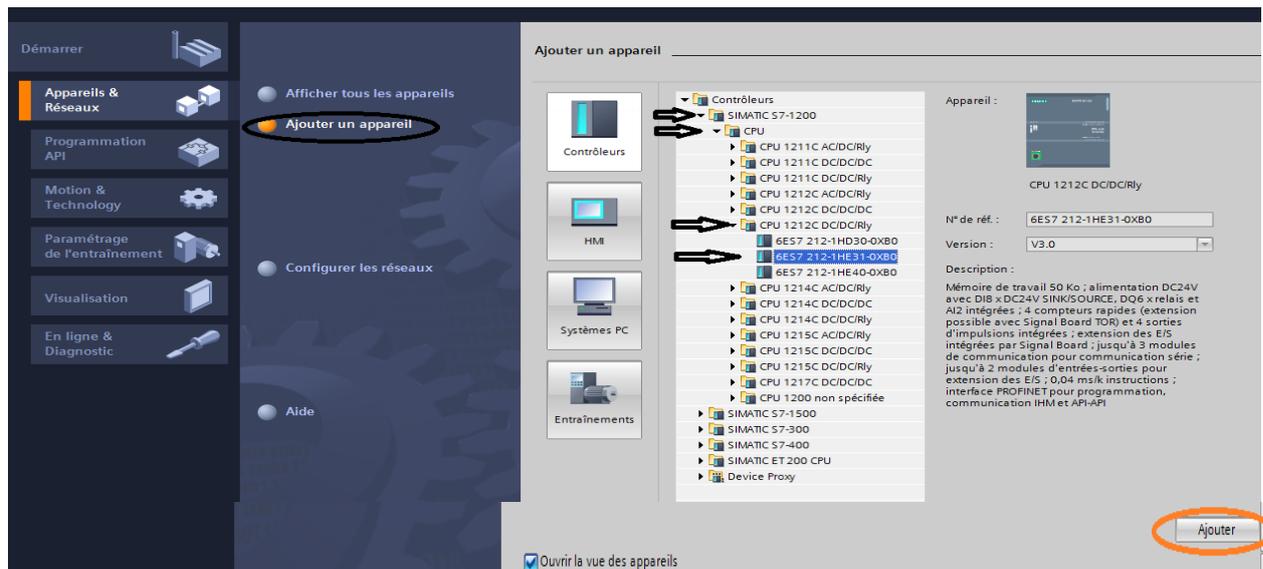


Figure III.17. Ajouter un appareil sur TIA Portal.

III.5.4. L'interface et l'adresse Ethernet de CPU

Après la configuration et l'ajoute d'appareil on a l'interface suivant :

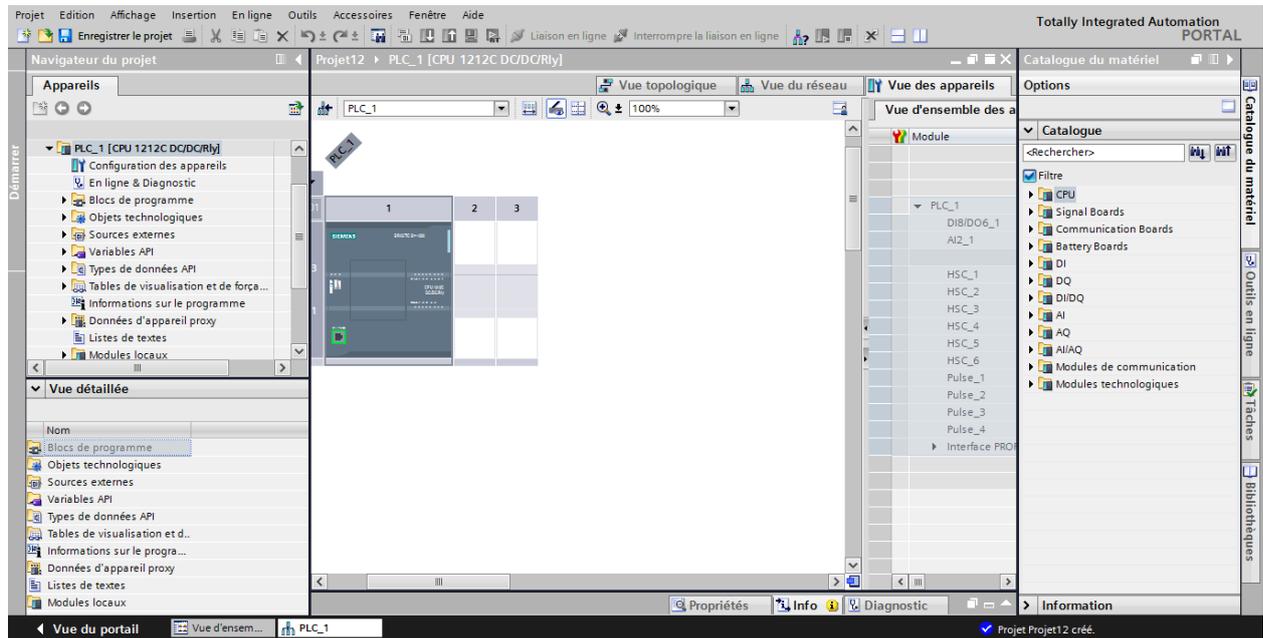


Figure III.18. Vue de l'interface de TIA Porta.

➤ l'adresse Ethernet de CPU

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur le connecteur Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés. Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau. On utilisera comme adresse pour l'automate 192.168.0.1 de l'automate.

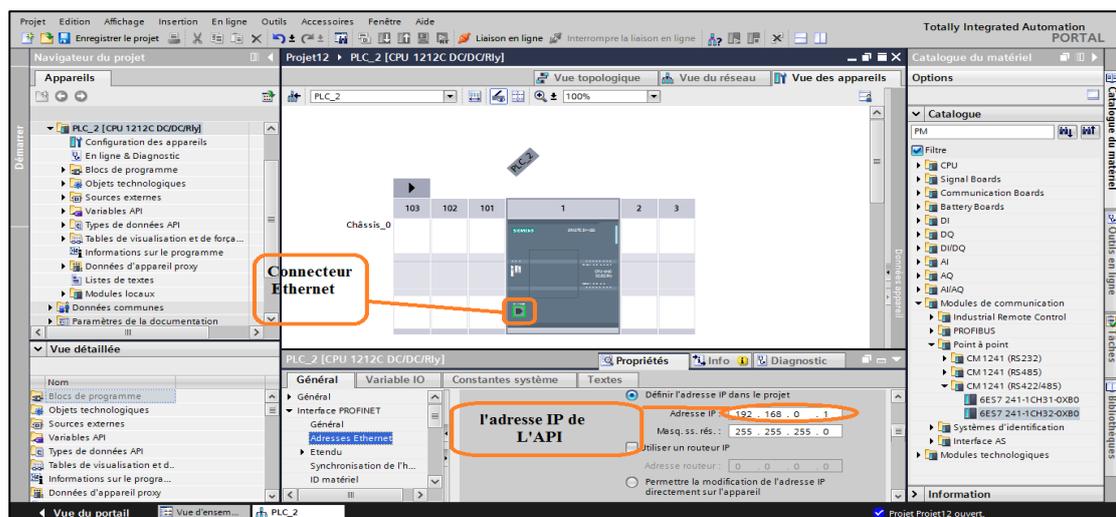


Figure III.19. Adresse Ethernet de la CPU.

III.5.5. Ajoute module de communication

Pour ajouter un module de communication dans le TIA Portal il faut suivre les étapes suivant :

- dans la partie droite de l'interface (catalogue du matériel) cliquer sur module de communication.
- cliquer sur le dossier point à point pour choisir.
- cliquer sur le dossier de module de communication concerné (RS 422/485).
- cliquer sur numéro de série correspond a notre module communication.

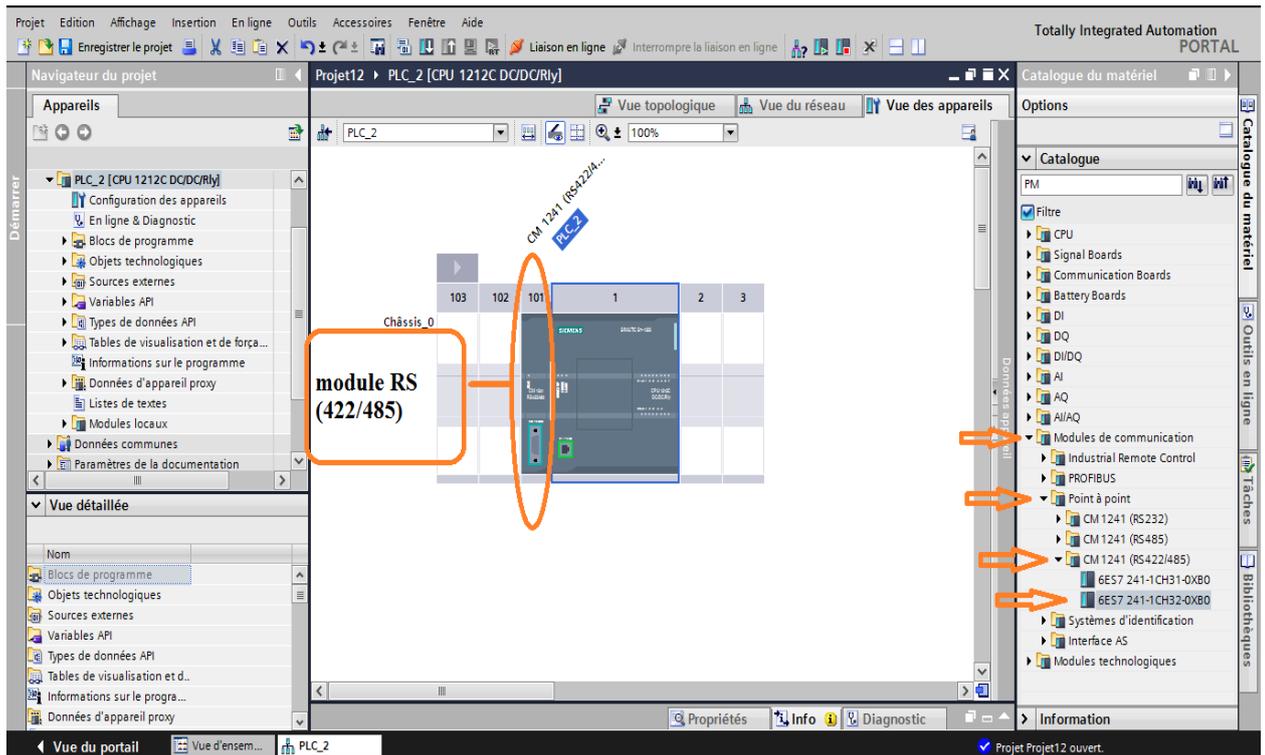


Figure III.20. L'ajoute de module communication.

III.5.6. Activation de compteur rapide (HSC)

Les compteurs rapides sont généralement utilisés comme entraînements pour des compteurs dont l'axe, tournant à vitesse constante, est équipé d'un générateur d'impulsions. Le générateur d'impulsions assure un nombre donné de valeurs de comptage par tour et une impulsion de réinitialisation une fois par tour. Les générateurs d'horloge et l'impulsion de réinitialisation du générateur d'impulsions fournissent les entrées du compteur rapide.

Dans notre banc d'essai on a travaillé par l'automate S-7 1200 type 1212C qui contient 4 HSC de HSC1 HSC4, on a besoin que deux donc on active deux HSC (HSC1 et HSC2), pour l'activer il faut suivre les étapes suivant :

- double cliquer sur le PLC pour entrer dans les propriétés de PLC
- choisir compteurs rapides (HSC)

Le tableau suivant représente l'activation, fonctionnement et l'adressage E/S de HSC.

	HSC1	HSC2
<p>Activation HSC</p>	<p>Compteurs rapides (HSC) _____</p> <p>HSC1 _____</p> <p>> Général _____</p> <p>Activer</p> <p><input checked="" type="checkbox"/> Activer ce compteur rapide</p>	<p>HSC2 _____</p> <p>> Général _____</p> <p>Activer</p> <p><input checked="" type="checkbox"/> Activer ce compteur rapide</p>
<p>Fonction HSC</p>	<p>> Fonction _____</p> <p>Mode de comptage: Fréquence</p> <p>Phase de fonctionnement: Compteur A/B quadruple</p> <p>Source d'entrée: Entrée de CPU intégrée</p> <p>Sens de comptage donné par: Entrée (commande de sens externe)</p> <p>Sens de comptage initial: Incrémental</p> <p>Période de mesure de fréquence: 0.01 sec</p>	<p>> Fonction _____</p> <p>Mode de comptage: Comptage</p> <p>Phase de fonctionnement: Compteur A/B quadruple</p> <p>Source d'entrée: Entrée de CPU intégrée</p> <p>Sens de comptage donné par: Entrée (commande de sens externe)</p> <p>Sens de comptage initial: Incrémental</p> <p>Période de mesure de fréquence: +- sec</p>
<p>Adresse E/S De HSC</p>	<p>> Adresses E/S _____</p> <p>Adresses d'entrée</p> <p>Adresse de début: 1000</p> <p>Adresse de fin: 1003</p> <p>Mémoire image: M1 cyclique</p>	<p>> Adresses E/S _____</p> <p>Adresses d'entrée</p> <p>Adresse de début: 1004</p> <p>Adresse de fin: 1007</p> <p>Mémoire image: M1 cyclique</p>

Tableau III.4. Activation et fonctionnement et adresse E/S d'un HSC.

III.5.7. Langages de programmation

Le TIA PORTAL met à disposition de puissants éditeurs pour la programmation des automates SIMATIC S7.

- ✓ Texte structuré (SCL).
- ✓ Schéma à contacts (CONT).
- ✓ Logigramme (LOG).

❖ **CONT et LOG – Langages de programmation graphiques**

➤ LOG

Le logigramme (LOG) est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

➤ CONT (LADDER)

Le langage LADDER ou schéma à contacts est un langage de programmation graphique très populaire auprès des automaticiens. Il ressemble un peu aux schémas électriques, et est facilement compréhensible. Il existe 3 types d'éléments de langage :

- ❖ Les entrées (ou contact), qui permettent de lire la valeur d'une variable booléenne.
- ❖ Les sorties (ou bobines) qui permettent d'écrire la valeur d'une variable booléenne.
- ❖ Les blocs fonctionnels qui permettent de réaliser des fonctions avancées.

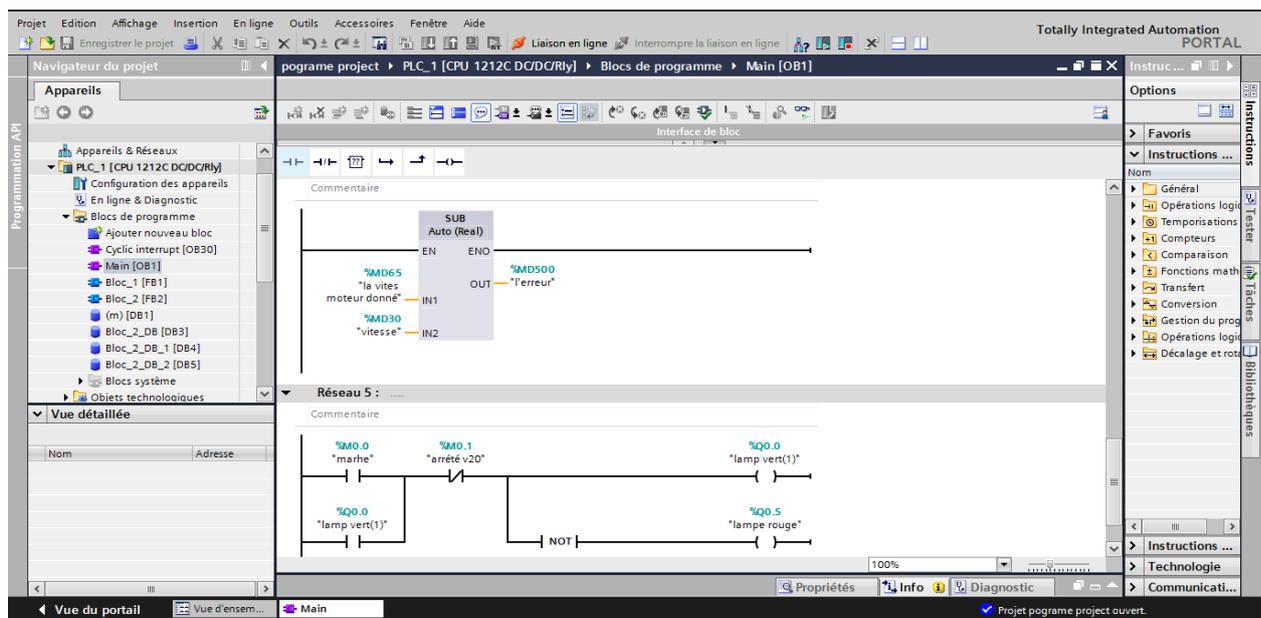


Figure III.21. Présentation d'un schéma à contacte (CONT).

❖ SCL (Structured Control Language)

Le SCL (Structured Control Language) correspond au langage textuel de haut niveau ST (Structured Text) défini dans la norme CEI 61131-3. SCL convient notamment à la programmation rapide d'algorithmes complexes et de fonctions mathématiques ou à des missions relevant du domaine du traitement des données.

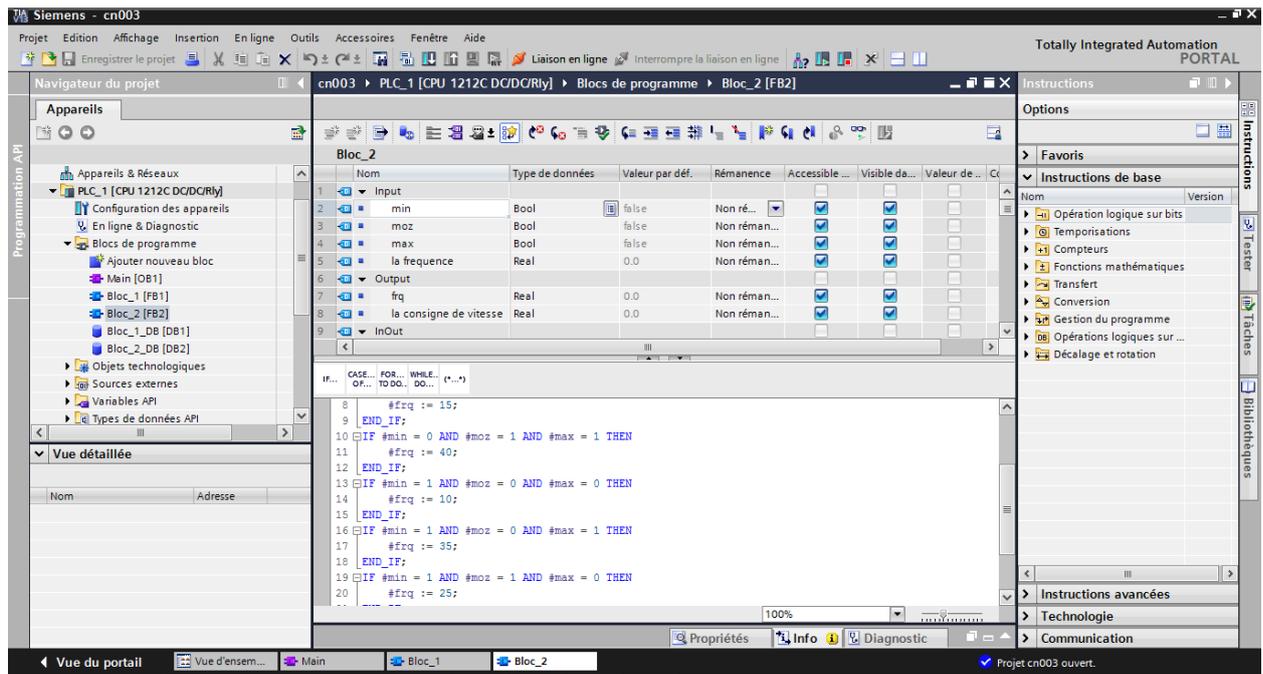


Figure III.22. Présentation d'un langage structuré (SCL).

III.5.8. Blocs de programme

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

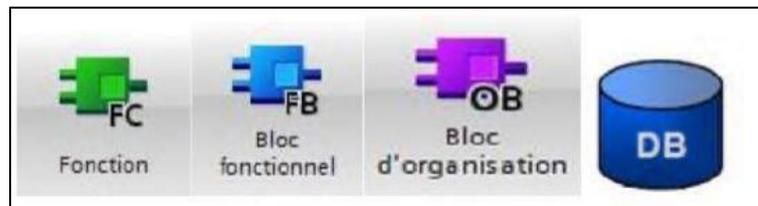


Figure III.23. Les différents blocs de programmation.

- ❖ Les blocs d'organisation **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).

- ❖ Les fonctions **FC** : sont des blocs de code sans mémoire.

Les données des variables temporaires sont perdues après l'exécution de la fonction. Si on veut mémoriser ces données, il faut utiliser des opérands globaux. Elles sont utilisées pour la programmation de fonctions utilisées plusieurs fois. On simplifie de ce fait la programmation.

- ❖ Les blocs fonctionnels **FB** : 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.
- ❖ Les blocs de donnée **DB** : sont des zones données du programme utilisateur qui contiennent des données utilisateur.

III.5.9. Compilation et chargement de programme et la configuration matérielle

Une fois la configuration matérielle et la programmation réalisée, il faut la compiler et la charger dans l'automate. La compilation se fait à l'aide de l'icône « **compiler** » de la barre de tâche. On sélectionne l'API dans le projet puis on clique sur l'icône « **compiler** ». En utilisant cette manière, on effectue une compilation matérielle et logicielle.

Pour charger la configuration et le programme dans l'automate, on effectue un clic sur l'icône « **charger dans l'appareil** ».

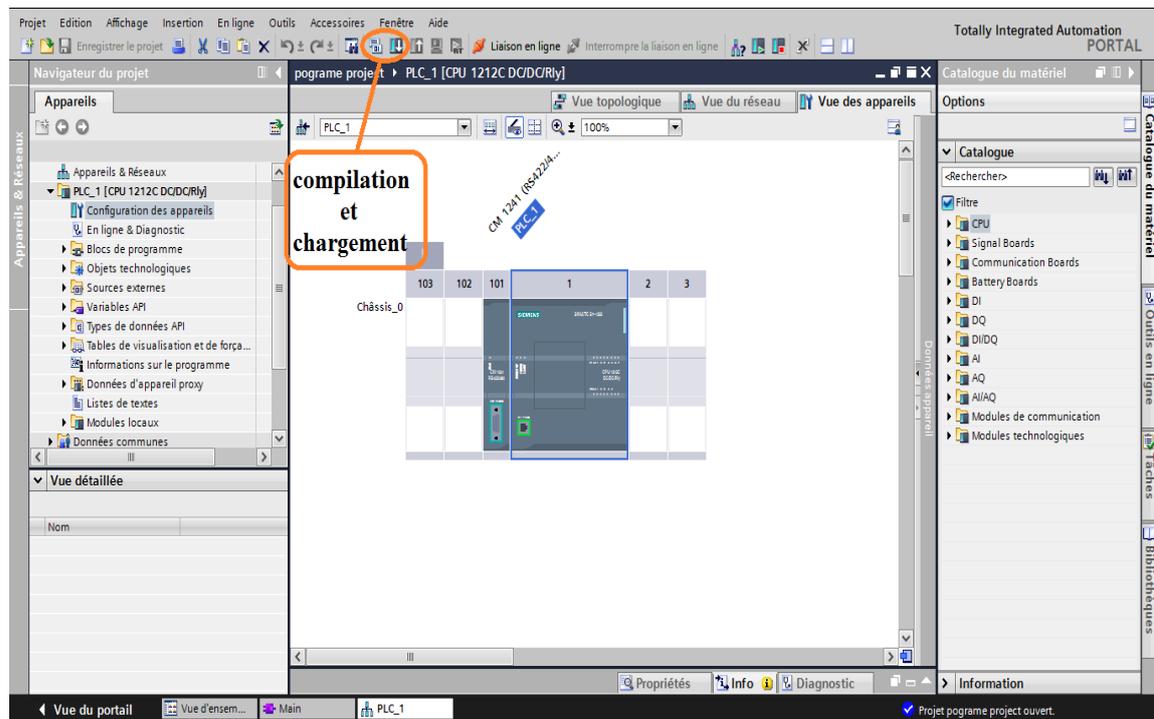


Figure III.24. La compilation et le chargement de configuration et de programme.

La fenêtre ci-dessous s'ouvre et on doit faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si on choisit le mode PN/IE, l'API doit posséder une adresse IP. Pour faire la liaison entre l'automate et le PC et on charge le programme contenu dans le PC à l'automate.

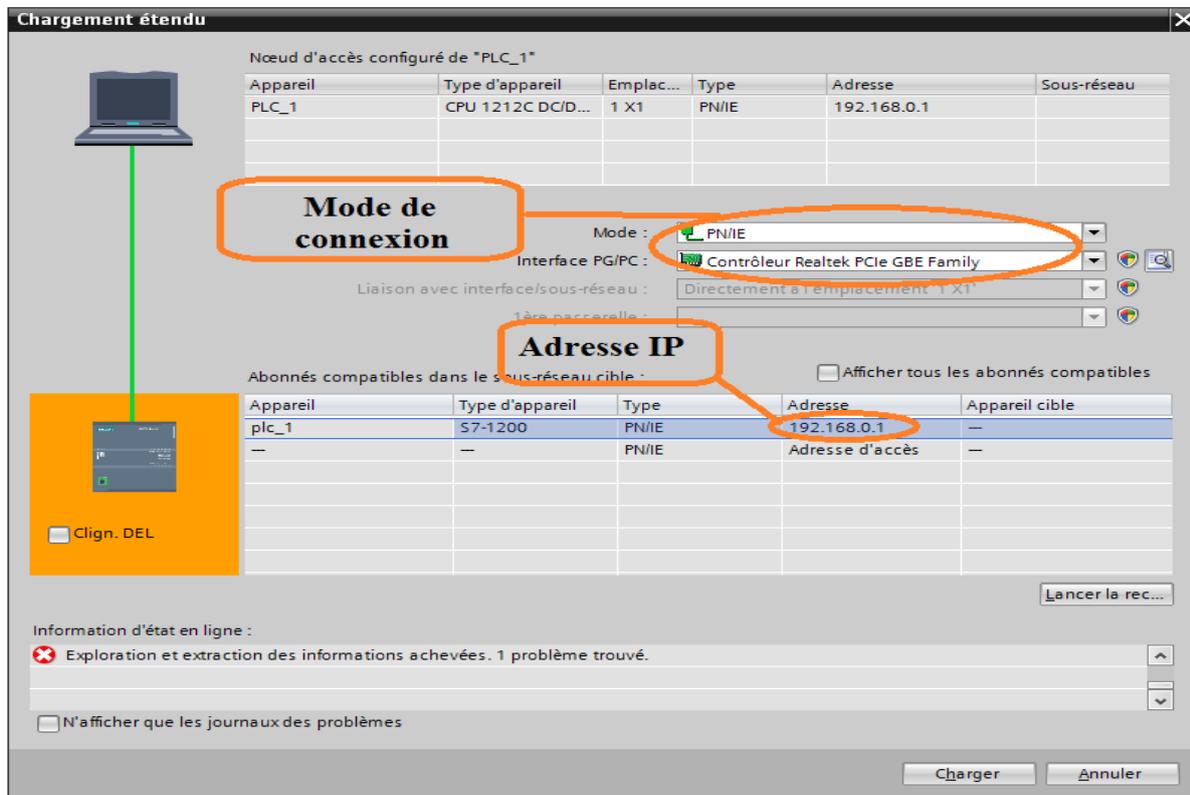


Figure III.25. Mode de connexion.

III.6. Conclusion

Dans ce chapitre nous avons étudié les systèmes automatisés, la partie commande, la partie opérative et la partie supervision de ces derniers, on parler sur les automates programmable et ces structure interne et externe et pour notre banc d'essai on a choisis CPU S7 1200, nous avons utilisé le module de communication RS 422/485 pour faire la communication entre le variateur de vitesse et l'API.

La partie IHM nous permettra de contrôler et de commander notre unité à distance et nous présenter l'IHM SAMKOON SK-102HS et la possibilité de communiquer avec les principaux API de marché.

Nous avons présenté la procédure à suivre pour la création de notre programme sous TIA PORTAL V13, un aperçu des blocs contact a été donné ainsi que les compteurs rapides utilisés lors de la programmation.

CHAPITRE IV

Réalisation du banc d'essai de la variation de vitesse d'un moteur asynchrone

IV.1. Introduction

Notre but est de réaliser un banc d'essai de la variation de vitesse d'un moteur asynchrone commandé par un automate programmable et supervisé par un écran tactile. Nous allons commander le variateur de vitesse «SIEMENS SINAMICS V20» avec l'automate programmable «SIEMENS S7-1200» via USS sur RS-485. Après en ayant établi la communication entre le variateur de vitesse et l'API nous allons envoyer la consigne via un écran tactile. Puis nous allons voir si le moteur suit vraiment la consigne et comparer entre la vitesse réelle du moteur avec celle lu depuis le variateur sachant qu'on obtiendra la vitesse réelle du moteur avec le codeur incrémental que nous allons raccorder à l'arbre du moteur. Toutes ces actions seront contrôler et supervisé avec un écran tactile.

IV.2. Schéma structurelle du banc d'essai

Notre système automatisé, est un banc d'essai de la chaine d'action « variateur de vitesse et moteur asynchrone », ce dernier est commandé par un API et est supervisé par un écran tactile.

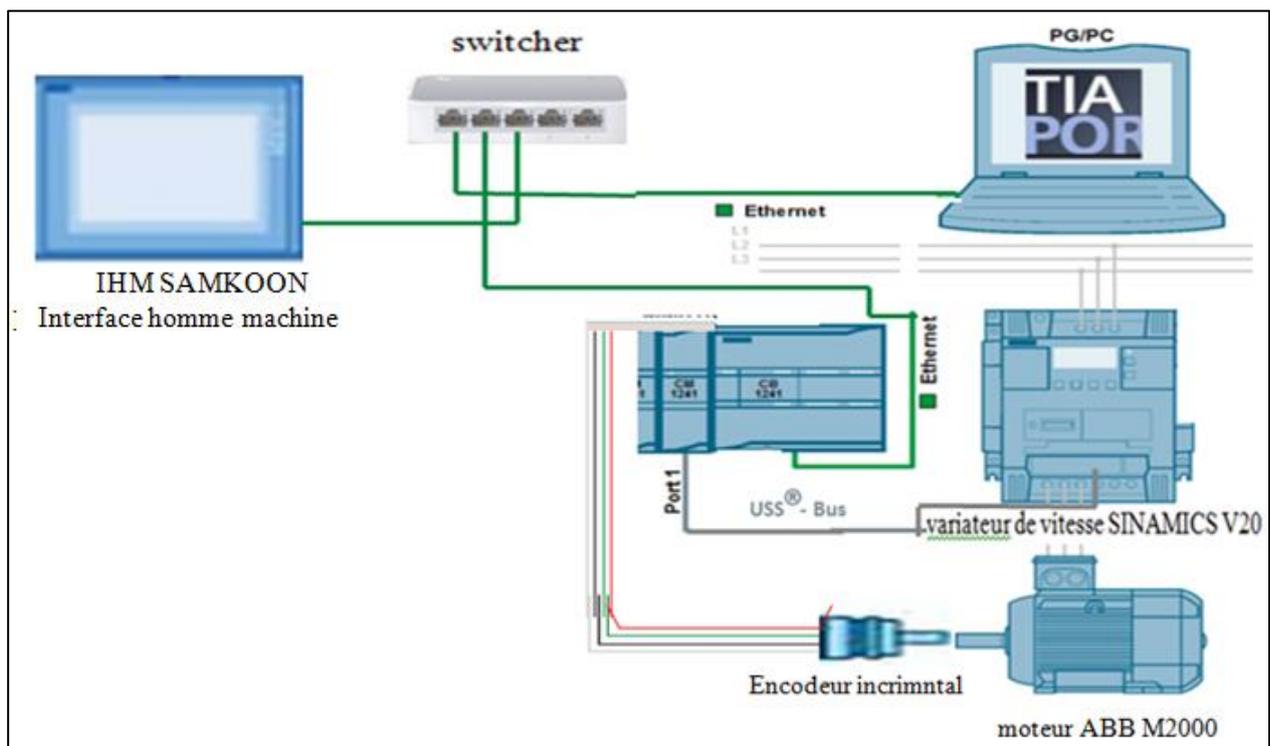


Figure IV.1. Schéma structurelle du banc d'essai

- L'IHM est lié à l'automate, cette liaison permet à l'écran de recevoir des informations depuis l'automate et donner des ordres à ce dernier via le bus de communication Ethernet.

- L'automate est relié au variateur de vitesse via le protocole USS utilisé l'interface sur RS-485, L'automate joue le rôle du maître, et le variateur de vitesse est l'esclave, ce qui veut dire que l'automate commande le variateur de vitesse.
- Le variateur est raccordé à un moteur asynchrone, ce dernier reçoit des tensions et des fréquences variables depuis le variateur pour qu'il change de vitesse et de sens de rotation, cette variation est due aux consignes de l'automate. Donc c'est l'automate qui ordonne et contrôle (avec le contrôleur PID) la variation de la vitesse et du sens de rotation du moteur.
- L'arbre du moteur est raccordé à un capteur de vitesse, ce capteur de vitesse est lié sur l'automate programmable, pour donner la vitesse réelle du moteur.

IV.3. Raccordements des différents matériels du banc d'essai

IV.3.1. Raccordement du variateur de vitesse avec l'API

Le variateur SINAMICS V20 prend en charge la communication avec les API Siemens via USS sur RS422/485. Il est possible de choisir si l'interface RS-485 doit appliquer le protocole USS ou MODBUS RTU. USS est le réglage de bus par défaut. Un câble à paire torsadée blindé est recommandé pour la communication RS422/485.

Veiller à raccorder correctement entre les bornes du bus (P+, N-) de l'appareil à l'une des extrémités du bus et un réseau de terminaison entre les bornes du bus de l'appareil à l'autre extrémité du bus[14].

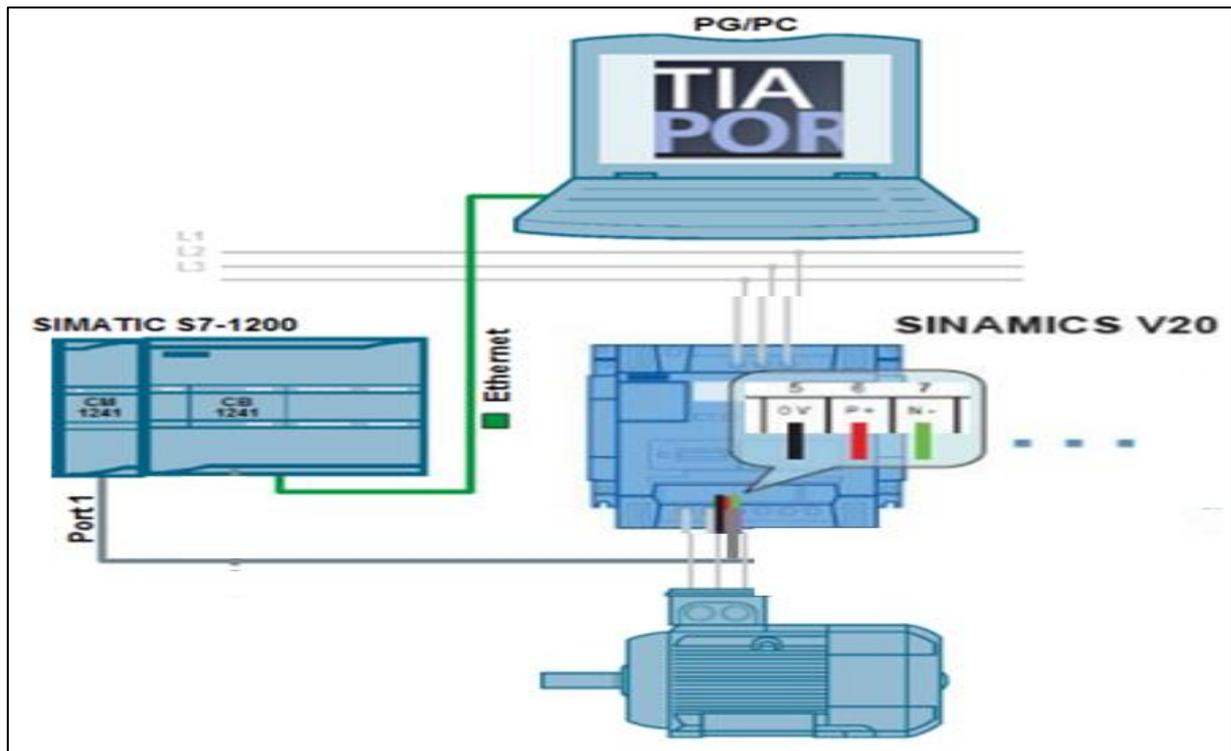


Figure IV.2. Communication du variateur SINAMICS V20 avec l'API S7-1200 par l'interface RS-485

➤ **Communication USS**

Un API (maître) peut connecter jusqu'à 31 variateurs (esclaves) sur la liaison série et les commander avec le protocole de bus série USS. Un esclave ne peut jamais transmettre sans être d'abord initié par le maître de telle sorte qu'aucun transfert d'informations direct entre des esclaves n'est possible [14].

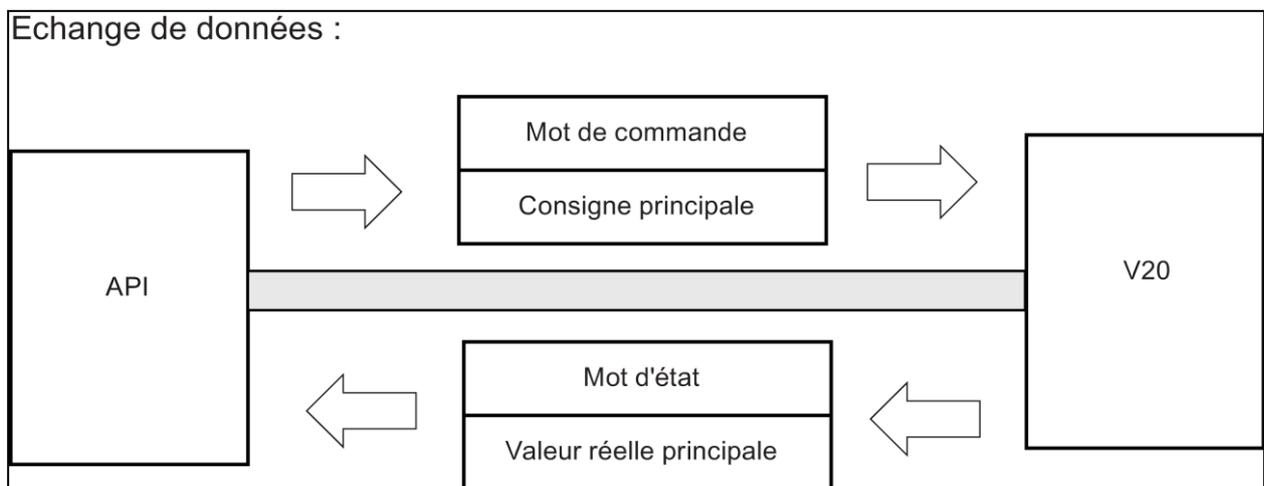


Figure IV.3.Présentation de communication USS

IV.3.2. Raccordement du variateur de vitesse avec le moteur asynchrone

Le variateur de vitesse est raccordé au moteur en utilisant un couplage triangle, vu que c'est le variateur qui alimente le moteur.



Figure IV.4. Raccordement du variateur de vitesse avec le moteur

IV.3.3. Raccordement l'arbre du moteur asynchrone avec l'encodeur incrémentale

Pour raccorder notre encodeur incrémentale avec le moteur, nous avons utilisé **accouplements flexibles** pour codeurs.

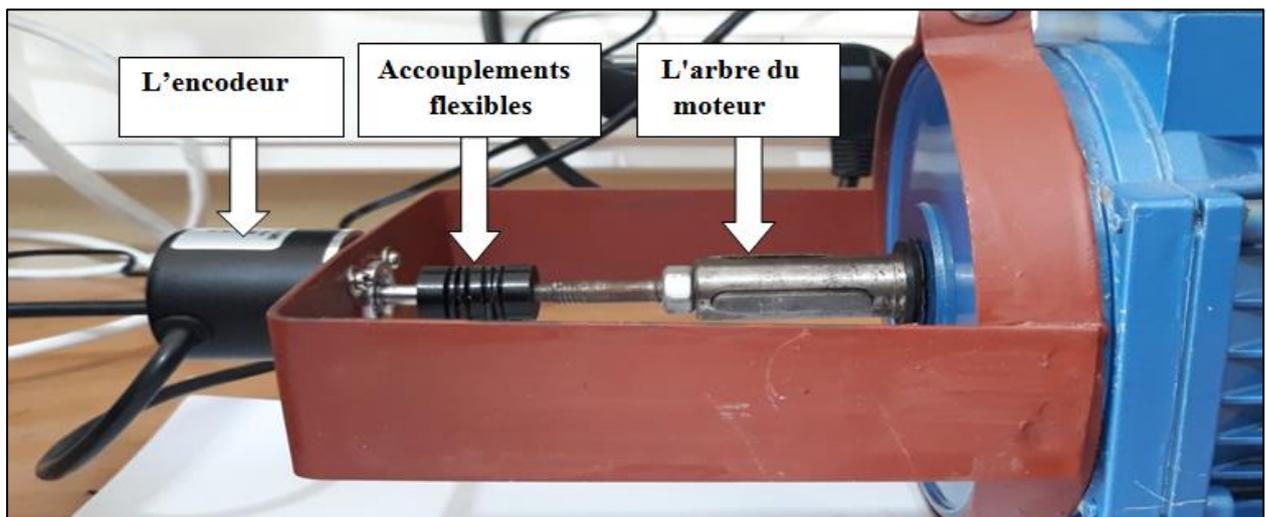


Figure IV.5. Raccordement l'arbre du moteur avec l'encodeur incrémental

IV.3.4. Raccordement de l'encodeur incrémentale avec l'API

Nous avons utilisé encodeur incrémentale pour calculer la vitesse de moteur. L'encodeur est lié sur l'automate pour compter le nombre d'impulsions. Notre encodeur contient quatre fils, Output Sortie collecteur ouvert, Il est nécessaire de prévoir une résistance de rappel ($R = 1k\Omega$) du côté du circuit utilisateur, compatible avec la plage de tension d'alimentation 5V à 24V, Les connecteurs ne sont pas inclus [24]. Le rouge pour entrée d'alimentation (5V à 24V) et le noire GND le branchement de ces deux fils dans les entré de l'api (L +, M) qui fournit 24 VDC, le fil Vert (Sortie de signal de phase A) et le Blanc (Sortie de signal de phase B) on les a raccordé aux entrées de l'API.

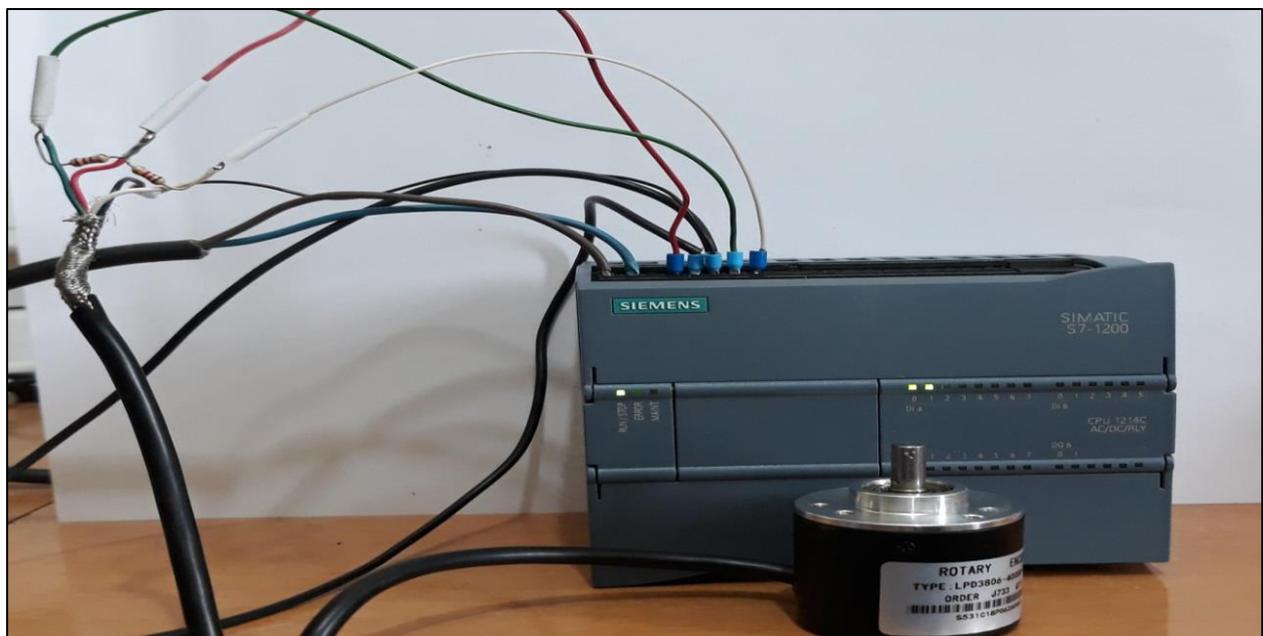


Figure IV.6.Raccordement des sorties du l'encodeur avec l'API

IV.3.5. La communication entre l'API et l'IHM

Pour établir une communication entre l'automate et l'IHM, nous avons utilisé la porte Ethernet interface de l'IHM qui réaliser la communication entre l'IHM et le dispositif de commande via le câble Ethernet RG45.

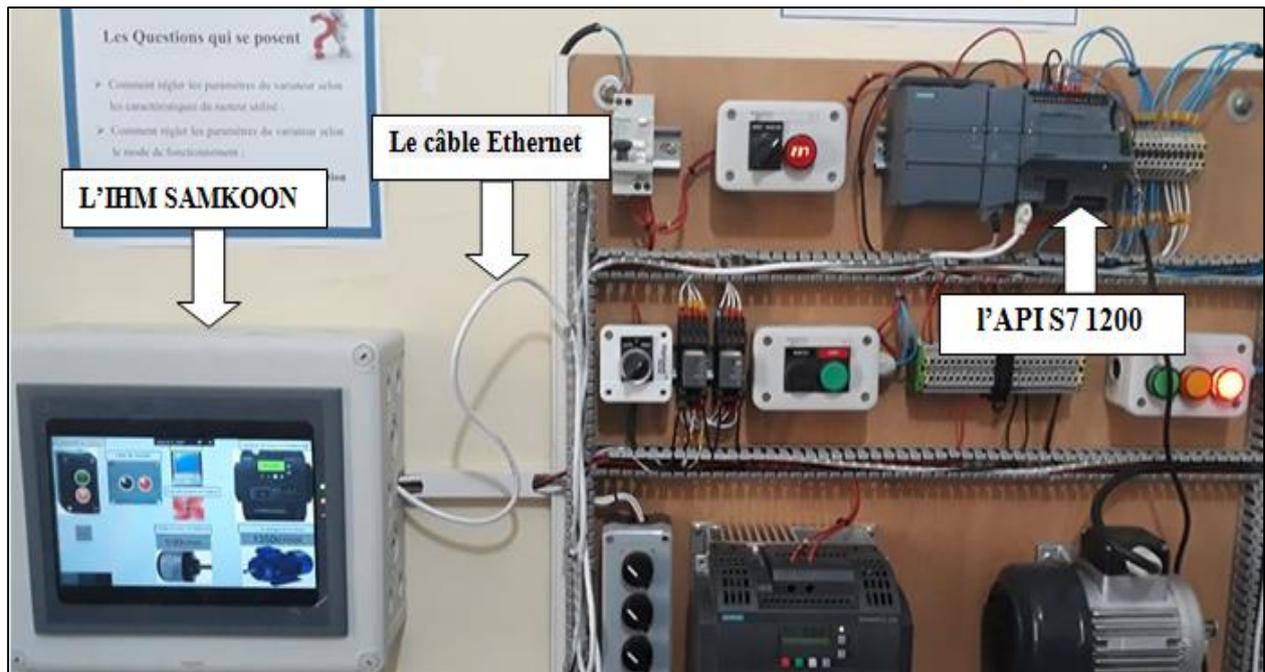


Figure IV.7. Connexion entre l'API et l'IHM

IV.3.6. Connexion entre le PC et l'IHM

Pour établir la connexion entre le PC et l'IHM, nous avons utilisé la porte USB A dans l'IHM ce qui permet de télécharger les fichiers de configuration du PC vers l'IHM. Dans le paragraphe suivant, on va présenter la méthode de programmation du logiciel de l'IHM « **SKTOOL** ».

IV.3.6.1. Logiciel « **SKTOOL** »

Créé pour la série IHM de Samkoon SK à partir de Shenzhen Samkoon Technologie, SKTOOL est un logiciel d'édition de configuration de Samkoon IHM, SKTOOL prend en charge les pilotes de communication PLC des principaux fabricants mondiaux, tels que Samkoon, Siemens, Mitsubishi, Omron, Fujitsu, Panasonic, Schneider, Emerso ...etc. Le logiciel de configuration SKTOOL est un système de développement avec un environnement de développement intégré et de nombreuses fonctions avancées, il est facile à apprendre et à appliquer, ainsi les ingénieurs peuvent facilement développer leurs propres projets pour la configuration utilisée dans SKTOOL. L'interface de développement et les domaines fonctionnels sont illustrés dans la figure suivante.

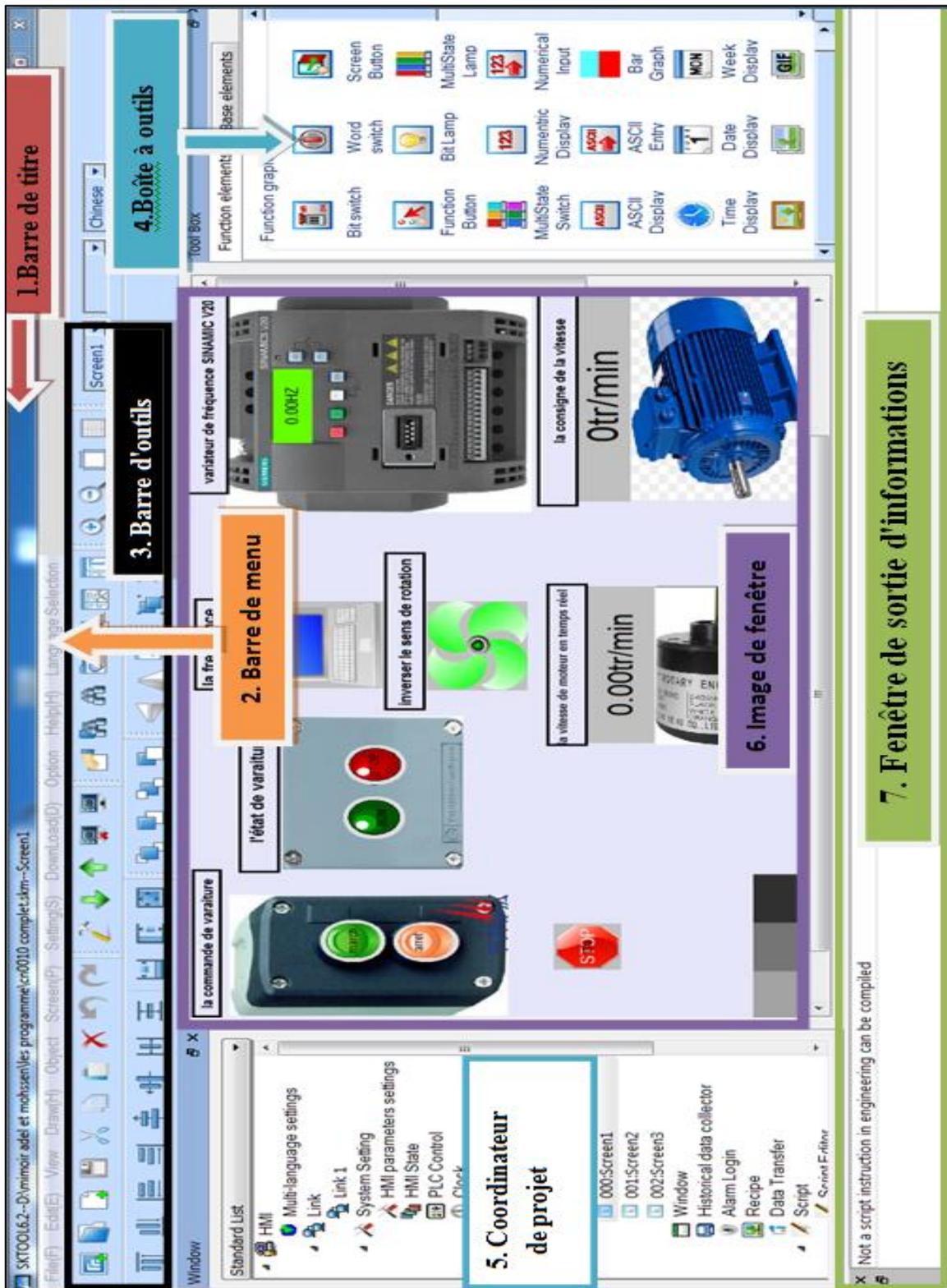


Figure III.8. Vu de l'interface de logiciel SKTOOL

- **Barre de titre** : Affiche le nom de la version de SKTOOL.
- **Barre de menus** : affiche les divers menus de commande de SKTOOL qui sont tous des menus déroulants.

- **Barre d'outils standard** : raccourcis vers le bouton de commande; Afficher le fichier; projet ouvert; nouveau projet; restauration et autres boutons de fonction.
- **Boîte à outils** : Contrôles de configuration de base conçus pour que les ingénieurs puissent les configurer; contient des contrôles de fonction et des contrôles de base.
- **Image de fenêtre** : fenêtre à l'intention du personnel d'ingénierie pour la configuration et la modification d'un objet
- **Coordinateur de projet** : connexions de communication, paramétrage, gestion des images, paramétrage des alarmes, formule définie et menu arborescent défini par le collecteur de données d'historique.
- **Fenêtre de sortie d'informations** : Invites relatives à la saisie, à la sortie, aux messages d'erreur et autres conseils fournis aux ingénieurs lors de l'édition et de la compilation de la configuration.

IV.4. Réglage les paramètres du variateur de vitesse (SINAMICS V20)

- **Etape 1** : Réglage de la configuration des paramètres du moteur que nous avons utilisé. Les caractéristiques nominales du moteur issues de la plaque signalétique. Pour les paramètres du moteur est réglé une seule fois dans notre banc d'essai comme on a expliqué dans le deuxième chapitre.
- **Etape 2** : Choix du macro de connexion selon les besoins, nous avons choisi deux macro cn003 (Vitesses fixes) et cn010 (Régulation USS). Lors de la mise en service du variateur, le réglage des macros de connexion est un réglage réalisé une fois pour toutes. Veiller à procéder comme suit avant de modifier le réglage des macros de connexion.
- **Etape 3** : Réglage des macros d'application, chaque macro d'application fournit un jeu de paramètres pour une application spécifique. Après avoir sélectionné une macro d'application, les réglages correspondants sont appliqués au variateur pour simplifier le processus de mise en service. nous avons choisi le macro d'application AP010 (Applications de pompe simples).
- **Etape 4** : Réglage des paramètres communs optimisé les performances du variateur.

IV.4.1. Les modes de fonctionnement de variateur de vitesse V20

Dans les modes de fonctionnements de variateur de vitesse V20 il y a onze macros, dans notre projet nous avons choisi uniquement deux macros cn003 et cn010 parmi les macros existent. Dans la suite on va présenter les deux macros et la configuration des leurs paramètres.

IV.4.2. Premier mode de fonctionnement Cn003 – Vitesses fixes

Cette macro connexion est représentée trois vitesses fixes avec MARCHE/ARRET. Chaque vitesse représente une fréquence fixe réglée dans les paramètres macro de connexion Cn003. Si plusieurs fréquences fixes sont sélectionnées simultanément, les fréquences sélectionnées sont additionnées.

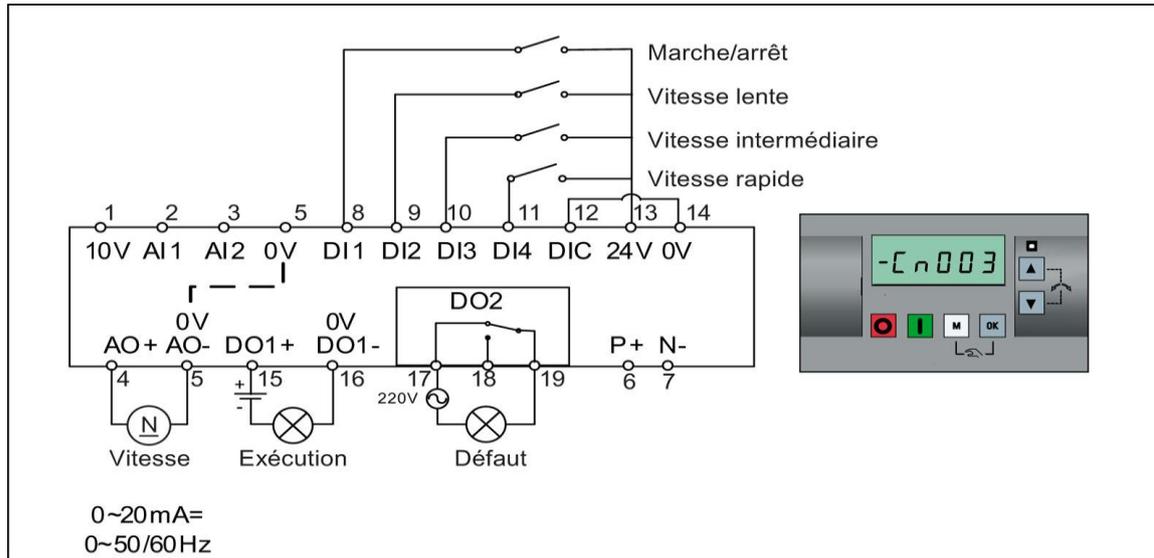


Figure IV.9. Macro connexion Cn003

IV.4.2.1. Réglages les paramètres de macro de connexion Cn003

Le tableau suivant représente les paramètres à régler dans le variateur de vitesse pour le macro connexion cn003.

Paramètre	Description	Valeur par défaut pour Cn003	Valeur par défaut pour Cn003
p1000	Sélection de la fréquence	3	Fréquence fixe
p1001	Fréquence fixe 1	10	Vitesse lente
p1002	Fréquence fixe 2	15	Vitesse intermédiaire
p1003	Fréquence fixe 3	25	Vitesse rapide
p0731	BI : Fonction de la sortie TOR 1	52.2	Variateur en fonctionnement
p0732	BI : Fonction de la sortie TOR 2	52.3	Défaut variateur actif

Tableau IV.1 . paramètres Macro de connexion Cn003

IV.4.2.2. Programmation de l'API dans TAI 13 pour le macro de connexion Cn003

Pour le mode fonctionnement Cn003 on a utilisé deux modes, le premier mode est manuel en utilisant des commutateurs pour change la vitesse, le deuxième mode c'est un mode automatique commandé par l'API et l'IHM.

a. Table de variables standard

Dans cette table on a représenté toutes les variables utilisées de noter programme pour le macro cn003.

	Nom	Type de données	Adresse	Réma...	Visibl...	Acces...	Commentaire
1	march	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	arrêt	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	lamp vert	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	le v20 march	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	botton vitesse MIN	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	vitesse MIN	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	botton MOY	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	vitesse MOY	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	botton MAX	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	vitesse MAX	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	lamp rouge	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	HSC 2	DInt	%ID1004	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	vitesse de moteur réel	Real	%MD30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	la frquence de varaitur	Real	%MD90	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	la consigne de vitesse	Real	%MD100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	l'error	Real	%MD200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	cette vr pour traces lerreur HMI	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	botton march	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	botton arrêt	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	<Ajouter>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure IV.10. Table des variables standards de macro cn003

b. Block d'organisation (programme cycle)

1. La figure ci-dessous représente le premier réseau, ce réseau pour mise en marche de variateur et visualisé leur état par des lampes.

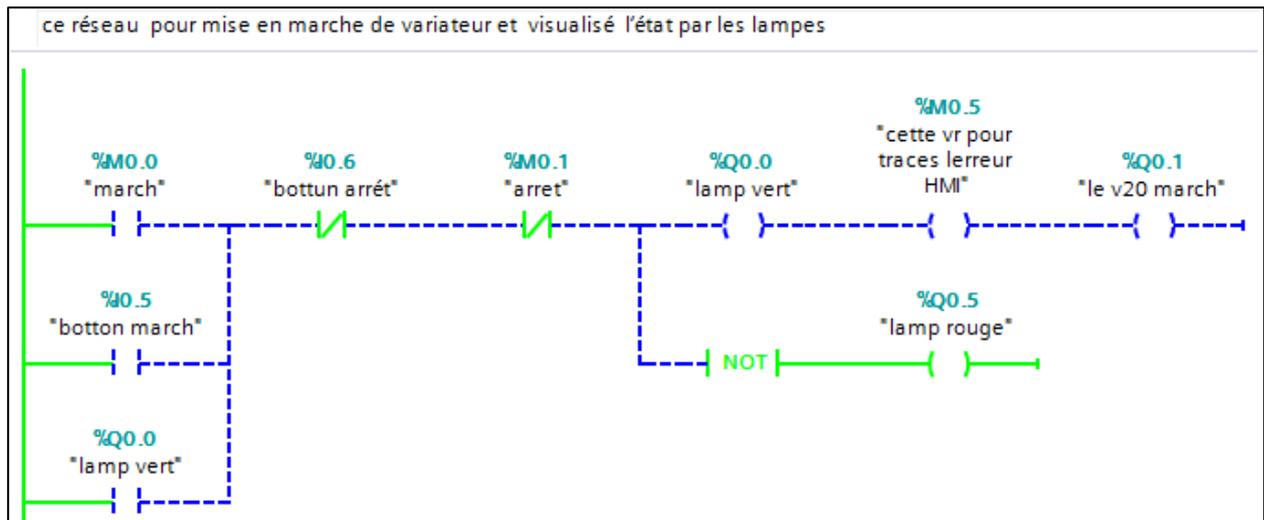


Figure IV.11. Réseau pour mise en marche du variateur et visualisé leur l'état

2. Le deuxième réseau représente la commande des trois vitesses fixes.



Figure IV.12. Réseau pour la commande des trois vitesses

3. Ce réseau représente le bloc fonctionnel "Bloc_2_DB" pour afficher la fréquence commandée et calculer la consigne de vitesse.

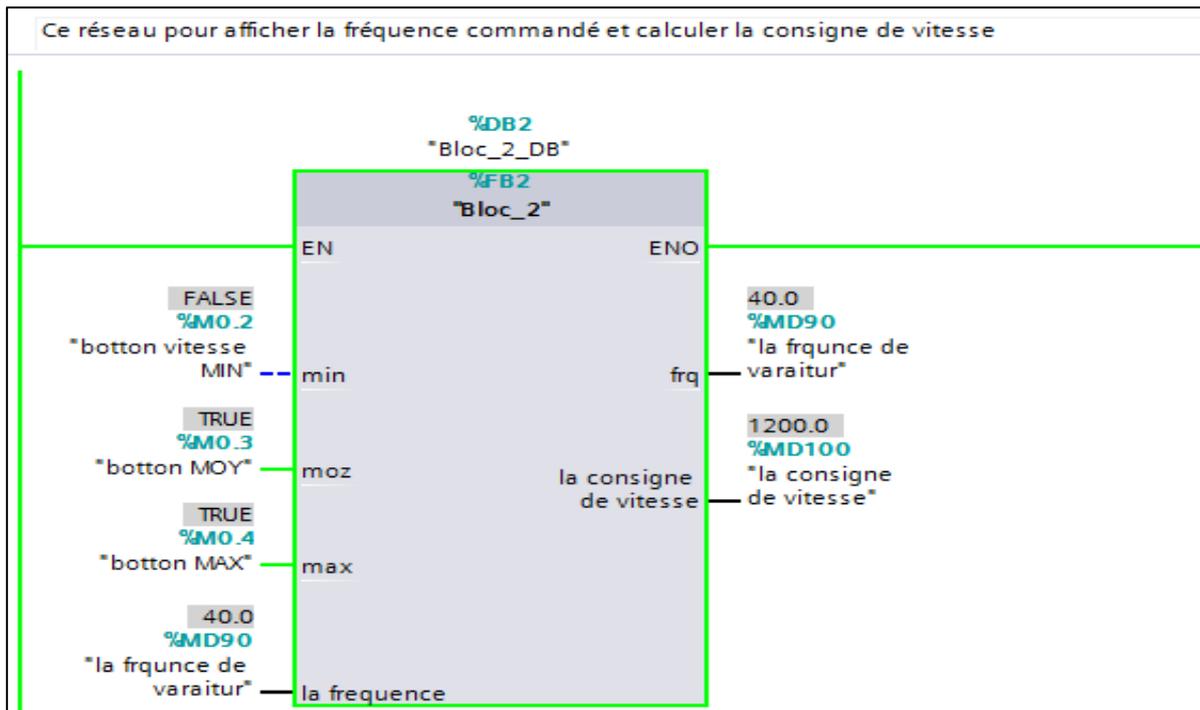


Figure IV.13.Bloc fonctionnel pour afficher la fréquence commandée et calculer la consigne de vitesse.

- La figure ci-dessous représente le bloc fonctionnel "Bloc_1_DB" pour l'encodeur qui sert à calculer la vitesse réelle du moteur.

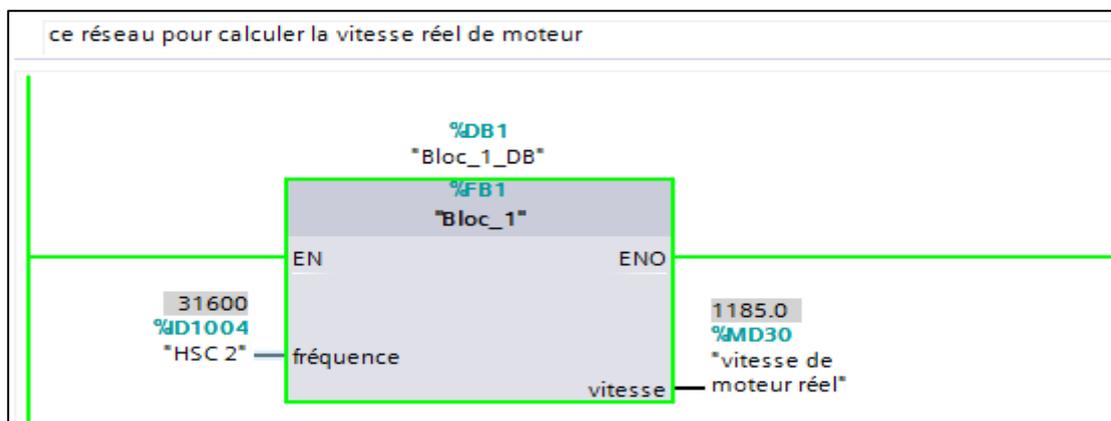


Figure IV.14.Réseau pour calculer la vitesse réelle de moteur

- Dans ce réseau on a calculé l'erreur entre la consigne et la vitesse réelle en utilisant la fonction mathématique soustraction, l'unité de l'erreur est la même que la sortie (tr/min).

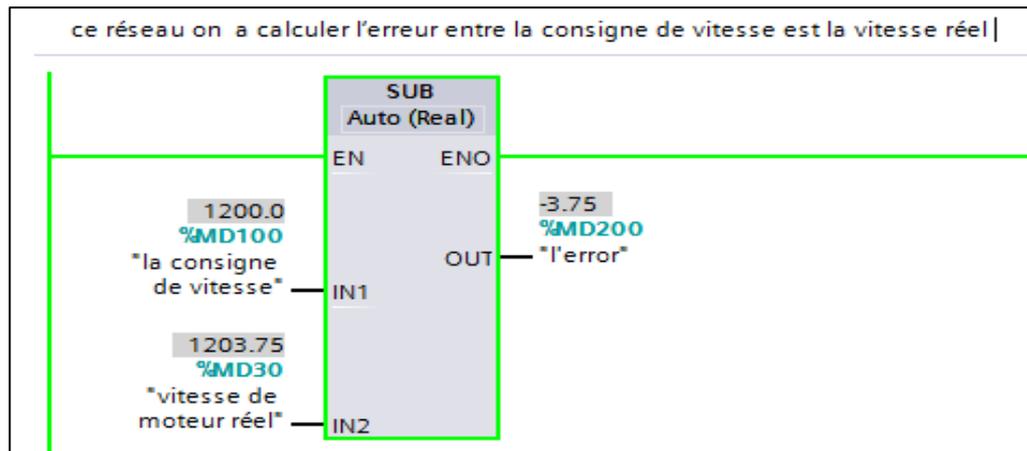


Figure IV.15. Réseau pour calculer l'erreur entre la consigne et la vitesse réelle.

c. Bloc fonctionnel

On a utilisé deux blocs fonctionnels, le premier bloc "Bloc_1_DB" pour calculer la vitesse réelle, et le deuxième bloc "Bloc_2_DB" pour calculer la consigne vitesse du moteur et donner la fréquence.

❖ Le bloc fonctionnel "Bloc_1_DB"

1. La figure ci-dessous représente la table des variables pour bloc "Bloc_1_DB"

	Nom	Type de données	Valeur par déf.	Rémanence	Accessible ...	Visible da...	Valeur
1	Input						
2	fréquence	Dint	0	Non ré...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Output						
4	vitesse	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure IV.16. Tabla des variables pour bloc "Bloc_1_DB"

2. Pour programmer le bloc "Bloc_1_DB", nous avons utilisé l'encodeur incrémental pour calculer la vitesse en (tr/min) d'après les imputions délivré par l'encodeur.

```

1  #vitesse := #fréquence / 1600.0 * 60.0;
2

```

#vitesse	1.211250E+003
#fréquence	32300

Figure IV.17. Le programme pour le block "Bloc_1_DB"

❖ *Le bloc fonctionnel "Bloc_2_DB"*

1. la table des variables du bloc "Bloc_2_DB" est représenté par la figure ci-dessous.

	Nom	Type de données	Valeur par déf.	Rémanence	Accessible ...	Visible da...	Valeur de ..
1	Input						
2	min	Bool	false	Non ré...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	moz	Bool	false	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	max	Bool	false	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	la frequence	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Output						
7	frq	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	la consigne de vitesse	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	InOut						

Figure IV.18. La table des variables du bloc "Bloc_2_DB"

2. Le programme du bloc "Bloc_2_DB"

```

IF... CASE... FOR... WHILE... (*...*)
OF... TO DO... DO...

1 IF #min = 0 AND #moz = 0 AND #max = 0 THEN
2   #frq := 0;
3 END_IF;
4 IF #min = 0 AND #moz = 0 AND #max = 1 THEN
5   #frq := 25;
6 END_IF;
7 IF #min = 0 AND #moz = 1 AND #max = 0 THEN
8   #frq := 15;
9 END_IF;
10 IF #min = 0 AND #moz = 1 AND #max = 1 THEN
11   #frq := 40;
12 END_IF;
13 IF #min = 1 AND #moz = 0 AND #max = 0 THEN
14   #frq := 10;
15 END_IF;
16 IF #min = 1 AND #moz = 0 AND #max = 1 THEN
17   #frq := 35;
18 END_IF;
19 IF #min = 1 AND #moz = 1 AND #max = 0 THEN
20   #frq := 25;
21 END_IF;
22 IF #min = 1 AND #moz = 1 AND #max = 1 THEN
23   #frq := 50;
24 END_IF;
25 # "la consigne de vitesse" := (# "la frequence" * 1500.0) / 50.0;
    
```

Figure IV.19. Le programme du bloc "Bloc_2_DB".

IV.4.2.3. Programme IHM dans SKTOOL pour le macro de connexion Cn003

SKTOOL est un logiciel d'édition de configuration IHM, SKTOOL prend en charge les pilotes de communication PLC des principaux fabricants mondiaux. Pour programmer IHM nous avons utilisé plusieurs éléments pour commander et visualiser notre banc d'essai de la chaîne d'action « variateur de vitesse – moteur asynchrone », pour de mode fonctionnement Cn003 en utilisent cinq fenêtres.

- ❖ La première fenêtre est pour représenter le schéma fonctionnel de macro Cn003.

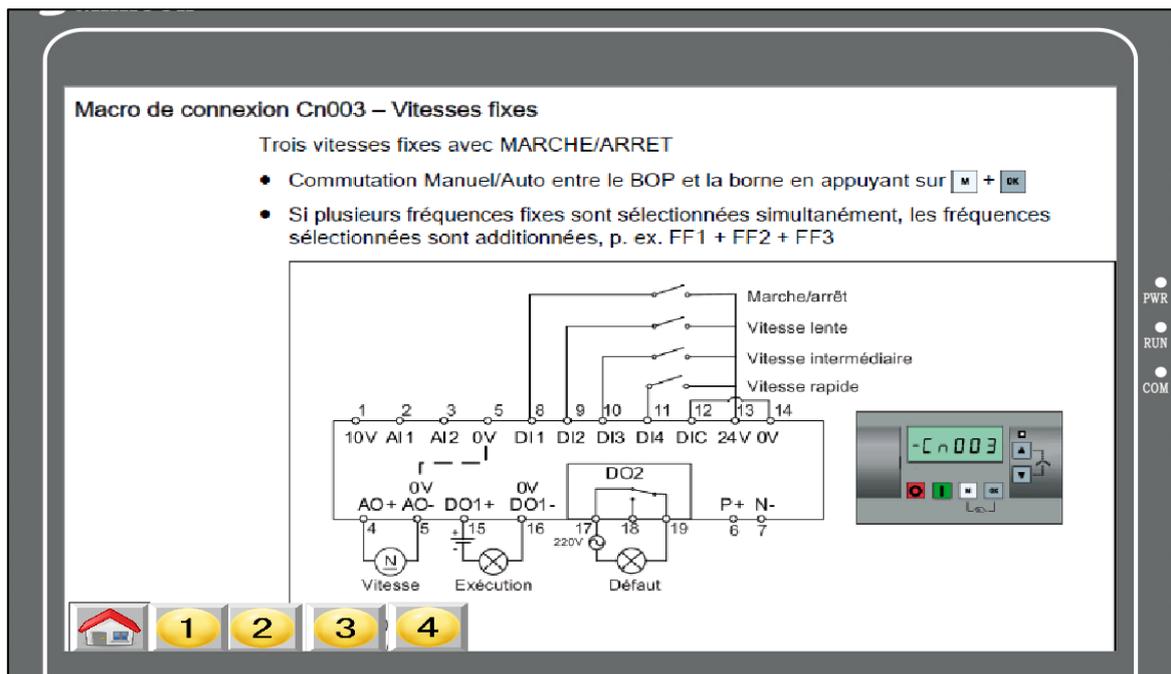


Figure IV.20. La première fenêtre de l'IHM.

- ❖ La deuxième fenêtre pour commander le variateur de vitesse et visualiser leur état, les deux vitesses la consigne et la vitesse réelle (encodeur).

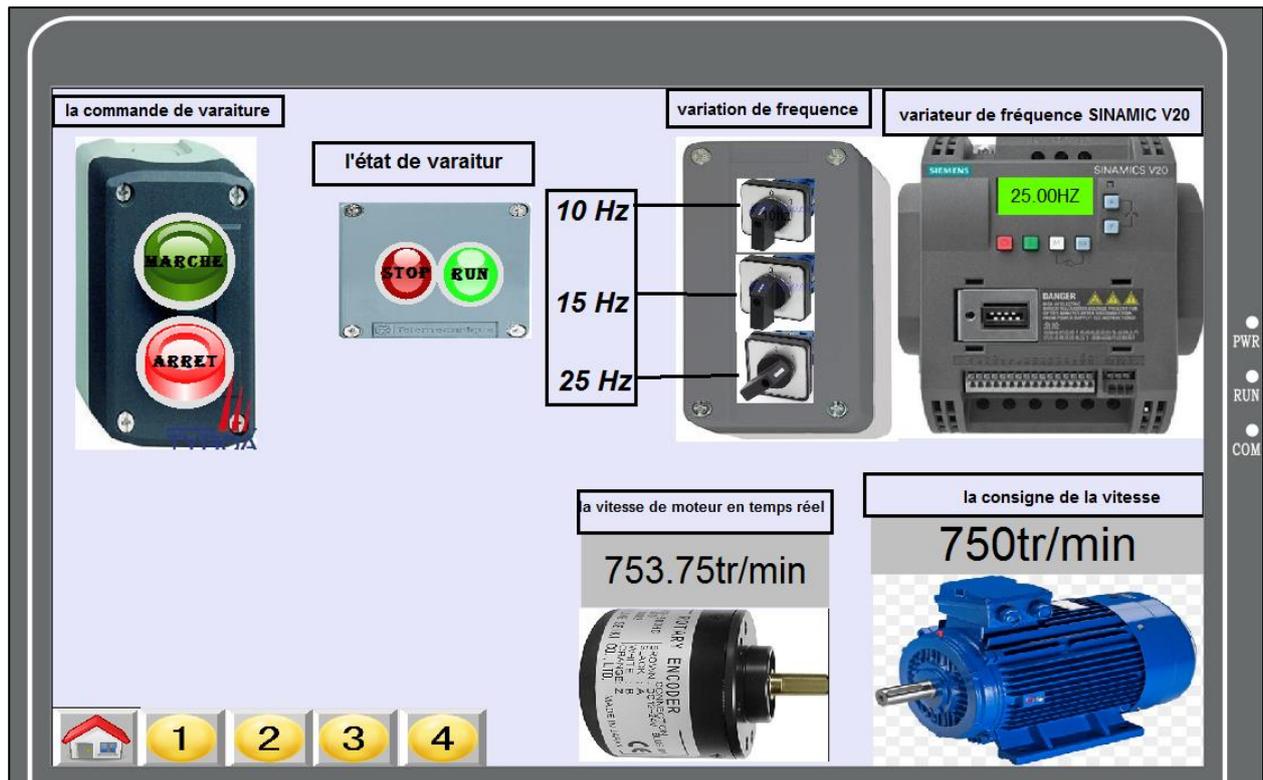


Figure IV.21. La deuxième fenêtre de l'IHM

- ❖ La troisième fenêtre représente la courbe de l'erreur de vitesse et leur valeur instantanée.

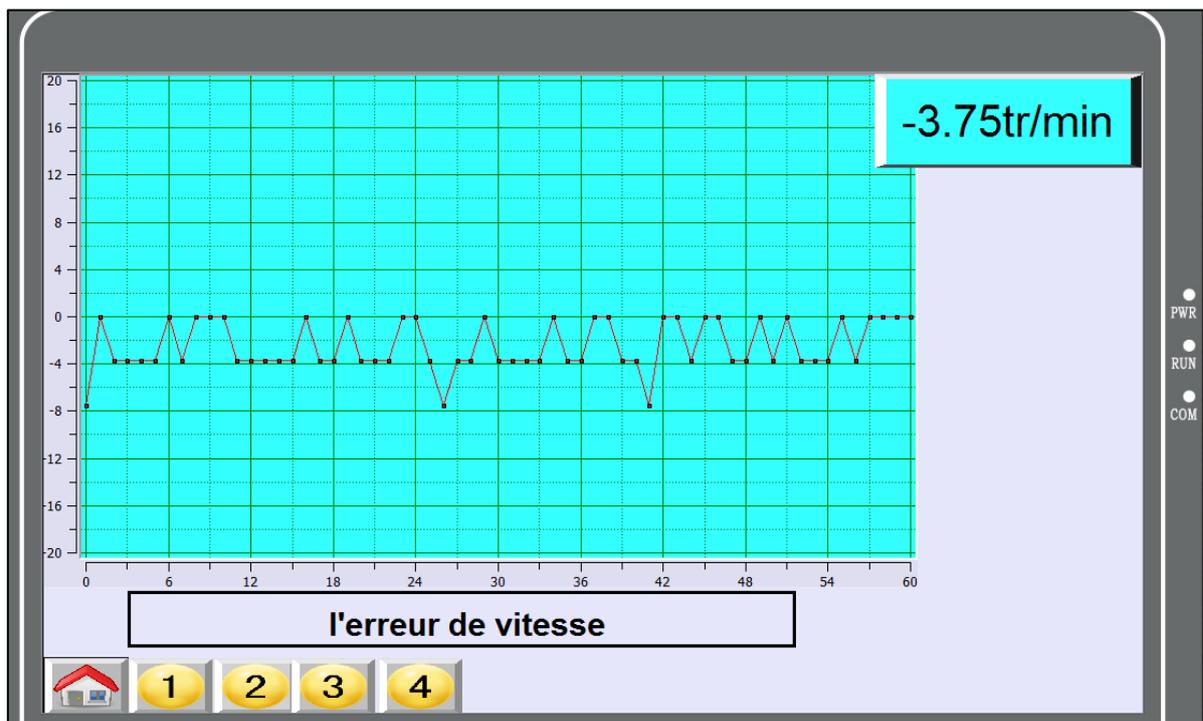


Figure IV.22. La courbe de l'erreur entre la vitesse et la consigne

- ❖ La quatrième fenêtre représente la courbe de la cosigne et la vitesse réelle.



Figure IV.23. La courbe de vitesse réelle de moteur

- ❖ La cinquième fenêtre représente la courbe de la variation de fréquence.

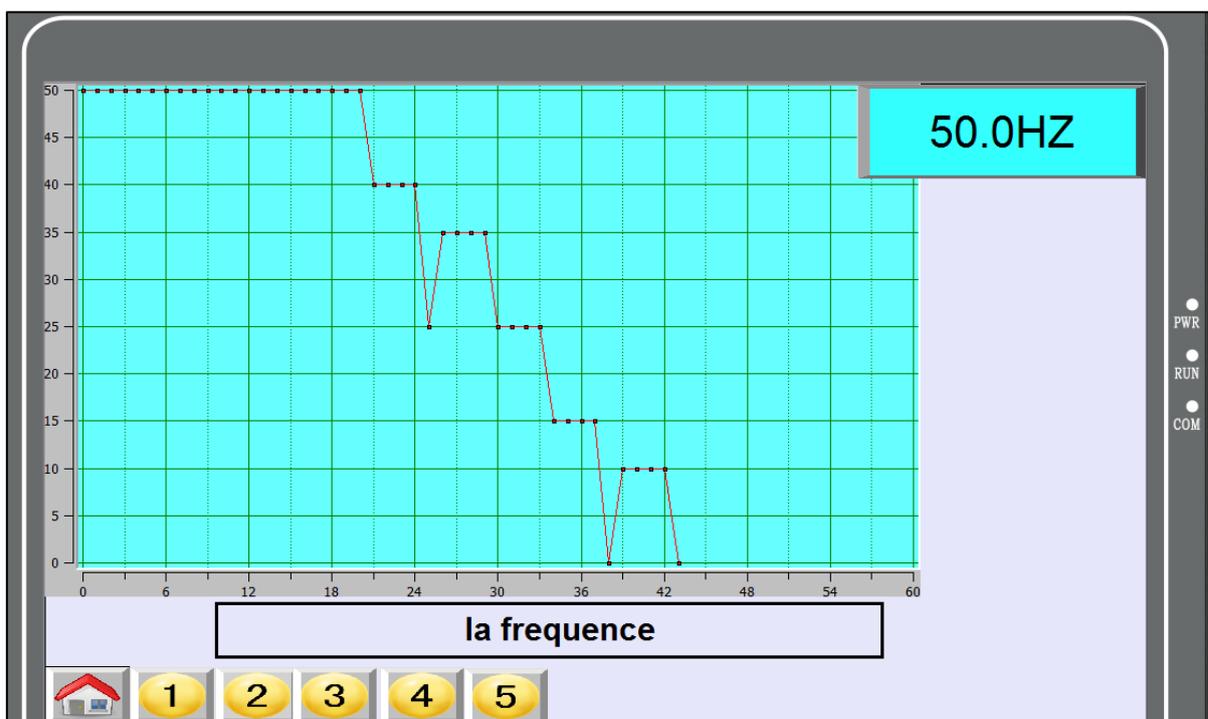


Figure IV.24. La courbe de la variation de fréquence

IV.4.3. Le deuxième mode de fonctionnement Cn010 – Régulation USS

Le variateur SINAMICS V20 prend en charge la communication avec les API Siemens via le protocole USS, pour appliquer ce protocole on utilise un module d'interface RS422/485. USS est le réglage de bus par défaut. Un câble à paire torsadée blindé est recommandé pour la communication RS-485.

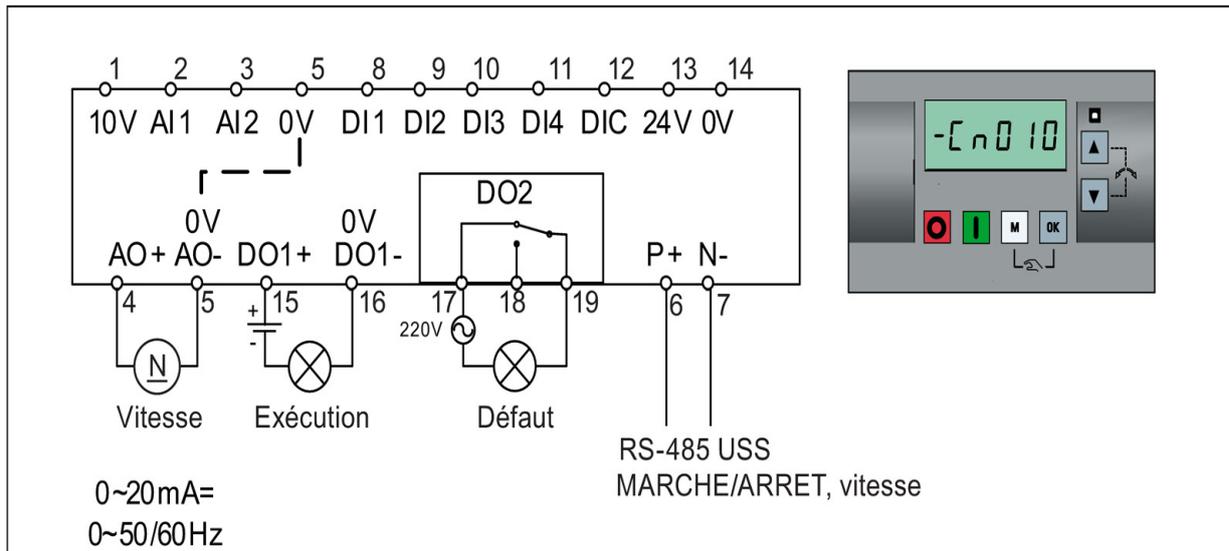


Figure IV.25. Le macro de connexion Cn010

IV.4.3.1 Réglages des paramètres de macro de connexion Cn010

Le tableau ci-dessous représente les paramètres réglés du variateur pour le macro de connexion Cn010.

Paramètre	Description	Valeur par défaut pour Cn010	Commentaires
p0700	Sélection de la source de commande	5	RS-485 en tant que source de commande
p1000	Sélection de la fréquence	5	RS-485 en tant que consigne vitesse
p2023	Sélection de protocole RS-485	1	Protocole USS
p2010	Vitesse de transmission USS/MODBUS	8	Vitesse de transmission 38 400 bit/s
p2011	Adresse USS	1	Adresse USS pour variateur

Tableau IV.2. Paramètres de Macro de connexion Cn010

IV.4.3.2. Programmation de l'API pour le macro de connexion Cn010

a. Table de variable standard

Dans cette table on a représenté toutes les variables utilisées de noter programme pour le macro cn003.

Variables API							
	Nom	able des variables	Type de données	Adresse	Réma...	Visibl...	Acces...
1	lamp vert	able de variables s...	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	lamp rouge	able de variables s...	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	HSC 2	able de variables s...	DInt	%ID1004	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	vitesse de moteur réel	able de variables s...	Real	%MD30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	botton march	able de variables s...	Bool	%IO.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	bottun arrêt	able de variables s...	Bool	%IO.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	arrêt cn010	able de variables s...	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	inversé le sens	able de variables s...	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	arrêt rapid	able de variables s...	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	la frequence	able de variables s...	Real	%MD210	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	spd sp	able de variables s...	Real	%MD300	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	la consigne de vitesse cn010	able de variables s...	Real	%MD270	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	l'errur cn010	able de variables s...	Real	%MD40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	ce variable pour tracer les courbe	able de variables s...	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	freq af cn 010	able de variabl...	Real	%MD400	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	arret lent	able de variables s...	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	output PID	able de variables s...	Real	%MD500	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	activé PID	able de variables s...	Bool	%M1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	march cn010	able de variables s...	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	<Ajouter>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure IV.26. Table des variables standards de macro Cn010

b. Le block d'organisation (programme cycle)

1. La figure suivante représente le bloc intégré dans notre automate programmable S7-1214C. L'instruction "USS_DRIVE" échange des données avec l'entraînement en générant des messages de requête et en analysant les messages de réponse de l'entraînement.

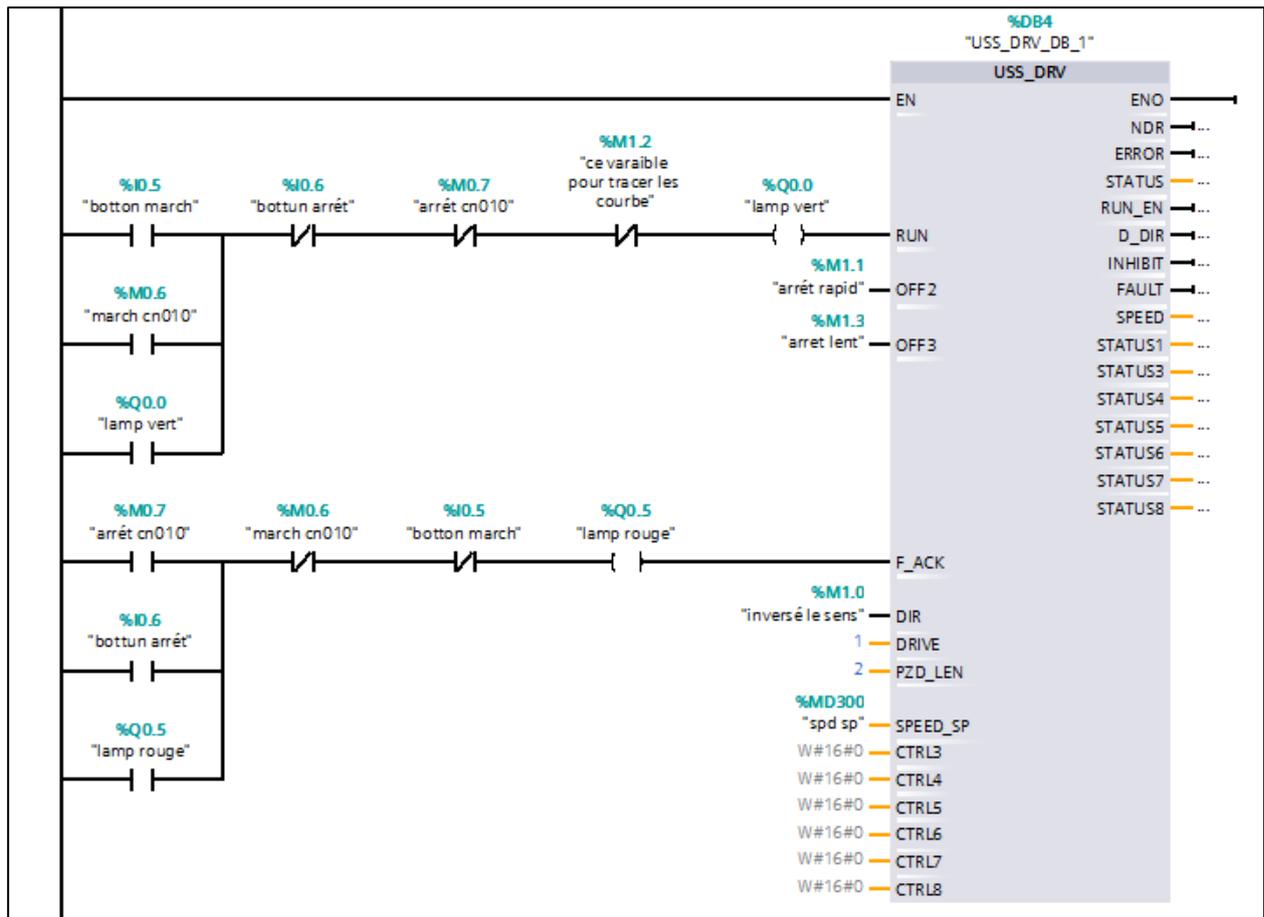


Figure IV.27. Bloc fonctionnel intégré "USS_DRIVE"

Le tableau suivant indique les paramètres importants de l'instruction "USS_DRIVE".

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
RUN	Input	BOOL	I, Q, M, D, L ou constante	Bit de démarrage de l'entraînement : Si ce paramètre à la valeur TRUE, cette entrée permet le fonctionnement de l'entraînement à la vitesse prédéfinie
OFF2	Input	BOOL	I, Q, M, D, L ou constante	Bit "ralentissement jusqu'à l'arrêt" : Si ce paramètre a la valeur FALSE, ce bit provoque alors le ralentissement de l'entraînement sans freiner.
OFF3	Input	BOOL	I, Q, M, D, L ou constante	Bit d'arrêt rapide - Si ce paramètre a la valeur FALSE, alors ce bit provoque un arrêt rapide en freinant

				l'entraînement.
F_ACK	Input	BOOL	I, Q, M, D, L ou constante	Bit d'acquiescement d'une erreur - Ce bit permet de réinitialiser le bit d'erreur d'un entraînement. Il est activé après la suppression de l'erreur et l'entraînement détecte alors que l'erreur précédente n'a plus à être signalée.
DIR	Input	BOOL	I, Q, M, D, L ou constante	Commande du sens de l'entraînement - Ce bit est activé lorsque l'entraînement doit tourner vers l'avant (si SPEED_SP est positif).
DRIVE	Input	USINT	I, Q, M, D, L ou constante	Adresse de l'entraînement : Cette entrée est l'adresse de l'entraînement USS. La plage valide est comprise entre l'entraînement 1 et l'entraînement 16.
PZD_LEN	Input	USINT	I, Q, M, D, L ou constante	Longueur de mots - C'est le nombre de mots de données PZD. Les valeurs valides sont 2, 4, 6 ou 8 mots. La valeur par défaut est 2
SPEED_SP	Input	REAL	I, Q, M, D, L ou constante	Vitesse de consigne - C'est la vitesse de l'entraînement en % par rapport à la fréquence configurée. Une valeur positive signifie que l'entraînement tourne vers l'avant (si DIR a la valeur TRUE).

Tableau IV.3. Paramètres de l'instruction "USS_DRIVE".

2. Ce réseau représente le bloc fonctionnel "Bloc_3_DB" pour transmettre le pourcentage (%) de fréquence dans le block USS par l'entrée « SPEED_SP » et calculer la consigne de vitesse et afficher la fréquence dans L'IHM.

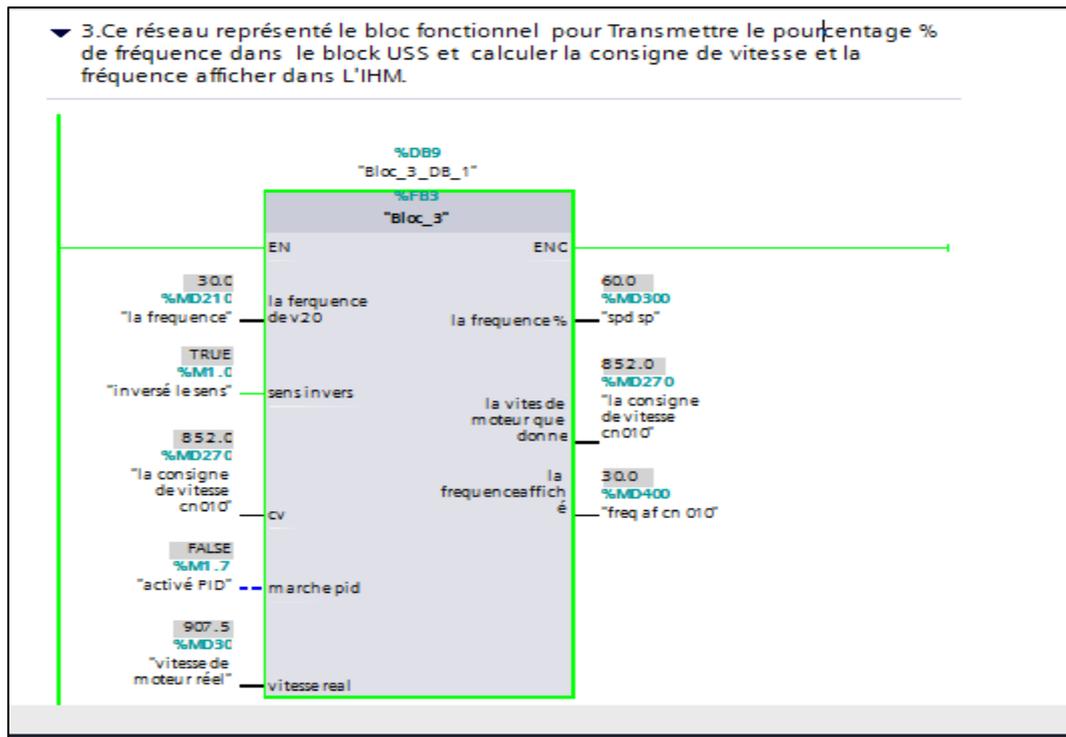


Figure IV.28. Bloc fonctionnel "Bloc_3_DB"

3. Dans la figure ci-dessous on va présenter le réseau qui calcul l'erreur de vitesse entre la consigne et la vitesse réelle donnée par l'encodeur, en utilisant la fonction mathématique «SUB».

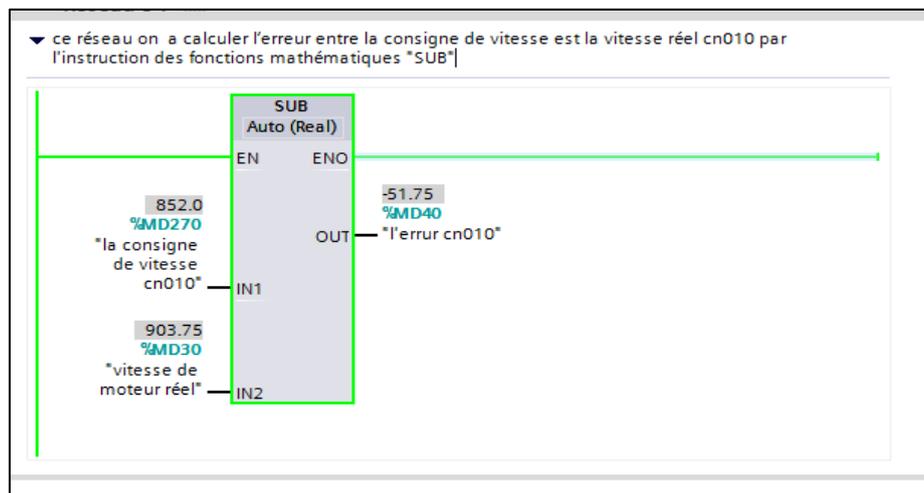


Figure IV.29. Réseau pour calculer l'erreur entre la consigne et la vitesse réelle.

c. le block d'organisation (cyclic interrupt)

1. La figure suivante représente le bloc intégré dans notre automate programmable. L'instruction "USS_PORT" traitée la communication dans le programme via le réseau

USS. En utilisant une instruction "USS_PORT" pour chaque port de communication PTP afin de commander la transmission vers ou depuis un entraînement. Le programme doit exécuter l'instruction "USS_PORT" suffisamment souvent pour qu'il n'y ait pas de dépassement de temps dans l'entraînement. Pour ce faire, on appelle l'instruction "USS_PORT" dans un OB d'alarme cyclique afin d'éviter les dépassements de temps dans l'entraînement et afin que les dernières mises à jour de données USS soient disponibles pour les appels de "USS_DRIVE".

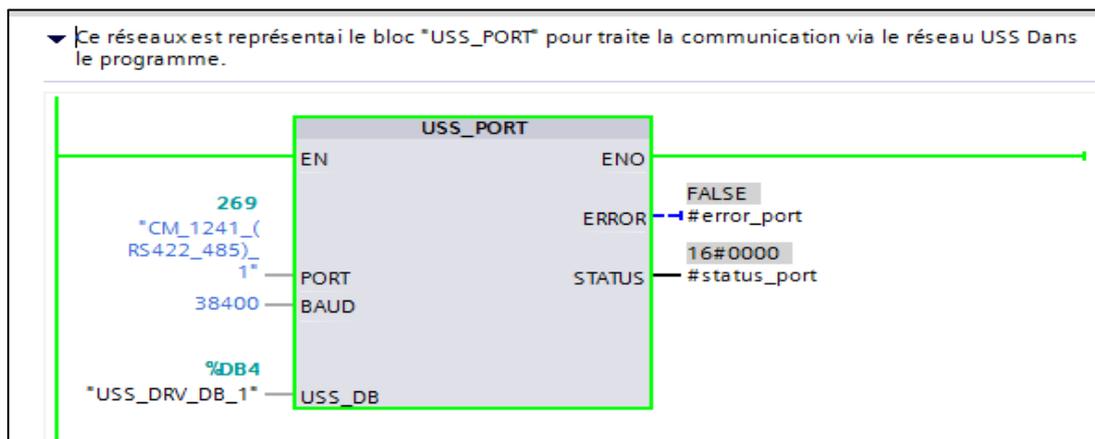


Figure IV.30. Bloc fonctionnel intégré "USS_PORT"

Le tableau suivant indique les paramètres importants de l'instruction "USS_PORT".

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
PORT	Input	PORT	D, L ou constante	Identification du port de communication PtP Constantes à laquelle il peut être fait référence dans la table des variables standards dans l'onglet "Constantes".
BAUD	Input	DINT	I, Q, M, D, L ou constante	Vitesse de transmission pour la communication USS.
USS_DB	Input	USS_BASE	D	Référence au DB d'instance de l'instruction " <u>USS_DRIVE</u> ".
ERROR	Output	BOOL	I, Q, M, D, L	Si une erreur survient, ERROR est mis à TRUE. Un code d'erreur correspondant est émis sur la sortie STATUS.

STATUS	Output	WORD	I, Q, M, D, L	Valeur d'état de la requête. Elle indique le résultat du cycle ou de l'initialisation. D'autres informations sont disponibles pour quelques codes d'état dans la variable " <u>USS_Extended_Error</u> ".
--------	--------	------	---------------	--

Tableau IV.3. Paramètres de l'instruction "USS_PORT".

d. Bloc fonctionnel

On a utilisé deux blocs fonctionnels, le premier bloc "Bloc_1_DB" qui présente dans le premier mode de fonctionnement Cn003 pour le but de calculer la vitesse réelle du moteur, et le deuxième bloc "Bloc_3_DB" pour transmettre le pourcentage (%) de fréquence dans le block USS par l'entrée « SPEED_SP » et pour calculer la consigne de vitesse et afficher la fréquence dans L'IHM.

❖ *Le bloc fonctionnel "Bloc_3_DB"*

1. La figure ci-dessous représente la table des variables pour bloc "Bloc_3_DB"

le projet final ▶ PLC_1 [CPU 1212C DC/DC/Rly] ▶ Blocs de programme ▶ Bloc_3 [FB3]							
Bloc_3							
	Nom	Type de données	Valeur par déf.	Rémanence	Accessible ...	Visible da...	Vale
1	Input						
2	la ferquence de v20	Real	0.0	Non ré...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	sens invers	Bool	false	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	cv	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	marche pid	Bool	false	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	vitesse real	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Output						
8	la frequence %	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	la vites de moteur que..	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	la frequenceaffiché	Real	0.0	Non réman...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure IV.31. La table des variables pour bloc "Bloc_3_DB"

2. Le programme du bloc "Bloc_3_DB" pour le but de transmettre le pourcentage % de fréquence dans le block USS par l'entrée « SPEED_SP » et calculer la consigne de vitesse et afficher la fréquence dans L'IHM.

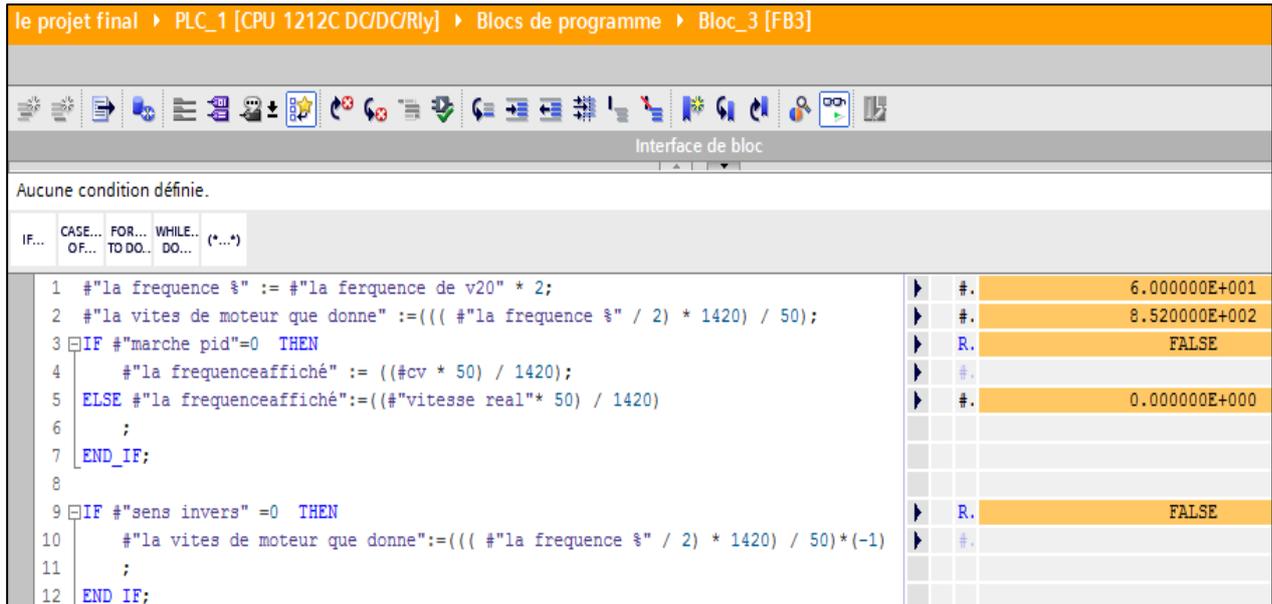


Figure III.32. Le programme du bloc "Bloc_3_DB".

IV.4.3.3. Programme l'IHM pour le macro de connexion Cn010

Nous avons utilisé cinq fenêtres pour le mode de fonctionnement Cn010.

- e. La première fenêtre est pour représenter le schéma fonctionnel de macro Cn010.

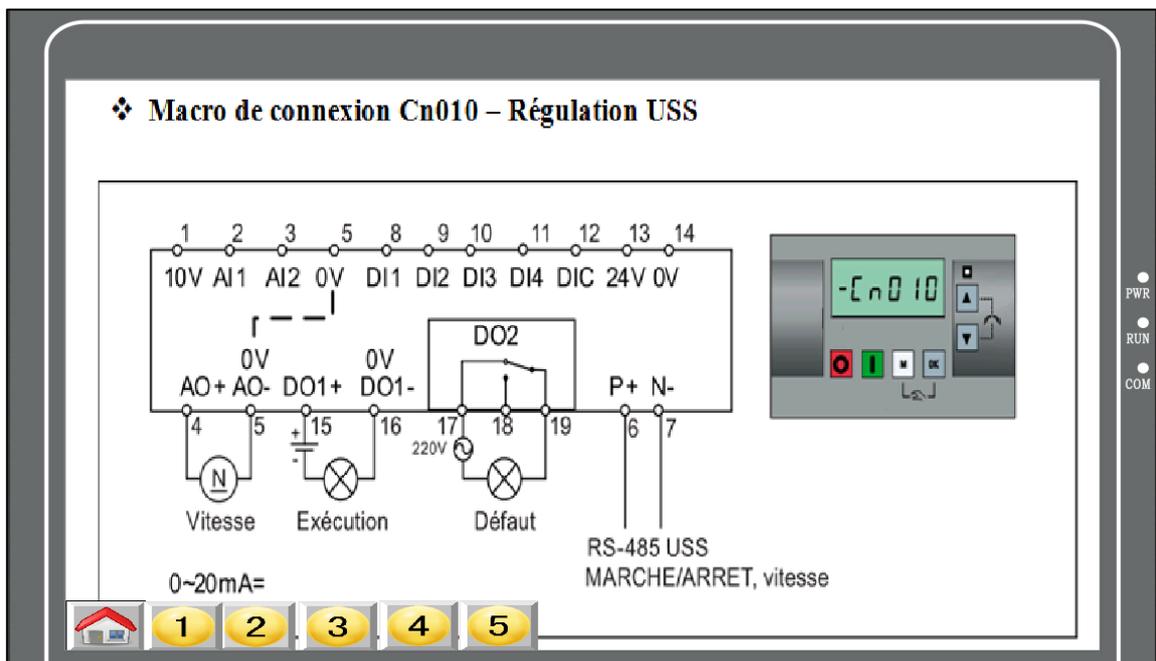


Figure IV.33. La première fenêtre de l'IHM pour macro Cn010.

- f. La deuxième fenêtre pour commander le variateur et visualiser leur état ainsi les deux vitesses, la consigne et la vitesse réelle (encodeur).

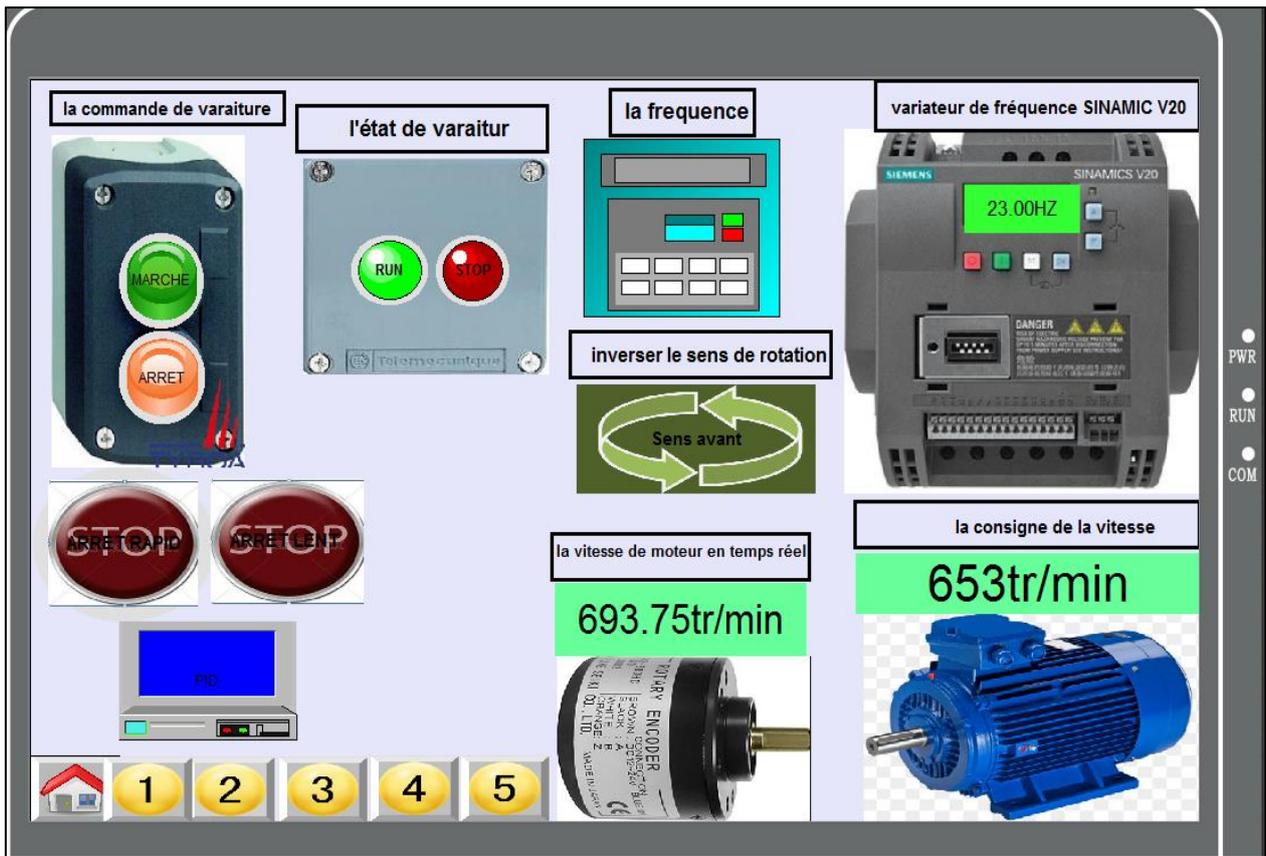


Figure IV.34. La deuxième fenêtre de l'IHM pour macro Cn010.

g. La troisième fenêtre représente la courbe de l'erreur de vitesse et leur valeur instantanée.

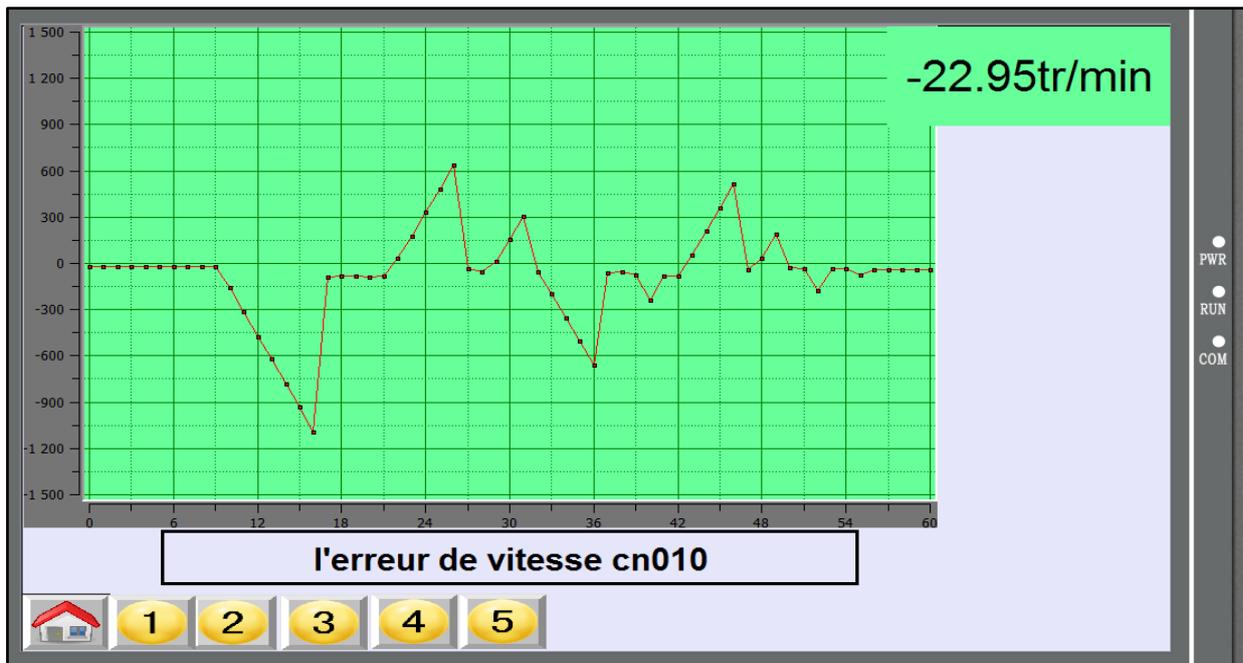


Figure IV.35. Courbe de l'erreur de vitesse

h. La quatrième fenêtre représente la courbe de vitesse réelle et leur valeur instantanée.

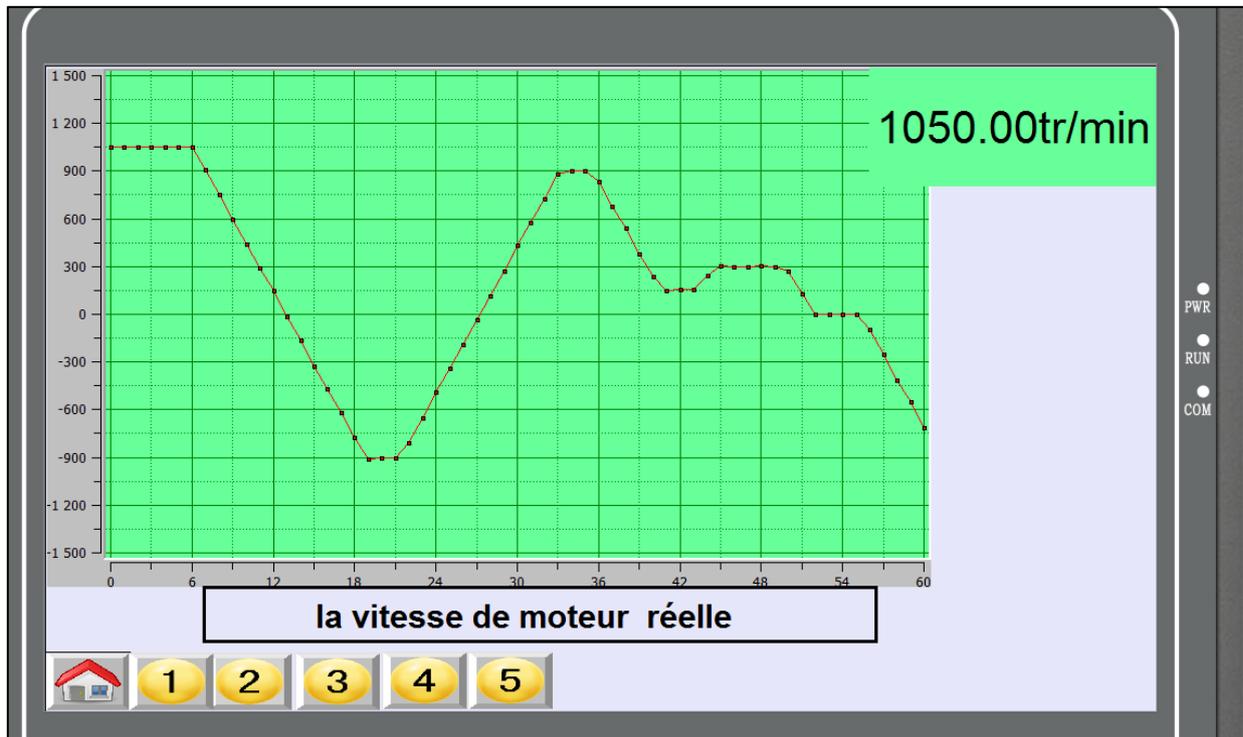


Figure IV.36. Courbe de vitesse réelle du moteur.

i. La cinquième fenêtre représente la courbe variation de fréquence et leur valeur.

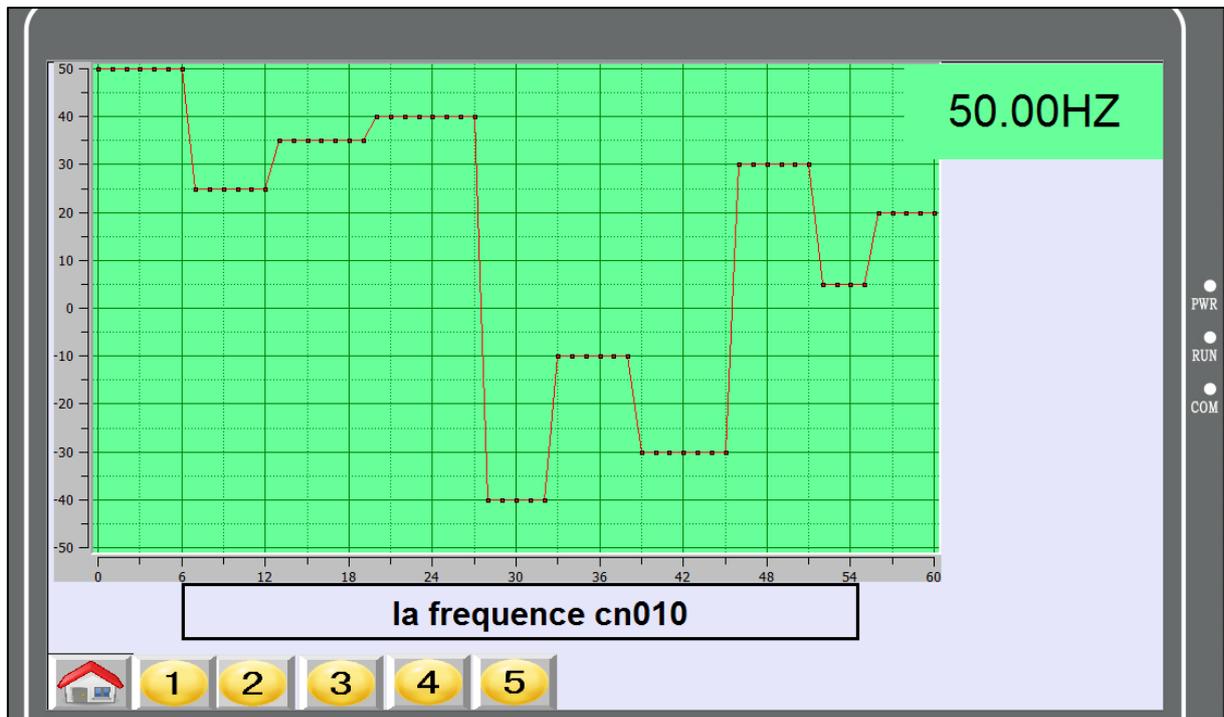


Figure IV.37. Courbe variation de fréquence et leur valeur.

A partir de la troisième fenêtre qui représente la courbe de l'erreur de vitesse et leur valeur instantanée, on remarque que l'erreur est importante entre la vitesse réelle et la consigne. Donc le moteur ne suit pas parfaitement la consigne de vitesse. C'est un obstacle dans le domaine industriel, pour régler ce problème il faut diminuer l'erreur entre la consigne et la vitesse réelle du moteur, pour ce la on utilise un régulateur PID intégré dans l'automate programmable industriel.

IV.4.3.4. Le Régulateur PID

Le régulateur PID, appelé aussi correcteur PID (proportionnel, intégral, dérivé) est un système de contrôle permettant d'améliorer les performances d'un asservissement, c'est-à-dire un système ou procédé en boucle fermée. C'est le régulateur le plus utilisé dans l'industrie où ses qualités de correction s'appliquent à de multiples grandeurs physiques.

a. Ajout de bloc PID control

L'instruction PID_Compact met à disposition un régulateur PID avec optimisation intégrée pour les modes automatique et manuel. L'appel de l'instruction PID_Compact s'effectue durant l'intervalle de temps constant du temps de cycle de l'OB. Pour ajouté un block de PID dans TIA PORTAL il faut suivre les étapes suivantes :

1. Dans la partie des l'instruction clique sur technologie.
2. Clique sur PID control.
3. Clique sur le PID compact.
4. Puis ajouter dans le block d'organisation (cyclic interrupt).

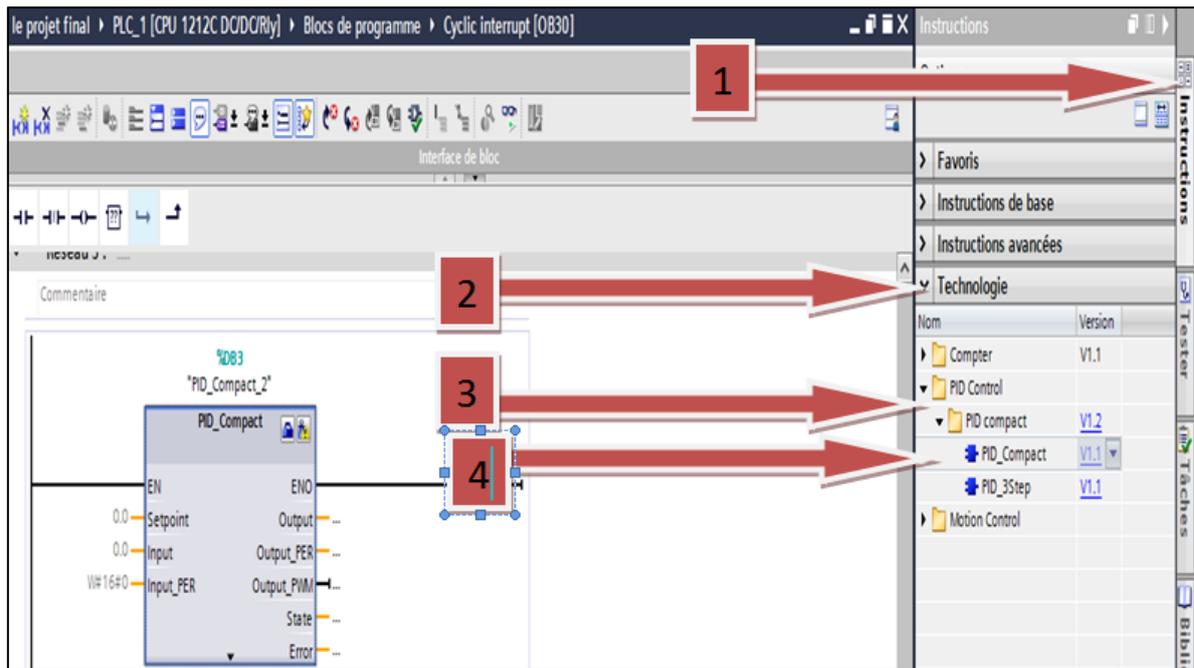


Figure IV.38. Ajout de bloc PID control

b. Paramètres d'entrée et sortie du contrôleur PID_Compact.

Les paramètres du contrôleur PID_Compact qui nous avons utilisé pour diminuer l'erreur entre la consigne et la vitesse réelle du moteur. Le tableau suivant indique les paramètres d'entrée PID_Compact.

Paramètre	Type de données	Valeur par défaut	Description
Setpoint	REAL	0.0	Consigne du régulateur PID en mode automatique
Input	REAL	0.0	Une variable du programme utilisateur est utilisée comme source de la mesure.
Output	REAL	0.0	Valeur de réglage au format REAL

Tableau IV.5. Paramètres du contrôleur PID_Compact.

c. Description du bloc PID_Compact

Nous avons utilisé le bloc PID_Compact pour diminuer l'erreur entre la consigne et la vitesse réelle du moteur. On a utilisé la technologie du régulateur PID intégré dans l'automate programmable industriel. On a pris comme la consigne du régulateur PID (set point) la consigne de vitesse qui calculer d'après le bloc "Bloc_3_DB " et on a utilisé la vitesse réelle du moteur qui mesurée par l'encodeur comme input dans le paramètre d'entrée PID_Compact. À la fin on crée un variable au format REAL pour donnée la valeur de réglage (output PID). Cette valeur représente le pourcentage de régulation PID qui transmettre par l' l'instruction "Copier valeur" "MOVE", cette instruction permet de transférer le contenu de Paramètre sortie de PID (output PID) dans l'entrée SPEED_SP de bloc "USS_DRIVE" sous la forme consigne de vitesse C'est la vitesse de l'entraînement en pourcentage (%). La figure suivante représente le bloc PID_Compact qui nous avons ajouté dans le block d'organisation (cyclic interrupt).

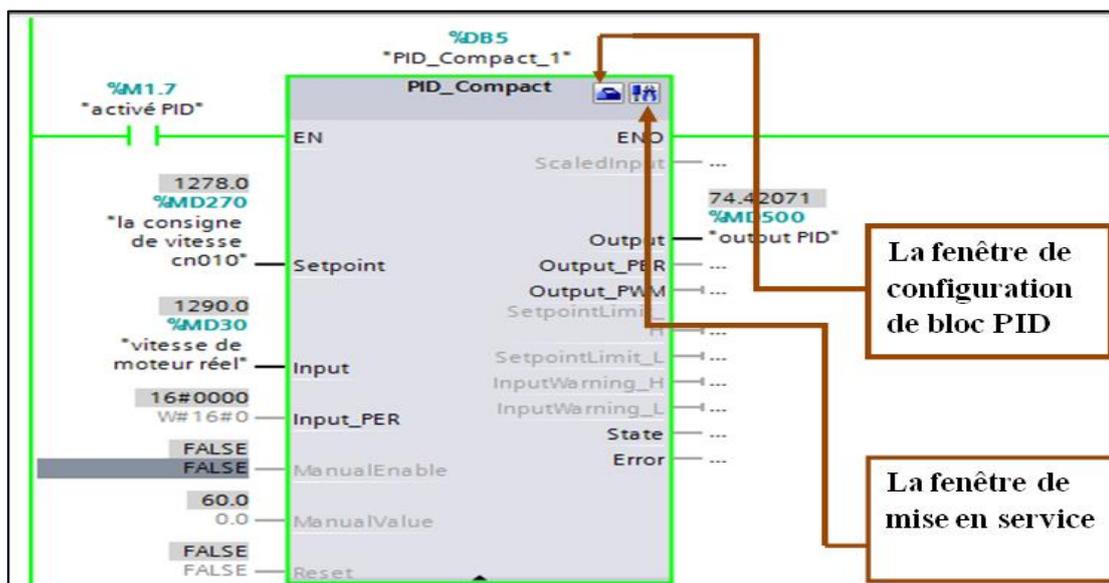


Figure IV.39. Représentations du bloc PID_Compact.

d. Réglage des paramètres du bloc PID_Compact

Pour configurer les paramètres du bloc PID_Compact, on click sur la fenêtre de configuration dans le bloc PID qui située au dessus dans le bloc sous la forme d'une mallette pour donner le type de régulation et configurer les paramètres des entrées/sorties, le sens de rotation , sélectionné les limites de mesure, et réglé les paramètres PID (gain proportionnel, temps d'intégration, temps de dérivation) la configuration des ces paramètres on le laisse en mode automatique. Les figures suivantes expliquent comment configurer les paramètres utilisés pour la régulation PID.

1. La figure suivante représente la configuration de type de régulation, paramètres des entrées/sorties et le sens de rotation.

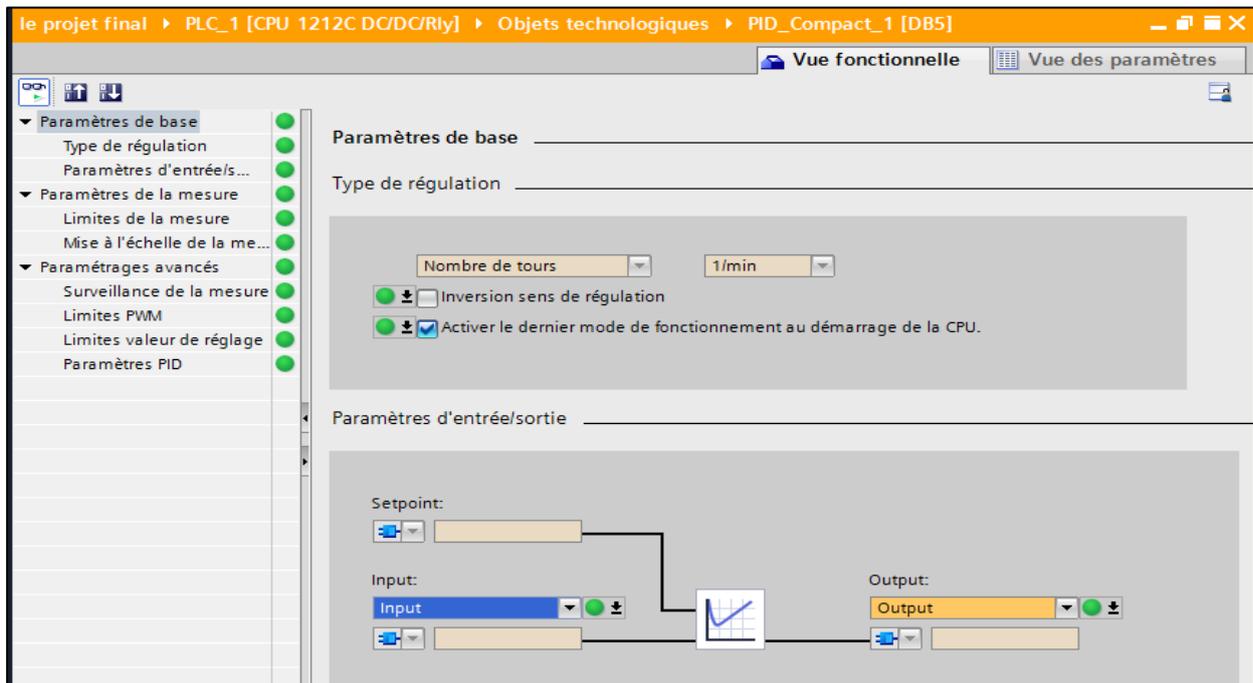


Figure III.40. Configuration de type de régulation, paramètres d'entrées/sorties et le sens de rotation.

2. La figure ci-dessous représente les paramètres de mesure.

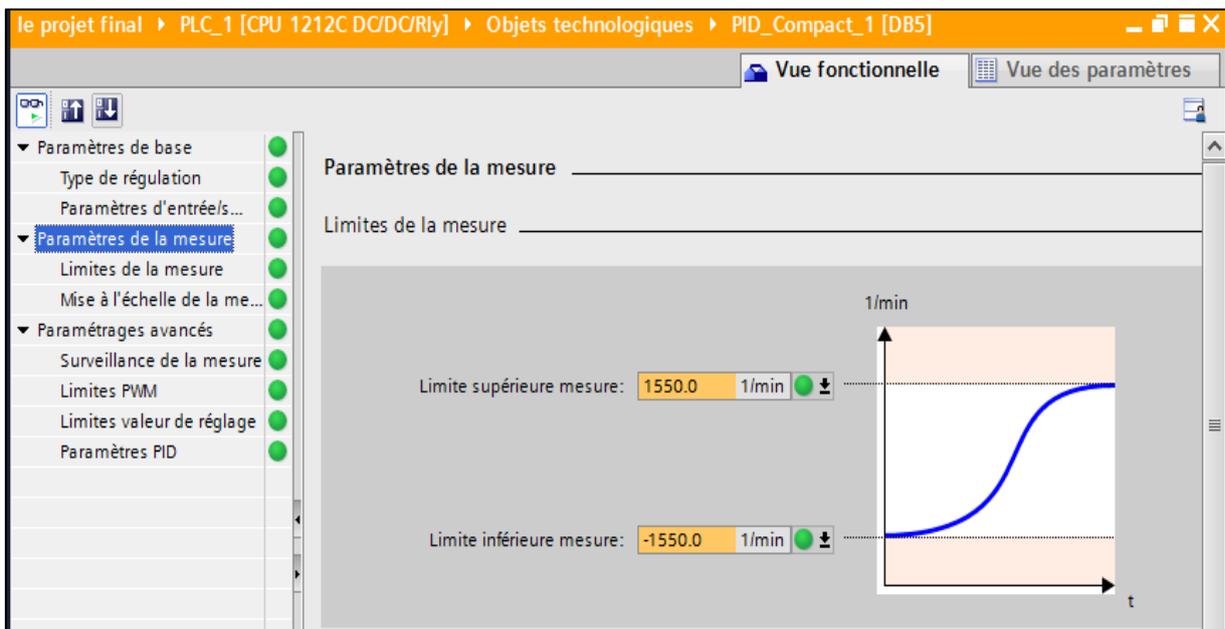


Figure IV.41. Configuration des paramètres de mesure

3. La figure suivante représente les paramètres du PID (Gain proportionnel, Temps d'intégration, Temps de dérivation).

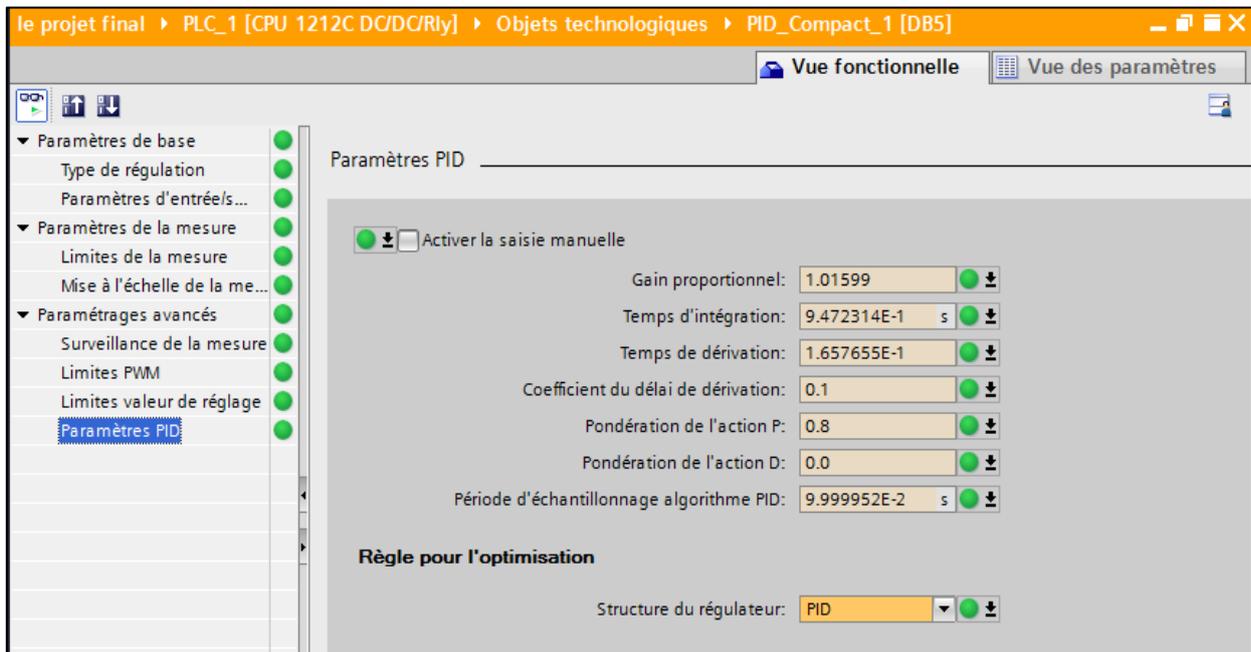


Figure IV.42. Les paramètres du contrôleur PID

e. La mise en service du contrôleur PID_Compact

PID_Compact est démarré dans le dernier mode de fonctionnement actif lors du démarrage de la CPU, Pour la mise en service de bloc PID_Compact il se fait de clic sur la fenêtre de mise en service. Pour démarré la régulation PID il se fait de clic sur "START" et "START PID_Compact". La figure suivante représente la fenêtre mise en service du contrôleur PID_Compact et leur régulation.

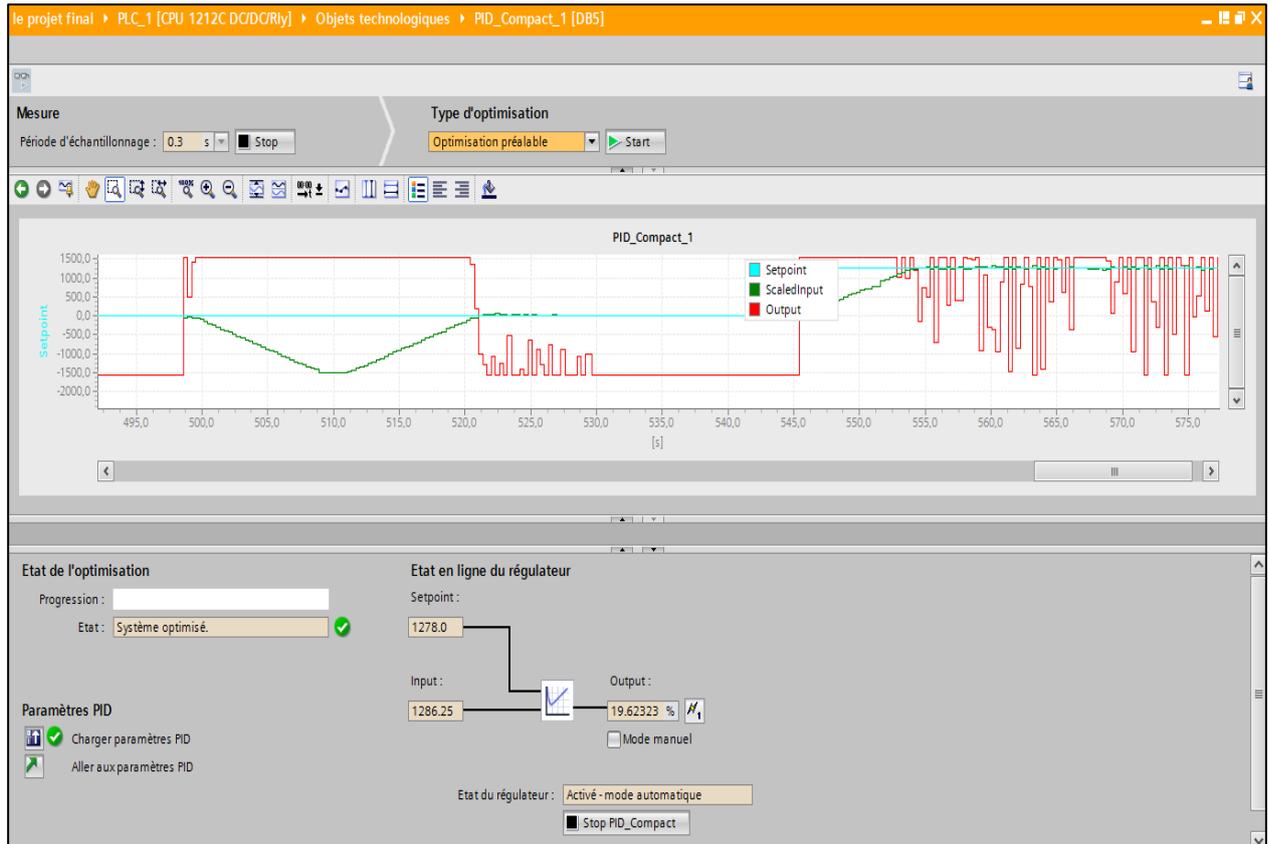


Figure IV.43. Représentation de la fenêtre mise en service du contrôleur PID_Compact et leur régulation.

e.1.la transmission de la valeur de sortie PID

La figure ci-dessous représente le réseau de transmettre le pourcentage de fréquence de la sortie PID à l'entrée « SPEED_SP » et le bloc "USS_DRIVE" par l'instruction "Copier valeur" "MOVE" cette instruction permet de transférer le contenu de l'opérande à l'entrée IN dans l'opérande à la sortie OUT1. Le transfert s'effectue toujours dans le sens croissant des adresses.

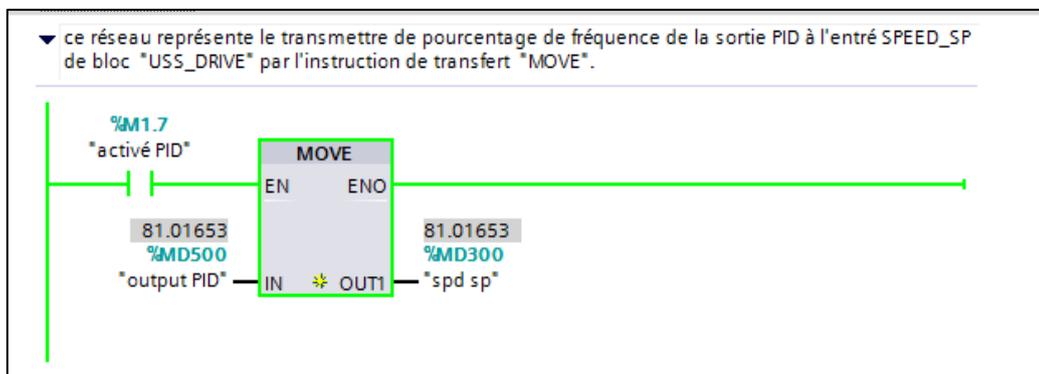


Figure IV.44. Réseau de transmettre de pourcentage de fréquence de la sortie PID à l'entrée SPEED_SP de bloc "USS_DRIVE".

Après l'utilisation du bloc PID_Compact dans notre programme, on observe que l'erreur entre la consigne et la vitesse réelle est diminuée. Les résultats que nous avons obtenus montrent que le moteur suit parfaitement les consignes. La figure suivante représente la courbe de l'erreur de vitesse et leur valeur instantanée d'après la régulation PID.

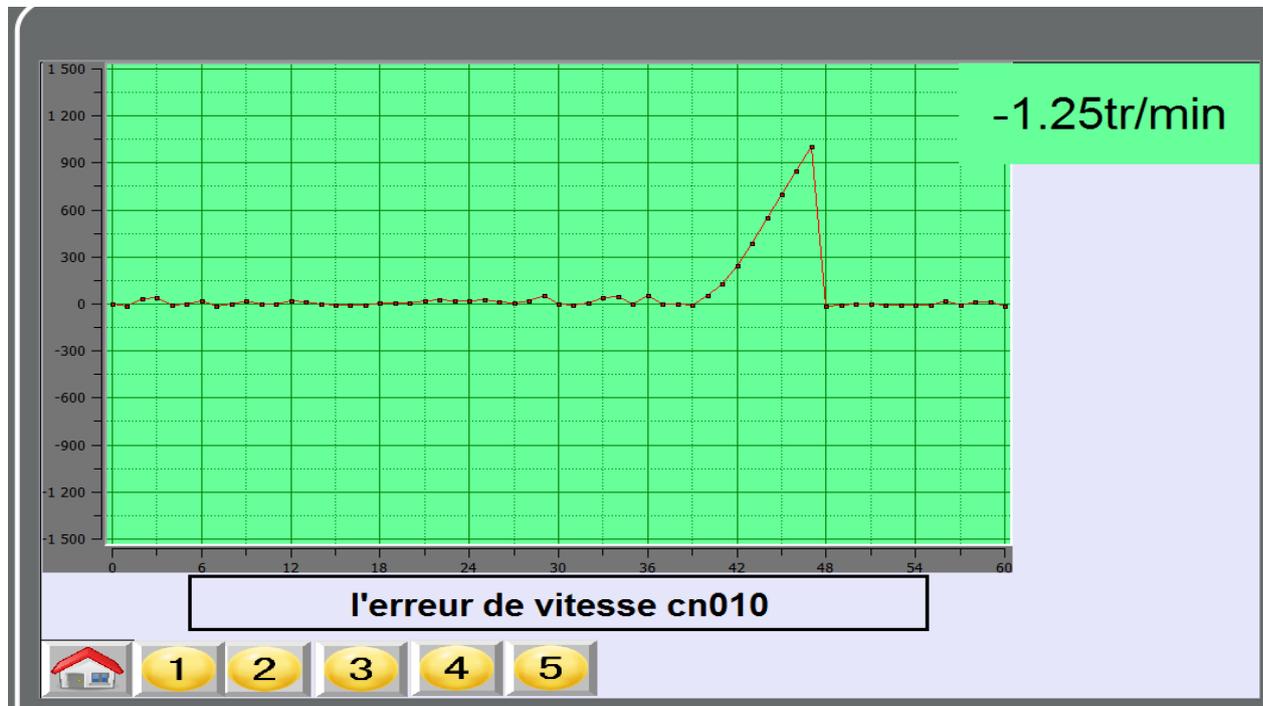


Figure IV.45. La courbe de l'erreur de vitesse et leur valeur (Régulation PID).

IV.5. Conclusion

Dans ce chapitre, nous avons présenté les raccordements des différents éléments utilisés pour la réalisation de notre banc d'essai et le logiciel utilisés pour la commande et la supervision notre projet. Ensuite nous avons parlé sur le réglage des paramètres du variateur de vitesse (SINAMICS V20) et les deux modes de fonctionnement qui nous avons utilisé Cn003 et Cn010.

On a présenté le premier mode de fonctionnement Cn003 et leur programme qui nous avons utilisé pour commander et supervisé en temps réel, en suite on a parlé sur le deuxième mode Cn010 (Régulation USS), leur paramètres et le programme qui nous avons utilisé, en remarque que l'erreur entre la vitesse réelle et la consigne est importante. Pour diminué l'erreur on a utilisé la technologie de régulateur PID intégré dans l'automate programmable industriel, par la suite on a vu que l'erreur de vitesse entre la consigne de vitesse et la vitesse réel est diminuée et que la vitesse du moteur suivie parfaitement la consignes.

Conclusion générale

Ce mémoire présente la réalisation et la commande d'un banc d'essai de la chaîne d'action « moteurs asynchrones et variateurs de vitesse » commandé par un API et supervisé avec un écran tactile.

Le premier chapitre aborde les moteurs asynchrones et le capteur de vitesse (encodeur incrémentale), Les moteur asynchrone jouent un rôle très important dans le domaine industriel, au début nous avons donné la définition et la constitution de ce dernier, par la suite on a parlé sur ces principes de fonctionnement et leur bobinage et couplage, en suite nous avons parlé sur la vitesse du moteur asynchrone et ces avantage et inconvénients, et après on a donné ces domaine d'utilisation.

A la fin on va passer au capteur de vitesse (l'encodeur incrémental) par donner la définition et la présentation de notre encodeur incrémental, par la suite nous avons présenté ces caractéristiques et ça correspondance de câblage.

Le deuxième chapitre concerne la variation de vitesse. La vitesse d'un moteur asynchrone dépend principalement de sa fréquence et pour varier la fréquence de ce dernier on utilise le variateur de vitesse. Ce dernier est un dispositif électronique de puissance constitué d'un redresseur, d'un circuit intermédiaire et d'un onduleur. Il fournit au moteur asynchrone une onde de tension à amplitude et fréquence variables tout en maintenant le rapport tension / fréquence sensiblement constant. En modifiant la fréquence de la tension d'alimentation du moteur, le variateur permet de modifier la vitesse du champ tournant du stator et donc la vitesse de rotation du moteur.

Le troisième chapitre concerne les systèmes automatisés. Ces derniers se divisent en trois parties, la partie commande, la partie opérative et la partie supervision. La partie commande est le centre de décision. Il donne des ordres à la partie opérative et reçoit ses comptes rendus, ces actions sont supervisé et contrôlé par la partie supervision qui sert d'interface homme machine, Nous avons présenté la procédure à suivre pour la création de notre programme sous TIA PORTAL V13, un aperçu des blocs contact a été donné ainsi que les compteurs rapides utilisés lors de la programmation.

Le quatrième chapitre concerne la réalisation et la commande d'un banc d'essai de la variation de vitesse d'un moteur asynchrone. On nous avons commandé le variateur de vitesse «SIEMENS SINAMICS V20» avec un automate programme industriel «SIEMENS S7 1200» via USS sur RS-485. Nous avons choisi l'interface RS-485 pour appliquer le protocole USS, et on l'a supervisé via un écran tactile IHM «SAMKOON». Pour avoir la vitesse réelle du moteur et les comparer avec la consigne de vitesse qui calculer d'après la fréquence celles qu'on a reçues depuis le variateur. Nous avons utilisés un encodeur incrémental raccordé avec L'API afin d'avoir le nombre de tours exactes du moteur. Les résultats que nous avons obtenus montrent que le moteur ne suit pas parfaitement les consignes qu'on lui envoyait. C'est un obstacle dans le domaine industriel, pour régler ce problème on a utilisé la technologie de régulateur PID intégré dans l'automate programmable industriel. Après l'utilisation de régulateur PID (proportionnel intégral dérivateur) on remarque que le moteur ne suit pas parfaitement les consignes.

Annexe

1. Le logiciel SKTOOL

Créé pour la série IHM de Samkoon SK à partir de Shenzhen Samkoon Technologie, SKTOOL est un logiciel d'édition de configuration de Samkoon IHM , SKTOOL prend en charge les pilotes de communication PLC des principaux fabricants mondiaux, tels que Samkoon, Siemens, Mitsubishi, Omron, Fujitsu, Panasonic, Schneider, Emerso , Le logiciel de configuration SKTOOL est un système de développement avec un environnement de développement intégré et de nombreuses fonctions avancées, il est facile à apprendre et à appliquer, ainsi les ingénieurs peuvent facilement développer leurs propres projets pour la configuration utilisée dans SKTOOL.

1.1. Création d'un projet et configuration

Cette opération est nécessaire créer un nouveau projet, Sélectionnez Nouveau projet dans le menu Fichier ou sur le bouton Nouveau projet, La figure ci-dessus représente la boîte de dialogue Nouveau projet dans laquelle l'utilisateur décide du nom, du chemin d'accès, du modèle et du mode d'affichage

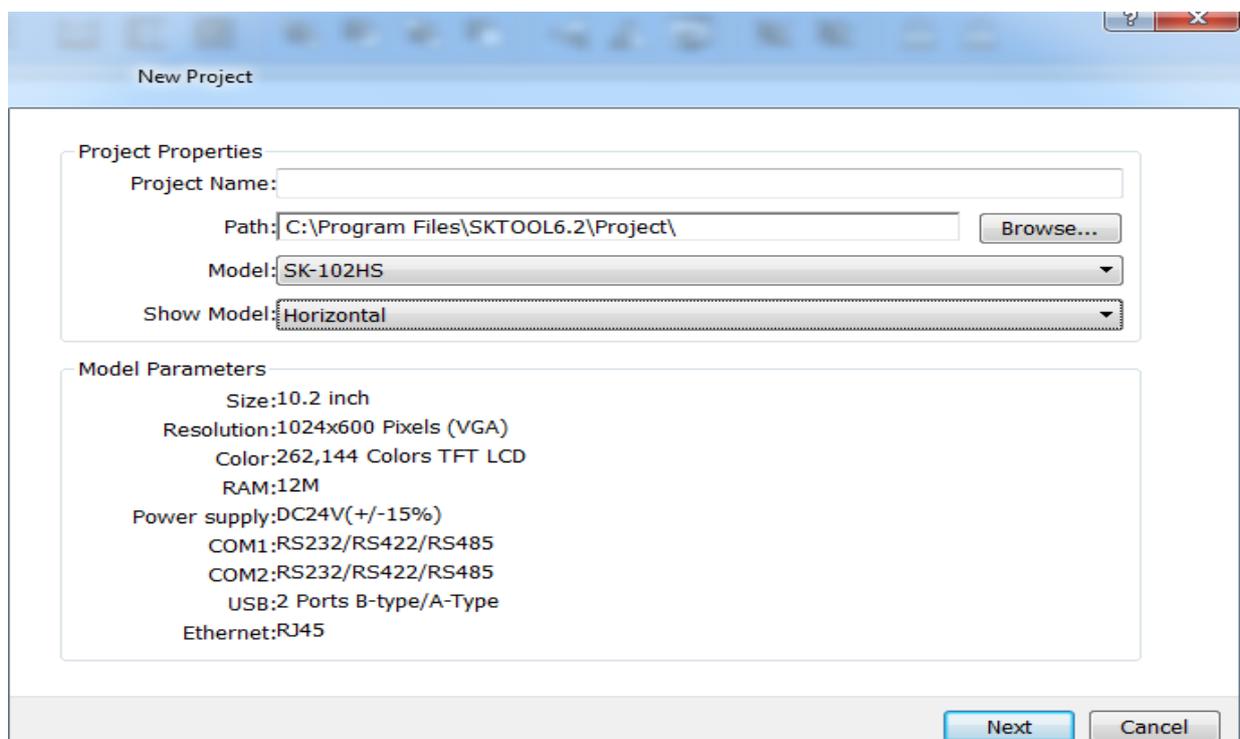


Figure 1 : Création d'un projet

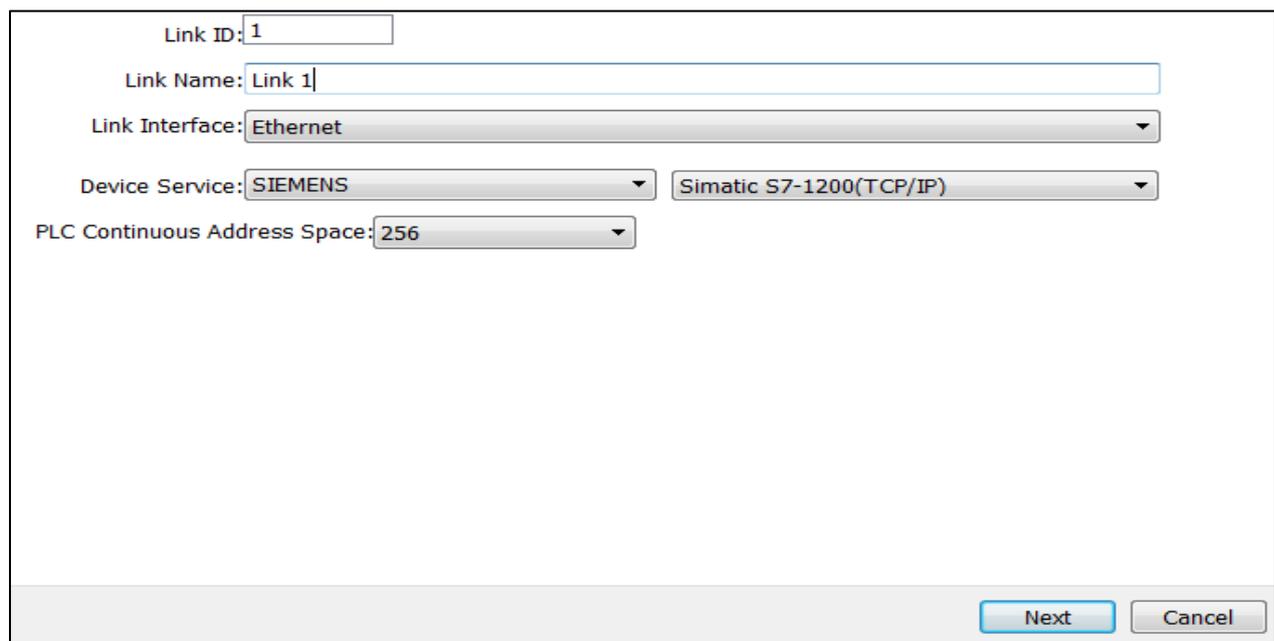
Le modèle spécifique est déterminé par le modèle IHM que utilisé, dans notre banc d'essai en utilise SK-102HS, puis cliquez sur Suivant pour définir le port de communication.

1.2. Boîte de dialogue Paramètres de communication

La boîte de dialogue Paramètres de communication sert à définir les paramètres de communication du port de communication de l'IHM. Seuls des paramètres de communication corrects peuvent assurer une communication normale entre l'automate et les écrans tactiles, Les paramètres de communication spécifiques dépendent du modèle d'automate à connecter.

Les paramètres le plus importants dans La boîte de dialogue Paramètres de communication :

1. Nom de la connexion: nom du port de communication.
2. Service de périphérique: sélection de la marque de l'automate et du type de CPU.
3. Connexion Interface: Select COM port or Ethernet port
4. Intervalle d'adresse continue de l'automate: définit l'espace adresse consécutive de l'automate; la valeur par défaut est recommandée.



The image shows a software dialog box titled 'Communication Parameters'. It contains several input fields and dropdown menus. The 'Link ID' field contains the number '1'. The 'Link Name' field contains 'Link 1'. The 'Link Interface' dropdown menu is set to 'Ethernet'. The 'Device Service' dropdown menu is set to 'SIEMENS', and the adjacent dropdown menu is set to 'Simatic S7-1200(TCP/IP)'. The 'PLC Continuous Address Space' dropdown menu is set to '256'. At the bottom right of the dialog box, there are two buttons: 'Next' and 'Cancel'.

Figure 2 : Boîte de dialogue Paramètre de communication

La figure ci-dessus représente la boîte de dialogue des paramètres du port de communication dans laquelle les utilisateurs choisissent les services de périphérique correspondants en fonction du type de périphérique de connexion. Pour utiliser une connexion Ethernet, choisissez l'option «Connexion directe (Ethernet)» dans le «Type de connexion». Cliquez sur «Suivant» dans la boîte de dialogue Paramètres de communication pour afficher une boîte de dialogue Nouvel écran.

1.3. Paramètres de l'écran

Boîte de dialogue Nouvel écran dans laquelle vous pouvez définir le nom et la couleur d'arrière-plan des écrans. Le nom par défaut est un numéro d'écran. la couleur d'arrière-plan par défaut est le bleu. Lorsque nous créons chaque fois une nouvelle image, le logiciel de configuration SKTOOL attribue un numéro d'écran au nouvel écran pour indiquer l'ordre d'affichage du projet en cours.

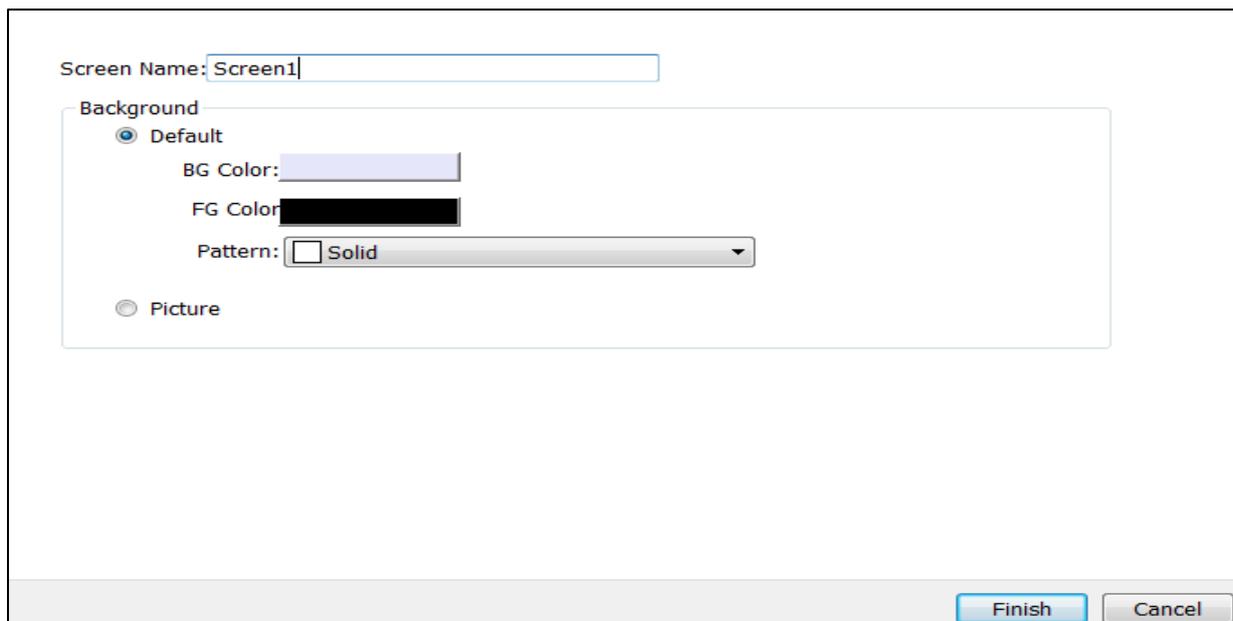


Figure 3 : Nouvelle boîte de dialogue

Cliquez sur Terminer pour terminer les étapes ci-dessus, le système ouvrira immédiatement l'écran que vous venez de créer.

1.4. La propriété du port de communication

La propriété du port de communication est appliquée pour définir les paramètres de communication entre l'IHM et les périphériques connectés. Double-cliquez sur "Link" et "Link 1" qui se trouvent dans Coordinateur de projet, comme le montre dans la figure III.10

General Parameter

Link ID: 1

Link Name: Link 1

Link Interface: Ethernet

HMI Site: Local Setting COM port (master-slave mode) port: 1

Injection Services: SIEMENS Simatic S7-1200(TCP/IP)

HMI IP

Ip: 192 . 168 . 0 . 10

Subnet mask: 255 . 255 . 255 . 0

Gateway: 192 . 168 . 0 . 1

Réglage des paramètres d'adresse IP pour permettre de réaliser la communication entre l'IHM et le dispositif de commande par Ethernet port.

ok cancel

Figure 4 : Boîte de dialogue propriété du port

Pour régler les paramètres d'adresse IP il nous faut cocher la case IHM IP :

- IP : c'est l'adresse IP spécialisée pour IHM lui-même.
- Gateway : Adresse IP de l'automate programmable communiqué avec L'IHM par la porte Ethernet.

1.5. Compilation et chargement de programme

Une fois l'édition de la configuration terminée, vous devez cliquer sur le bouton  puis cliquez sur  Pendant la compilation, la fenêtre de sortie d'informations affichera la situation en temps réel, Il ne peut pas terminer la compilation si une erreur existe. Il est nécessaire de compiler avant de télécharger, alors que l'étape de conservation est nécessaire avant de compiler. Après pour charger le programme vous devez cliquer sur le bouton .

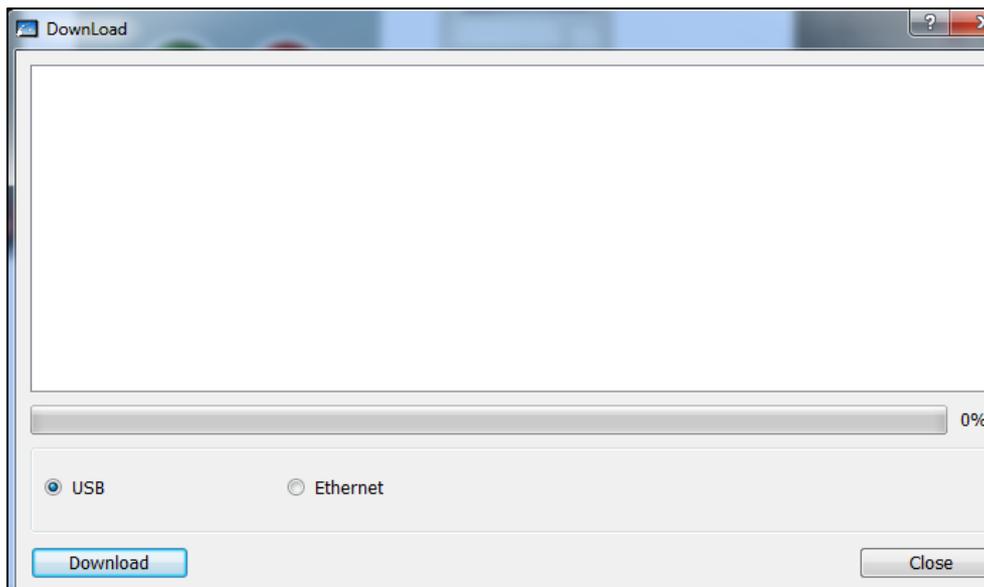


Figure 5 : Boîte de dialogue de téléchargement

Enfin, exécutez la commande USB dans le menu Télécharger pour télécharger via un câble USB. Cliquez sur télécharger; la barre de progression du milieu montre la progression du téléchargement. Après le téléchargement, une fenêtre contextuelle montre l'achèvement du téléchargement sur USB. Cliquez sur OK pour terminer la tâche de téléchargement. Comme le montre dans la figure 6.

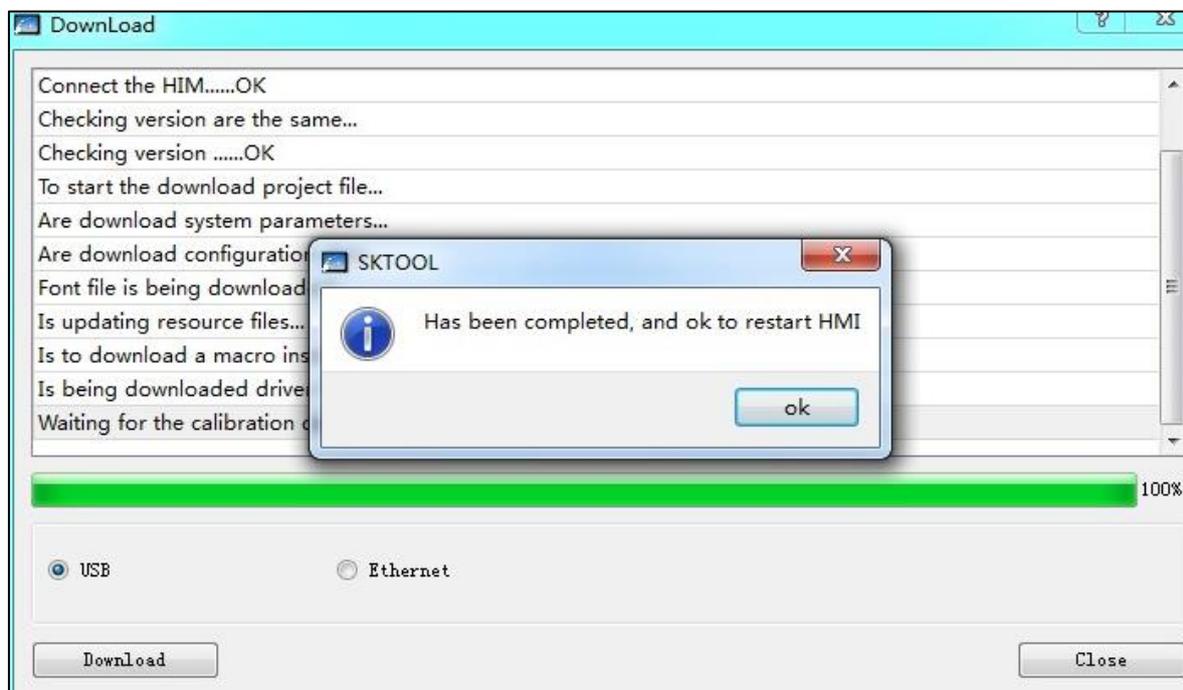


Figure 6 : Téléchargement terminé

L'étape suivante est la connexion à l'automate pour voir si le projet de configuration s'exécute correctement dans l'IHM.

Bibliographie

- [1] M .L . FAS, " cours actionneur ", support de cours, université Saad Dahleb de Blida 1, 2017.
- [2] W. Boudjtat, "Etude et simulation d'une machine asynchrone alimente par un onduleur de tension a commande MLI", Mémoire d'obtention du diplôme de magister, université Badji-Mokhtar Annaba, 2017.
- [3] P. LE BRUN, "Cours machine asynchrone", support de cours, Lycée Louis ARMAND, Strasbourg.
- [4] E. GAUCHERON, " Les moteurs électriques ... pour mieux les piloter et les protéger", Collection technique Cahier technique n° 207, Schneider Electric, 2004.
- [5] A. TITAOUINE, " Etude comparative des différents types de bobinage d'un moteur asynchrone triphasé à cage", Mémoire d'obtention du diplôme MASTER, Université Mohamed Khider Biskra, 2018.
- [6] G. Chateigner, M. Boës, D. Bouix, J. Vaillant, D. Verkindère, "Manuel de génie électrique ", Livre DUNOD, 2005.
- [7] Z. DEMECH, B. MOKHTARI, "Modélisation et simulation d'une génératrice asynchrone en saturation", Thèse de master en génie électrique, université Kasdi merbah OUARGLA, 2012.
- [8] O. HAMMACHI, S. CHAABANA," Etude expérimentale de quelques variateurs de vitesse d'un moteur asynchrone", Mémoire d'obtention du diplôme MASTER 2 en électrotechnique, Université Bejaia Abderrahmane Mira de Bejaia, 2018.
- [9] MAS - le Moteur Asynchrone.odt, version du 15.04. 2012.
- [10] H. BOUDJELLA, Z. IDER, "Étude et simulation d'un variateur de vitesse commande un moteur à courant continu", Mémoire d'obtention du diplôme MASTER 2 Electrotechnique Industrielle, université Kasdi Merbah OUARGLA, 2016.
- [11] N. ZENATI, A. AIT AMRANE, "Réalisation d'un banc d'essai de la chaine d'action « variateur de vitesse – moteur asynchrone » commandé par un API et supervisé par un écran tactile", Mémoire d'obtention du diplôme MASTER 2 Automatique des systèmes de production, Université Djilali Bounâama - Khemis Miliana, 2016.
- [12] L. HAMACHE et M. MEBARKI, "Étude d'un moteur asynchrone alimenté par un variateur de vitesse", Mémoire de fin d'études d'ingénieur en électrotechnique. Université A. MIRA Bejaïa. 2004.

- [13] F. Bouabida, A. Benkhelifa, "Migration d'un variateur de vitesse (SEW) De l'ancienne version vers la nouvelle", Mémoire d'obtention du diplôme MASTER 2 Automatique, Université Abderrahmane Mira de Bejaïa, 2017.
- [14] Industry Support Siemens, "Variateur SINAMICS V20", Instructions de service, DATASheet, 2013.
- [15] C. Palerme, "Le moteur asynchrone triphasé", Cours d'Electricité 2 - Électrotechnique, 2009.
- [16] D. Jacob, "Moteurs à courant alternatif : Moteurs asynchrones et brushless", Cours et problèmes, Livre TECHNOSUP, Amazon, 2005.
- [17] Mr. YOUSFI Lounis, MELLALI Sofiane, " Etude de l'automatisation et de la supervision d'un procédé de lavage de filtres Niagara à CEVITAL - TIA PORTAL V12 -", Mémoire d'obtention du diplôme MASTER 2 Automatique, 2017.
- [18] Dr k.labadi" introduction a la commande des systèmes ", support de cours, grand école d'ingénieurs EPMI, 2009.
- [19] Mr. AYAD Hocine," cours automate programmable ", support de cours, université Saad Dahleb de Blida 1, 2017.
- [20] Mr. SLIM BEN SAOUD, "cours automatismes", support de cours, institut national des sciences appliquées et de technologie tunis , 2015.
- [21] Industry Support Siemens, "automate programmable s7-1200", Manuel système, DATASheet, 2014.
- [22] Industry Support Siemens," Alimentation SITOP ", SITOP avec design SIMATICdisgn, DATASheet, 2015.
- [23] Industry Support SAMKOON," SKTOOL HELP "shenzhen Samkoon technology cororation lts, DATASheet, 2018.
- [24] Industry Support LPD3806-400 BM, " Codeur rotatif biphasé de type incrémental", Instructions de service, DATASheet, 2015.