REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Département Génie Électrique et Informatique

Mémoire de fin d'études de Master

Filière : Génie électrique
Spécialité : Automatique et systèmes

Commande d'un moteur asynchrone avec un automate programmable via un variateur de vitesse

---

Présenté par :
HENTABLI Mohamed
RAMOUL ABDERREZAK

Proposé et Dirigé par :
Dr BETTACHE Farouk
Dr BOUHEDDA Mounir

Promotion 2019

ملخص:

يتمثل هدفنا في تحقيق ناقل بسيط ثنائي الاتجاه يمكن التحكم فيه يدويًا إما عن طريق أزرار الضغط أو واجهة HMI (واجهة الجهاز البشري) باستخدام جهاز تحكم منطقي قابل للبرمجة(PLC) تسمى لغة البرمجة المستخدمة Ladder على البرنامج TIA portal V13، بالإضافة إلى ذلك ، يمكن تغيير سرعة الناقل (محرك ثلاثي الطور) من خلال محرك متغير السرعة في أي وقت.

كلمات مفتاحية

وحدات تحكم المنطق القابلة للبرمجة، متغير السرعة، محرك غير متزامن، شاشة لمسية.

Résumé :

Notre travail consiste à réaliser un simple convoyeur double sens qui peut être commande manuellement soit par des bouton poussoir, ou soit par une interface HMI (Human machine interface) en utilisant un automate programmable API.

Le langage de programmation utiliser est le ladder sous le logiciel TIA portal V13,

En outre la vitesse du convoyeur (moteur asynchrone triphasée) peut être changer à travers un variateur de vitesse à n'importe quel moment.

Mots clés : Automate programmable, Variateur de vitesse, TIA portal, moteur asynchrone, Ladder, Convoyeur.

Abstract:

Our work is to realize a simple two-way conveyor that can be manually controlled either by push buttons or by HMI interface (Human machine interface) using a programmable logic controller (PLC).

The programming language used is called Ladder on the software TIA portal V13

In addition, the speed of the conveyor (Three phase motor) can be changed through a variable speed drive at any time

Key words: Programable logic controllers, Speed variator, Asynchronous motor, HMI, TIA portal.

TABLE DES MATIERES

Liste des figures I
LISTE des Tableaux IV
Liste des acronymes V
INTRODUCTION GENERALE 2
Chapitre I : Automate Programmable 3
1. Introduction 4
2. Historique sur les automates programmables industriel 4
3. Principe de fonctionnement d'un API 4
4. Architecture industrielle 5
5. Domaines d'utilisation des API 6
6. Les avantage d'API 6
7. AUTOMATE PROGRAMABLE S7 300 7
7.1. Caractéristique du l'automate programmable s7 300 7
8. Composants d'un S7-300 8
8.1. Alimentation 8
8.2. CPU 9
8.3. Connexion Multi Point Interface MPI 11
8.4. Modules d'entrées/sortie 11
9. TIA PORTAL 13 13
9.1. Vue du portail et vue du projet 13
9.2. Vue du portail 14
9.3. Vue du projet 14
9.4. Création d'un projet 15
9.5. Configuration et paramétrage du matériel 16
9.6. Compilation et chargement de la configuration matérielle 17
9.7. Liaison avec l'automate 17
9.8. Interface PG/PC 18
9.9. Compilation 19
9.10. Création d'une table de variables 21
9.11. Types de variables 21
9.12. Blocs d'organisation 22
9.13. Fonctions et blocs fonctionnels 24
9.14. Bloc de données 24
9.15. Le simulateur des programmes PLCSIM : 26

9.15.1. Exécution du programme 27
9.16. SIMATIC WinCC 29
9.17. Interface Homme-Machine (IHM) 29
9.17.3. Programmation des Boutons et des voyants 31
10. Conclusion 31
Chapitre II : Moteur asynchrone triphasé 33
1. Introduction 34
2. Historique Sur Les Machines Asynchrones : 34
3. Présentation 35
4. Constitution de la machine asynchrone : 35
4.1. Le stator 36
4.2. Le Rotor 36
5. Symboles 38
6. Principe de fonctionnement du moteur à induction 39
7. Le glissement 40
8. Vitesse de synchronisme 41
8.1. Action sur le nombre de pairs de pôles 41
8.2. Action sur le glissement 43
8.3. Action sur la fréquence 44
9. Couplage de la plaque à bornes 44
10. Branchement du moteur sur le réseau triphasé 44
11. La plaque signalétique 45
12. Les avantages et les inconvénients du moteur asynchrone : 46
12.1. Les avantages du moteur asynchrone 46
12.2. Les inconvénients du moteur asynchrone 46
13. Les variateurs de vitesse 47
14. Constitution et principe de fonctionnement d'un variateur de vitesse 47
14.1. Interface utilisateur 47
14.2. Redresseur 48
14.3. Filtre 49
14.4. L'onduleur 50
15. Avantages et inconvénients d'un variateur de vitesse 52
15.1. Avantages 52
15.2. Inconvénients 53
16. Choix d'un variateur de vitesse 53

17. Altivar12HU15M2 : 54
17.1. Schéma de câblage général....55
17.2. Bornes de contrôle....57
18. Conclusion....58
Chapitre III : realisation....59
1. INTRODUCTION....60
2. Schéma structurelle di convoyeur 60
3. Configuration des éléments utilisée 61
3.1. Réglages usine du variateur 61
3.2. Réglage des paramètres du moteur....61
3.3. Définition des paramètres de base 61
3.4. Définition des paramètres de contrôle....62
3.5. Câblage de la commande choix de la configuration 62
.4 Moteur Asynchrone 63
4.1. Plaque signalétique du moteur utilisé 63
5. Les Boutons poussoirs et les voyants 64
5.1. Les Boutons poussoirs....64
5.2. Symbolisation....64
5.3. Fonctionnement....65
5.4. Les Boutons coup de poing 66
5.5. Les voyants....67
6. HMI TP 177B 68
6.1. Présentation générale du produit :....68
6.2. Structure du pupitre opérateur TP 177B 6"....68
6.3. Vue de dessous 69
6.4. Vue arrière 70
6.5. Raccordement alimentation 70
6.6. Mettre en marche et tester le pupitre opérateur 71
7. Raccordements des différents éléments 73
7.1. Raccordement du variateur de vitesse avec l'API....73
7.2. Raccordement du variateur de vitesse avec le moteur asynchrone....74
8. Pilotage du variateur de vitesse ATV312 avec l'API....74
8.1. Explication des variables utiliser avec leur adresse 76
8.2. Explication de chaque réseau 77
9. Supervision....78

9.1. Le menu Start 78
9.2. Le menu Main 78
10. Conclusion 79
CONCLUSION GENERALE 80
BIBLIOGRAPHIE 81

LISTE DES FIGURES

Figure 1 structure interne d'un automate programmable 5
Figure 2 Automate compact Allen-Bradley 5
Figure 3 Automate Modulaire 6
Figure 4 automate programmable s7 300 7
Figure 5 Caractéristique technique S7 300 CPU 313 8
Figure 6 Composant S7 300 8
Figure 7 Alimentation de l'automate S7 300 9
Figure 8 CPU 313c 10
Figure 9 Raccordement du CPU 10
Figure 10 câble MPI 11
Figure 11 Adaptateur MPI - USB 11
Figure 12 E/S et adresses 12
Figure 13 Bus d'adresses 13
Figure 14 Vue portail 14
Figure 15 Vue Projet 14
Figure 16 Création d'un projet 15
Figure 17 l'ajout d'un appareil 16
Figure 18 Vue d'ensemble des appareils 17
Figure 19 En ligne et diagnostic 18
Figure 20 Interface PG/PC 18
Figure 21 Compilation et chargement 19
Figure 22 Choix de connexion 20
Figure 23 Chargement d'un programme 20
Figure 24 Table des variables 21
Figure 25 Langage de programmation des blocs 25
Figure 26 Fenêtre de configuration du simulateur 28
Figure 27 Création d'un HMI 30
Figure 28 Cable Profibus Figure 29 Cable Ethernet 30
Figure 30 Liaison entre IHM et API 31
Figure 31 : Programmation bouton et voyants 31

Figure 32 Moteurs asynchrones 35
Figure 33 Stator d’une machine asynchrone 36
Figure 34 Principe d’un rotor bobine 37
Figure 35 Principe d’une cage d’écureuil 38
Figure 36 Symboles d’une machine asynchrone 38
Figure 37 Principe de base d'un moteur 39
Figure 38 Principe de fonctionnement d’un moteur à induction 40
Figure 39 Bobinage en parallèle 42
Figure 40 Bobinage en série 42
Figure 41 Enroulements séparés 42
Figure 42 La plaque à bornes 44
Figure 43 Couplage des moteurs 45
Figure 44 Structure d’un variateur de vitesse 130 47
Figure 45 Redresseur triphasé à diodes 48
Figure 46 Tension redressée 49
Figure 47 Schéma du principe de filtrage 49
Figure 48 Schéma de principe de l'onduleur 50
Figure 49 Principe de fonctionement 52
Figure 50 Commande MLI 52
Figure 51 Altivar12HU15M2 54
Figure 52 Schéma de câblage général 55
Figure 53 interface utilisateur 56
Figure 54 interface utilisateur 57
Figure 55 schéma des entrées/sorties 57
Figure 56 Schéma structurelle 60
Figure 57 configuration 61
Figure 58 Paramètre de base 61
Figure 59 paramètre de contrôle 62
Figure 60 Câblage de potentiomètre 62
Figure 61 schéma de câblage de control 2 fils 62
Figure 62 schéma de câblage de control 3 fils 63
Figure 64 Bouton poussoir 64
Figure 65 symbole de bouton poussoir NC 65

Figure 66 Symbole bouton poussoir normalement ouvert 65
Figure 67 Fonctionnement NO.....65
Figure 68 Fonctionnement NC.....66
Figure 69 Bouton coup de poing.....66
Figure 70 Schéma d'un bouton coup de poing 66
Figure 71 Fonctionnement d'un boutons coup de poing 67
Figure 72 Voyant 24V.....67
Figure 73 Symbole d'un voyant.....67
Figure 74 Fonctionnement d'un voyant.....68
Figure 75 Vue de face et vue de profile HMI177b 69
Figure 76 Vue de dessous HMI177b.....69
Figure 77 Vue arrière HMI177b 70
Figure 78 schéma du câblage électrique 71
Figure 79 Menu Loader.....71
Figure 80 Menu Transfer 72
Figure 81 Schéma du raccordement TOR ATV12/API 73
Figure 82 Schéma du raccordement analogique ATV12/API.....73
Figure 83 Schéma du câblage moteur asynchrone avec ATV12 74
Figure 84 Le code ladder utiliser pour le control du moteur.....76
Figure 85 Le menu start 78
Figure 86 Le menu main 79

LISTE DES TABLEAUX

Tableau 1 Plaque signalétique.....64
Tableau 2 Liste des variables 77

LISTE DES ACRONYMES

API : Automate Programmable Industriel

PLC: Programmable Logic controller

TIA: Totally Integrated Automation

HMI: Human Machine Interface

IHM: Interface Homme Machine

CPU: Central Processing Unit

TOR: Tout Ou Rien

IP: Internet Protocol

PN/IE : Profinet/Industrial Ethernet

E/S : Entrées/Sorties

MPI: Message Passing Interface

ROM : Read Only Memory

PROM : Programmable ROM

EEPROM : Electrically Erasable

PROM RAM : Random Access Memory

DI : Digital Input

DO : Digital Output

AI : Analogique Input

AO : Analogique Output

INTRODUCTION GENERALE

Un convoyeur est un moyen efficace de circulation des matériaux. Ce système est fréquemment utilisé dans le transport de matériaux entre les points spécifiques sur un chemin fixe. Il Ya une grande variété de convoyeurs (par exemple, convoyeur à chaîne, a rouleaux, a bande etc.…), qui peut différent en fonction des caractéristiques du matériau à déplacé, ainsi que le déplacement à effectuer.

Notre problématique est premièrement se situe au niveau de câblage, pour contrôler un convoyeur pareil on peut avoir une saturation un encombre dans l'armoire si on utilise la logique câblée, et comme on peut avoir des problèmes de control au niveau de moteur asynchrone ou ça sera faite manuellement par l'utilisateur.

L'objectif principal de notre travail est de réaliser et commander un moteur asynchrone par un automate programmable industriel. Comme on va aussi le supervisé avec un écran tactile, des boutons poussoir et des voyant. Et même varier sa vitesse en utilisent un variateur de vitesse.

Pour atteindre notre objectif, nous avons organisé notre mémoire en 3 chapitres :

-Le premier chapitre consacré pour la présentation des automates programmable et le logiciel TIA Portal et son langage de programmation.

-Dans le deuxième chapitre une brève introduction sur les, moteurs asynchrones, nous donnons leur constitution et leur principe de fonctionnement suivi de leurs caractéristiques mécaniques et électriques. Et l'étude de la variation de vitesse d'un moteur asynchrone.

Puis nous décrivons les variateurs de vitesses tout en expliquant leur fonctionnement.

Ensuite nous donnons les avantages et les inconvénients de ces derniers.

Enfin, nous donnons une expertise sur le choix de variateur de fréquence.

-Dans le troisième chapitre, qui est la partie réalisation de notre projet. On présente la structure globale et le raccordement entre les parties du convoyeur, Puis la configuration et le câblage utilisé pour la commande et la supervision de notre système. Ensuite on explique les différentes étapes suivis pour la commande du moteur asynchrone avec l'automate programmable. Et on termine avec les différents écrans qu'on a créés pour la supervision du système. Enfin, nous clôturons notre document avec une conclusion générale qui est dédiée à une synthèse du travail présenté.

CHAPITRE I : AUTOMATE PROGRAMMABLE

1. Introduction

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

Il peut être directement connecté aux capteurs et pré actionneurs grâce à ses entrées/ sortie industrielles.

Il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, microcoupures de la tension d'alimentation, parasites, etc.).

Enfin, sa programmation à partir des langages spécialement développés pour le traitement des fonctions d'automatismes facilitent son exploitation et sa mise en œuvre.

2. Historique sur les automates programmables industriel

Les automates programmables industriels (API) sont apparus aux U.S.A en 1969, dans le secteur de l'industrie automobile, ils furent utilisés en Europe environ deux ans plus tard.

Sa date de création coïncide donc avec le début de l'ère du microprocesseur et avec la généralisation de la logique câblée modulaire.

L'API est la première machine a langage c'est-à-dire un des calculateurs logiques dont le jeu d'instruction est orienté vers les problèmes de logique et des systèmes à évolution séquentielles [1].

3. Principe de fonctionnement d'un API

L'automate programmable reçoit des données par ses entrées, celles-ci sont ensuite traitées par un programme défini, le résultat obtenu étant délivré par ses sorties, Figure 1. Ce cycle de traitement est toujours le même, quel que soit le programme, néanmoins le temps d'un cycle d'API varie selon la taille du programme et la puissance de l'automate. C'est l'unité centrale qui gère l'automate programmable : elle reçoit, mémorise et traite les données entrantes et détermine l'état des données sortantes en fonction du programme établi [2].

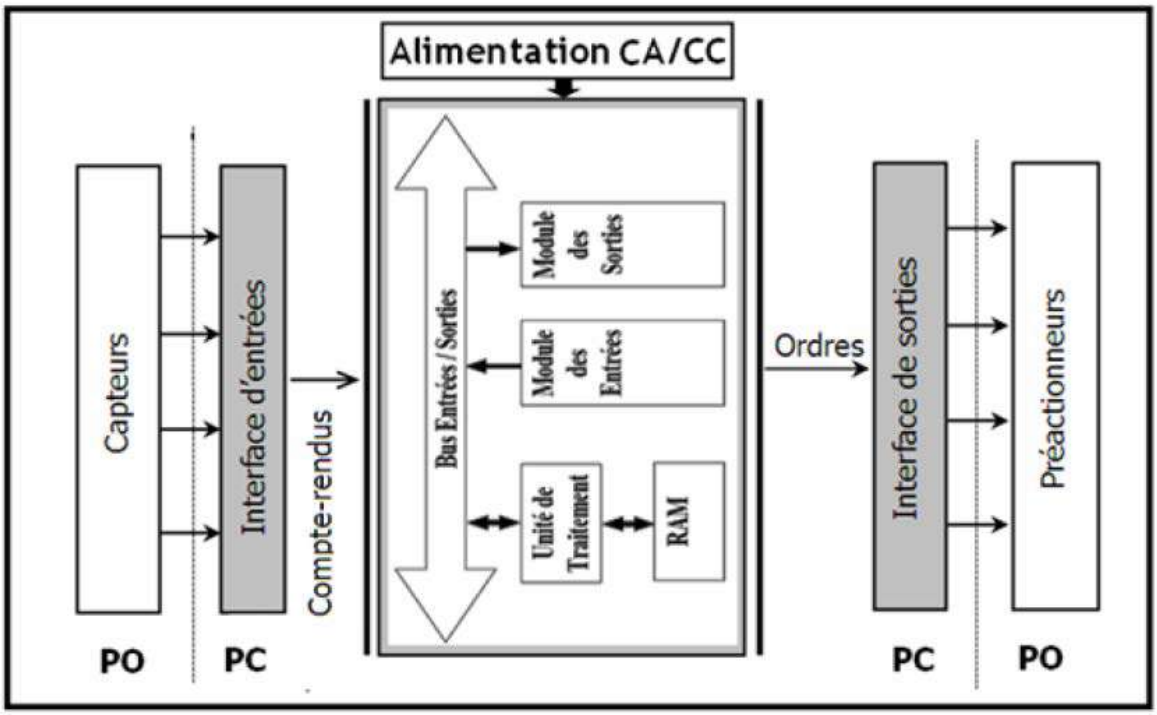


Figure 1 structure interne d'un automate programmable

4. Architecture industrielle

Les automates peuvent être de type compact ou modulaire.

- Les automates programmables de types compacts se présentent comme des modules ou l'ensemble des fonctions à savoir alimentation, CPU, entée et sortie sont sur le même boitier par exemple l'automate Allen-Bradley [3], Figure 2

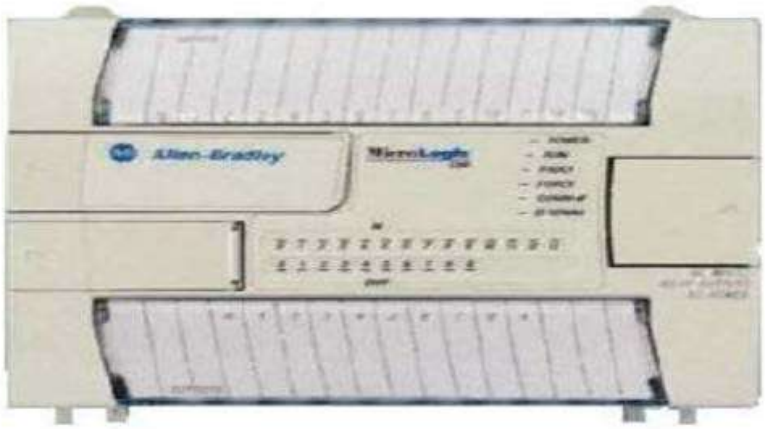


Figure 2 Automate compact Allen-Bradley

- Les automates programmables modulaires présentent des alimentations, des CPU, des modules entrées et sorties dans des boitiers indépendants les uns des autres et relier entre eux par bus ou fond de panier par exemple l'automate Siemens [3], Figure 3.



Figure 3 Automate Modulaire

5. Domaines d'utilisation des API

On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage.) ou des chaines de production (automobile, agroalimentaire ...) ou il peut également assurer des fonctions de régulation de processus (métallurgie, chimie ...).

Ils sont de plus en plus utilisés dans le domaine du bâtiment (tertiaire et industriel) pour le contrôle du chauffage, de L'éclairage, de la sécurité ou des alarmes [3] Aspect extérieur d'un API

6. Les avantage d'API

- La capacité de production accélérée
- L'aptitude à convenir à tous les milieux de production.
- La souplesse d'utilisation.

7. AUTOMATE PROGRAMABLE S7 300

On a choisie l'API S7-300 CPU313C. L'automate choisi, figure, se caractérise par une mémoire de 64Ko, 24 entrées/16 sorties TOR, 5 entrées/2sorties analogique, et une interface de communication Profibus-DP (Périphériques Décentralisés).



Figure 4 automate programmable s7 300

7.1. Caractéristique du l'automate programmable s7 300

La Figure 5 contient quelques caractéristiques Techniques du Siemens S7- 300- CPU 313C [4]

Mémoire de travail intégrée	64 kOctet
Nombre des entrées et sorties	Entrées TOR : 24, Sorties TOR : 16, Entrées Analogiques : 4 (mesure de tension ou de courant) et 1 (mesure de résistance ou température) ; Sorties analogiques : 2
Port de communication MPI	Oui
Nombre de temporisation	256
Régulation PID	Oui
Communication Profibus	Oui

Figure 5 Caractéristique technique S7 300 CPU 313

8. Composants d'un S7-300

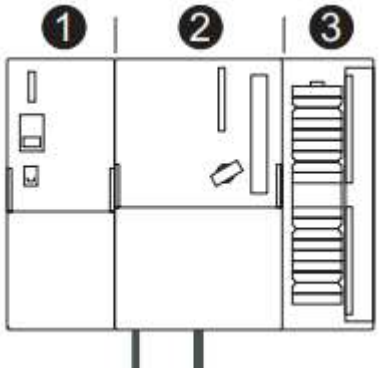


Figure 6 Composant S7 300

1 Alimentation
2 Module CPU
3 Module entrée/sortie

8.1. Alimentation

Elle convertit la tension réseau (AC 120/230 V) en tension de service DC 24 V et assure l'alimentation du S7-300 ainsi que l'alimentation externe pour les circuits de charge DC 24 V.

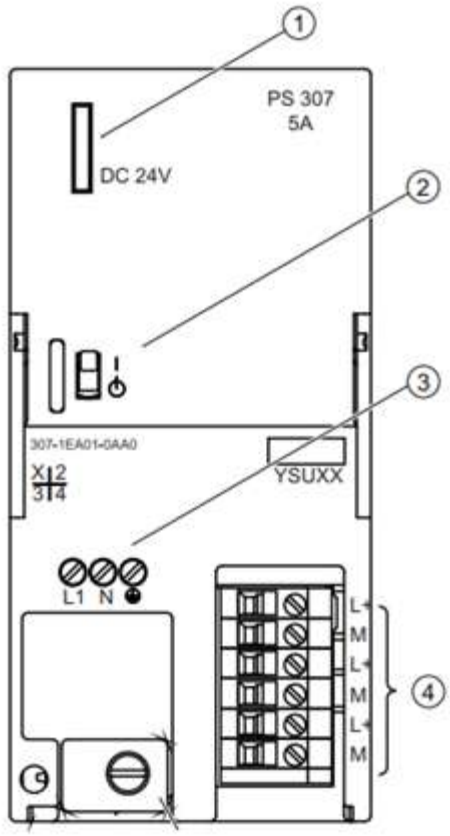


Figure 7 Alimentation de l'automate S7 300

1 LED Display de mise en marche

2 ON/OFF switch

3 Entrée d'alimentation 24V

4 Sortie 24V

8.2. CPU

Son rôle principal est d'exécuté le programme d'utilisateur, alimente le bus interne du S7-300 en Communique avec les autres partenaires d'un réseau MPI via l'interface MPI.



Figure 8 CPU 313c

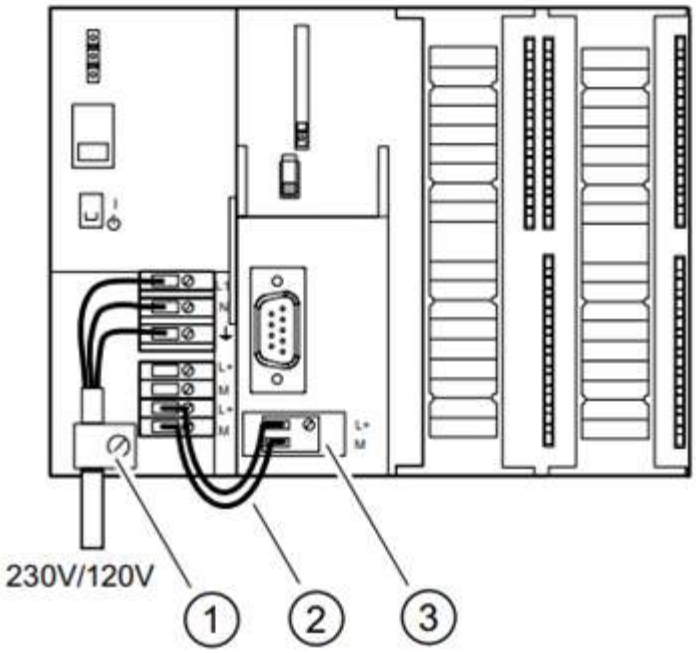


Figure 9 Raccordement du CPU

1. Collier d'arrêt de traction de l'alimentation
2. Câbles de liaison entre PS et CPU
3. Raccordement enfichable de l'alimentation
4. Entrée MPI
5. Switch ON/OFF
6. Entrée pour carte mémoire

8.3. Connexion Multi Point Interface MPI

Le Multi Point Interface est une interface propriétaire des automates programmable industriel SIMATIC S7 de la compagnie SEIMENS. Elle est utilisée pour le raccordement des stations de programmation (PC), et les pupitres opérateurs, ainsi que pour d'autres appareils appartenant à la famille SIMATIC.



Figure 10 câble MPI

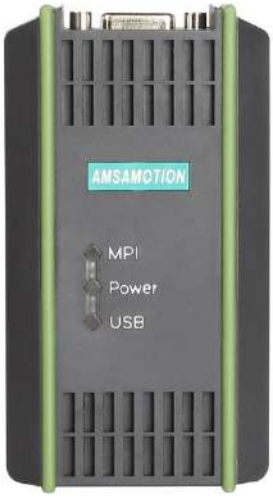


Figure 11 Adaptateur MPI - USB

8.4. Modules d'entrées/sortie

Ils adaptent les différents niveaux de signaux des signaux de processus au S7-300. [5]

Les entrées et les sorties intégrées de cette CPU comportent les adresses suivantes sur la Figure 12

Entrées/sorties	Adresses par défaut
24 entrées TOR	124.0 à 126.7 dont 12 entrées pour les fonctions technologiques : 124.0 à 125.0 125.4 à 125.6
16 sorties TOR	124.0 à 125.7 dont 3 sorties pour les fonctions technologiques : 124.0 à 124.2
4+1 entrées analogiques	752 à 761
2 sorties analogiques	752 à 755

Figure 12 E/S et adresses

L'adresse d'une entrée ou d'une sortie d'un module TOR est composée de l'adresse d'octet et de l'adresse de bit

Exemple : E 1.2 L'exemple se compose des éléments suivants :

- Entrée E
- adresse d'octet 1 et
- adresse de bit 2

L'adresse d'octet est basée sur l'adresse initiale des modules. L'adresse de bit est indiquée sur le module. Si le premier module TOR est placé sur l'emplacement 4, il a l'adresse de début par défaut 0. L'adresse de début de chaque modules TOR suivant augmente de 4 par emplacement. La figure suivante présente le schéma selon lequel sont créées les adresses des différentes voies du module TOR.

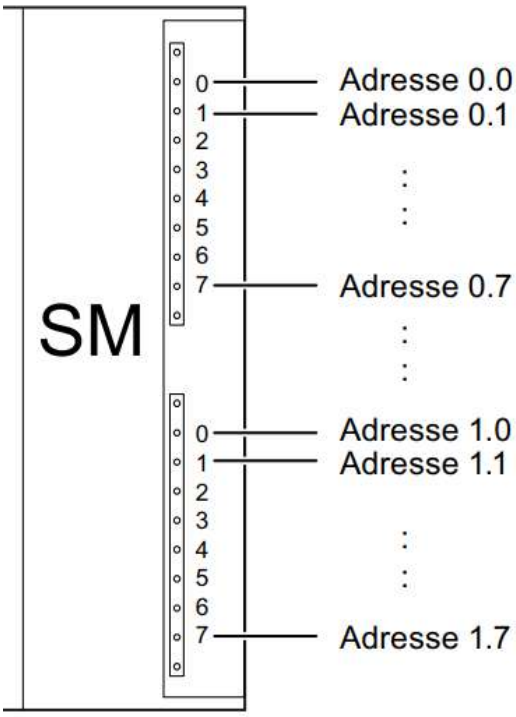


Figure 13 Bus d'adresses

9. TIA PORTAL 13

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

Ce logiciel permet la communication avec l'automate programmable (Software/Hardware)

9.1. Vue du portail et vue du projet

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

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

9.2. Vue du portail

Chaque portail permet de traiter une catégorie de tâche (actions). La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée.

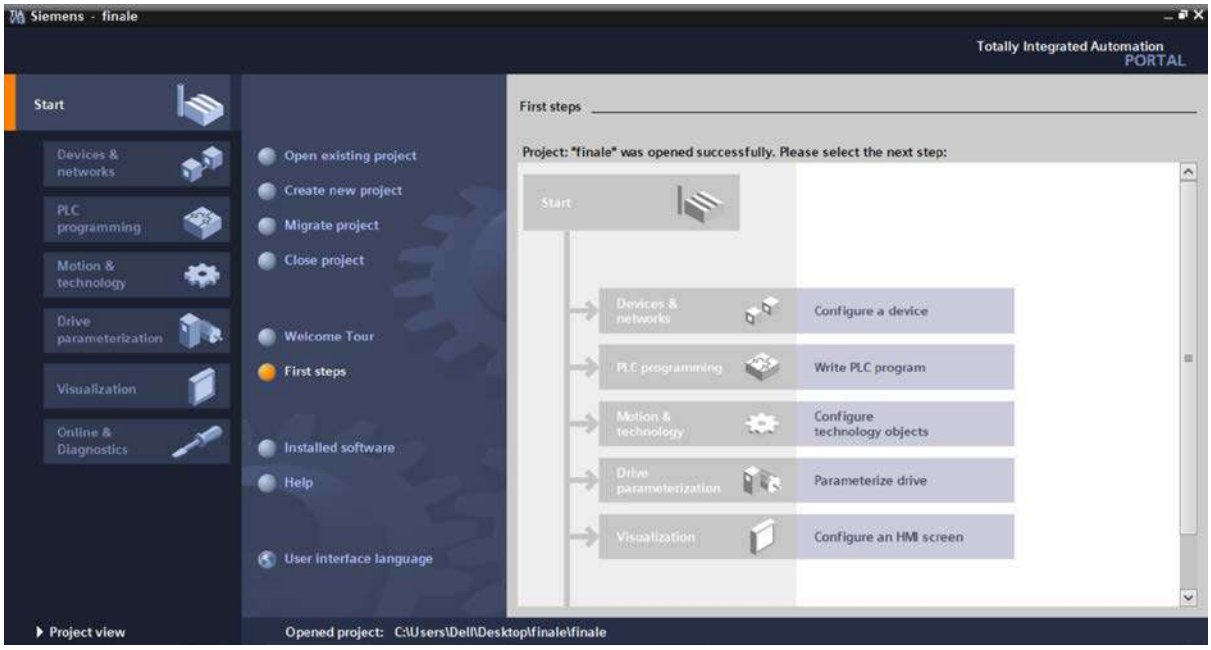


Figure 14 Vue portail

9.3. Vue du projet

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

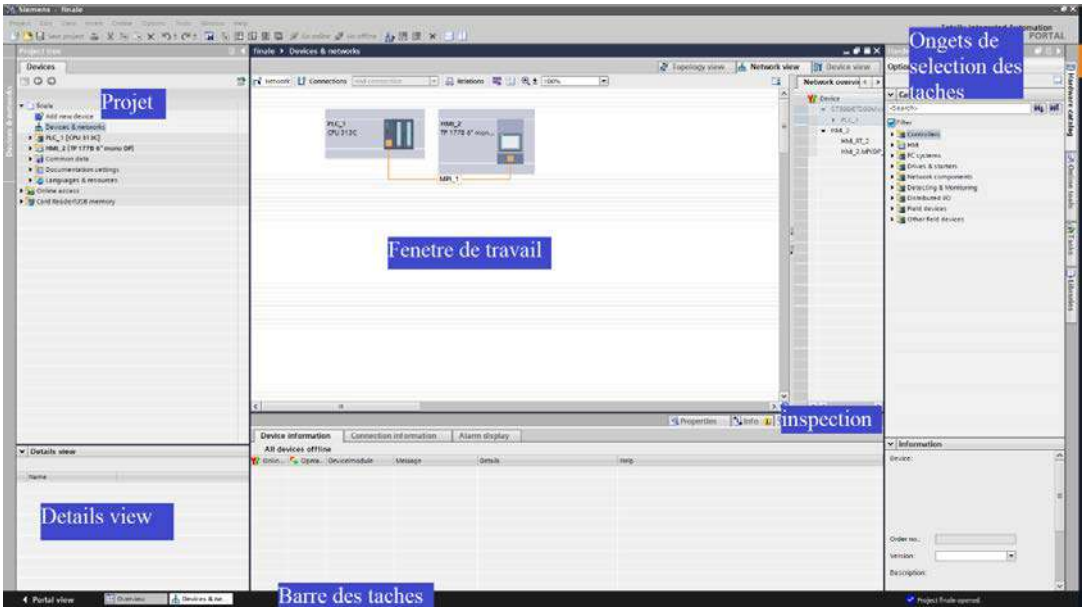


Figure 15 Vue Projet

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

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

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

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

Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres.

9.4. Création d'un projet

Pour créer un projet dans la vue du portail, il faut sélectionner l'action ***« Créer un projet».***

On peut donner un nom au projet, choisir un chemin ou il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet.

Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton ***« créer »***

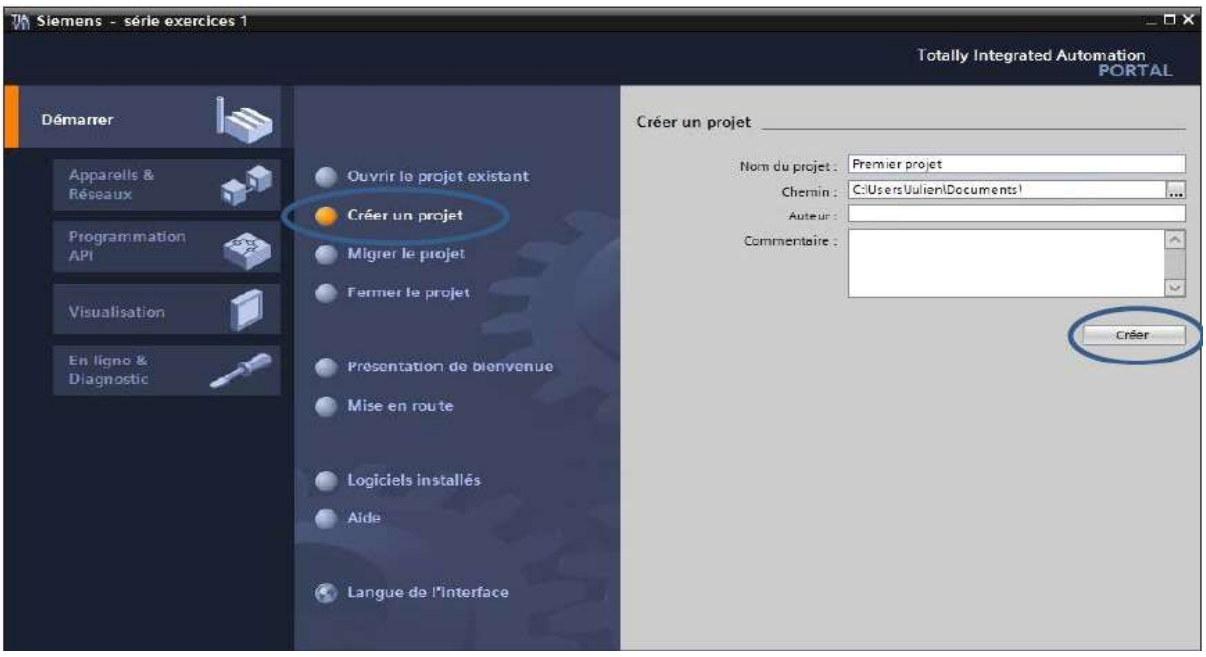


Figure 16 Création d'un projet

9.5. Configuration et paramétrage du matériel

Une fois votre projet crée, on peut configurer la station de travail.

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

La liste des éléments que l'on peut ajouter apparait (API, HMI). On commencera par faire le choix de notre CPU pour ensuite venir ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication.).

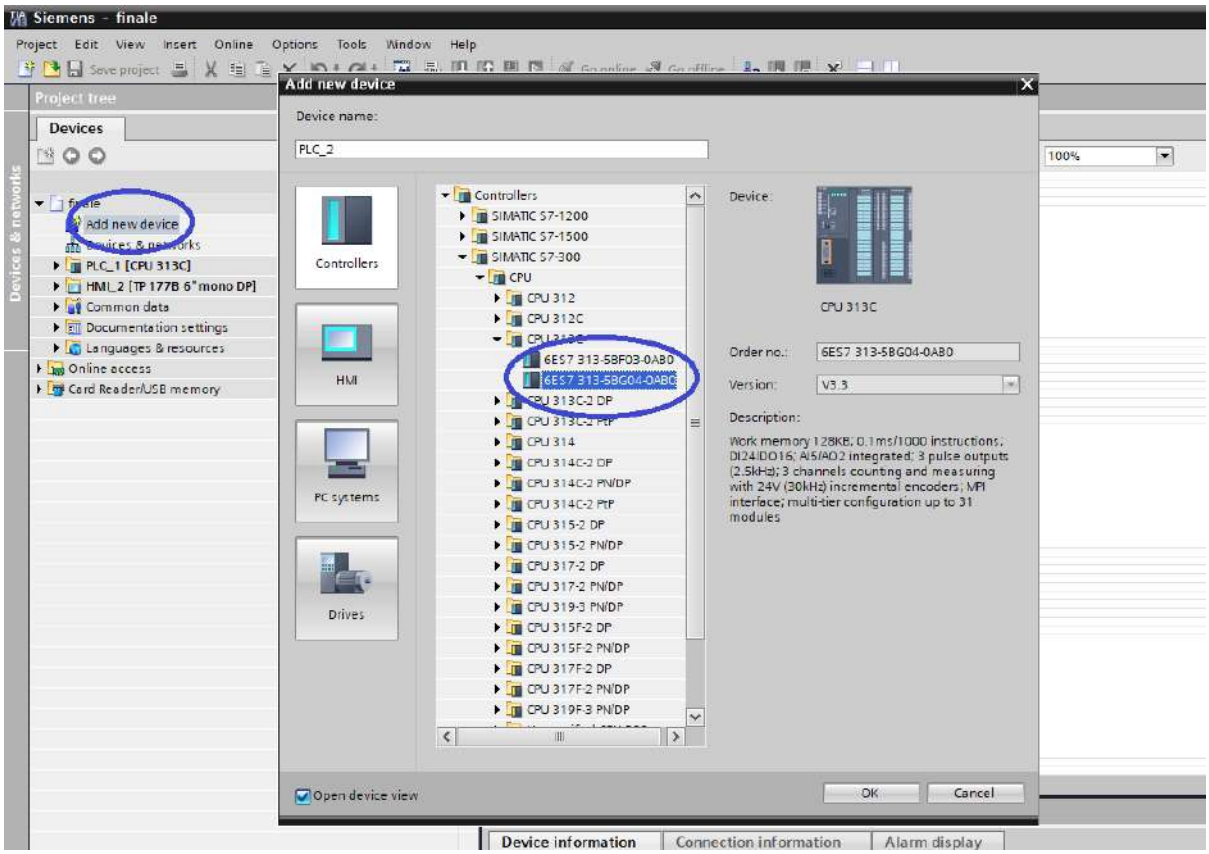


Figure 17 l'ajout d'un appareil

Les modules complémentaires de l'API peuvent être ajoutés en utilisant le catalogue. Si l'on veut ajouter un écran ou un autre API, il faut repasser par la commande « ajouter un appareil » dans le navigateur du projet. [7]

Lorsque l'on sélectionne un élément à insérer dans le projet, une description est proposée dans l'onglet information.

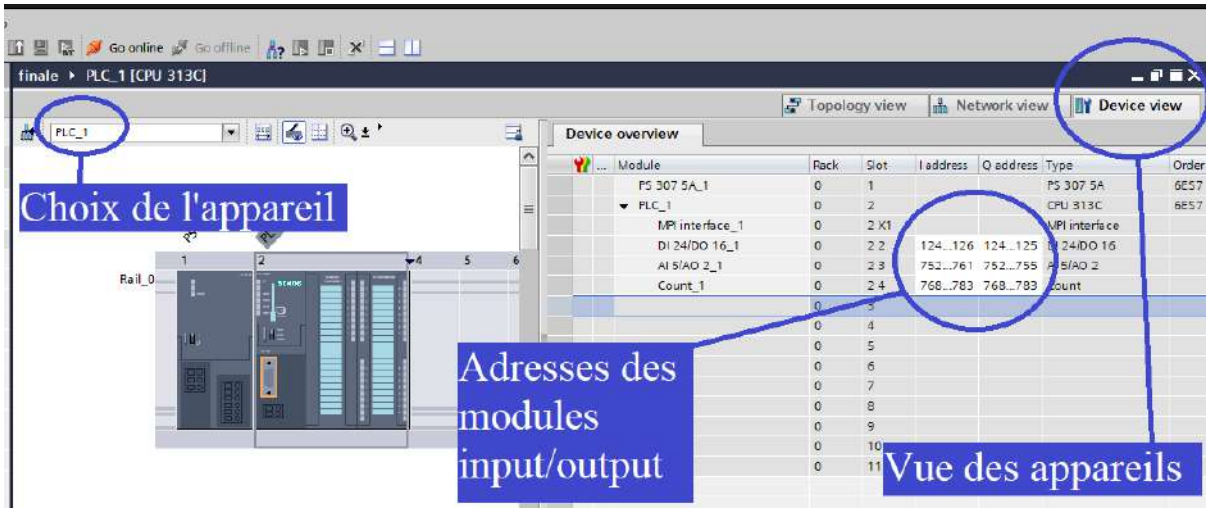


Figure 18 Vue d'ensemble des appareils

On sélectionne la CPU puis à l'aide des deux petites flèches (voir figure), on fait apparaitre l'onglet ***« Vue d'ensemble des appareils***.

Les adresses des entrées et sorties apparaisse. Vous pouvez les modifier en entrant une nouvelle valeur dans la case correspondante.

9.6. Compilation et chargement de la configuration matérielle

9.7. Liaison avec l'automate

Il faut maintenant charger la configuration de l'automate dans celui-ci. Pour cela, il faut tout d'abord connecter l'automate au PC en utilisant l'interface Simatic S7 PC USB adapter. Ensuite, après avoir sélectionné la vue « En ligne et diagnostique », sélectionnez les options suivantes :

- Mode : MPI
- Interface PG/PC : Pc Adapter Il ne reste plus qu'à cliquer sur le bouton « Liaison en ligne » pour vous connecter à l'automate.

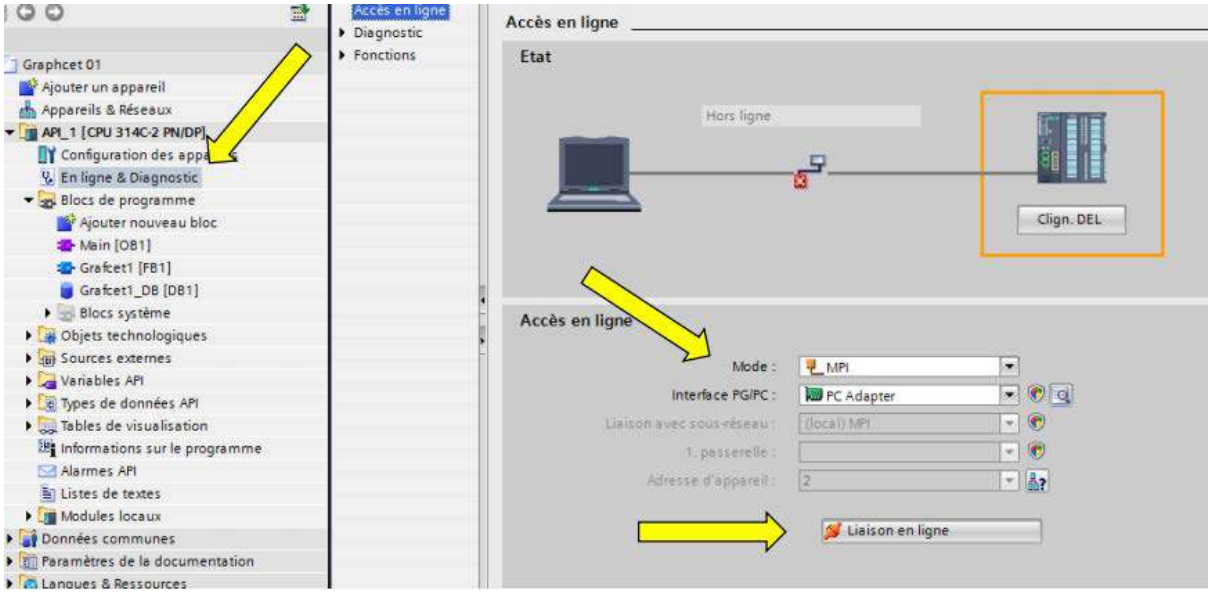


Figure 19 En ligne et diagnostic

9.8. Interface PG/PC

Un PG/PC est le dispositif de programmation qui est utilisé pour programmer les automates, les contrôleurs et les panels opérateur Siemens. Ce sont en général des ordinateurs portables très puissants avec les logiciels de mise en service (Step7,TIA Portal,Wincc etc...) installés et possédant les ports de communication nécessaires pour se connecter à des automates et autres équipements industriels. [3]



Figure 20 Interface PG/PC

9.9. Compilation

Une fois la configuration matérielle 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 cliquer sur l'icône *« compiler »*.

En utilisant cette manière, on effectue une compilation matérielle et logicielle.

Une autre solution pour compiler est de faire un clic droit sur l'API dans la fenêtre du projet et de choisir l'option « Compiler ➔Configuration matérielle ».

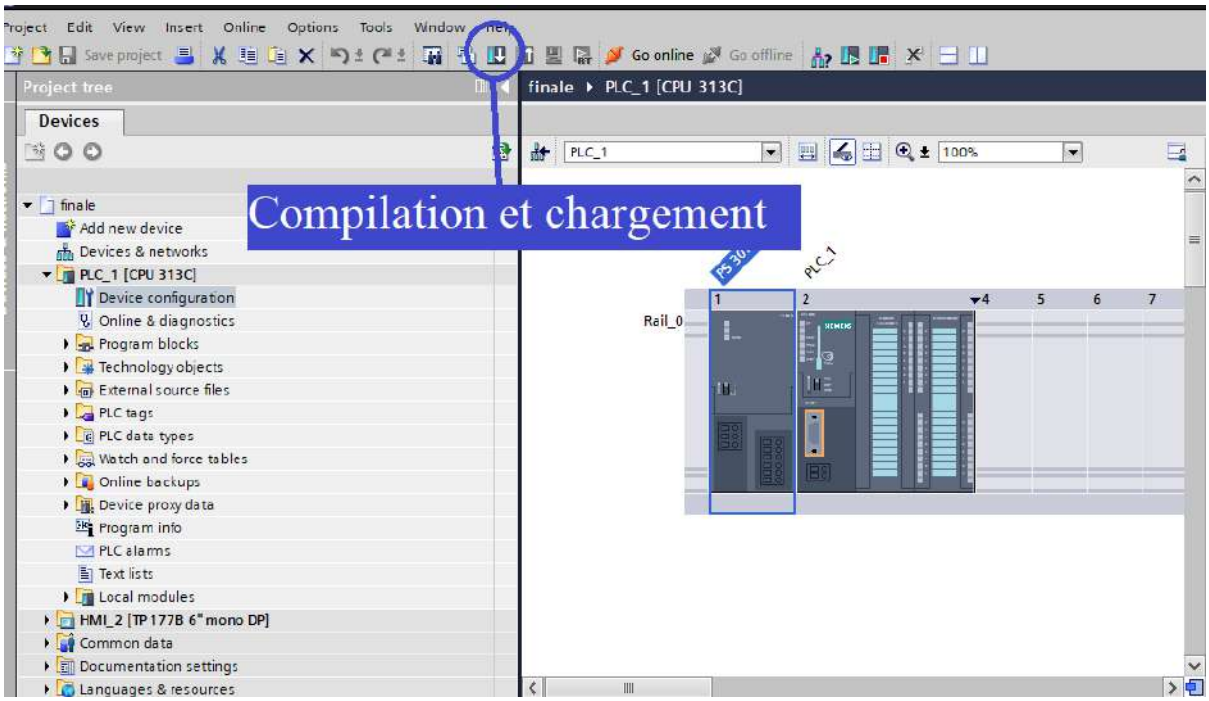


Figure 21 Compilation et chargement

Pour charger la configuration dans l'automate, on effectue un clic sur l'icône ***« charger dans l'appareil »***. La fenêtre ci-dessous s'ouvre et vous devez faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si vous choisissez le mode PN/IE, l'API doit posséder une adresse IP.

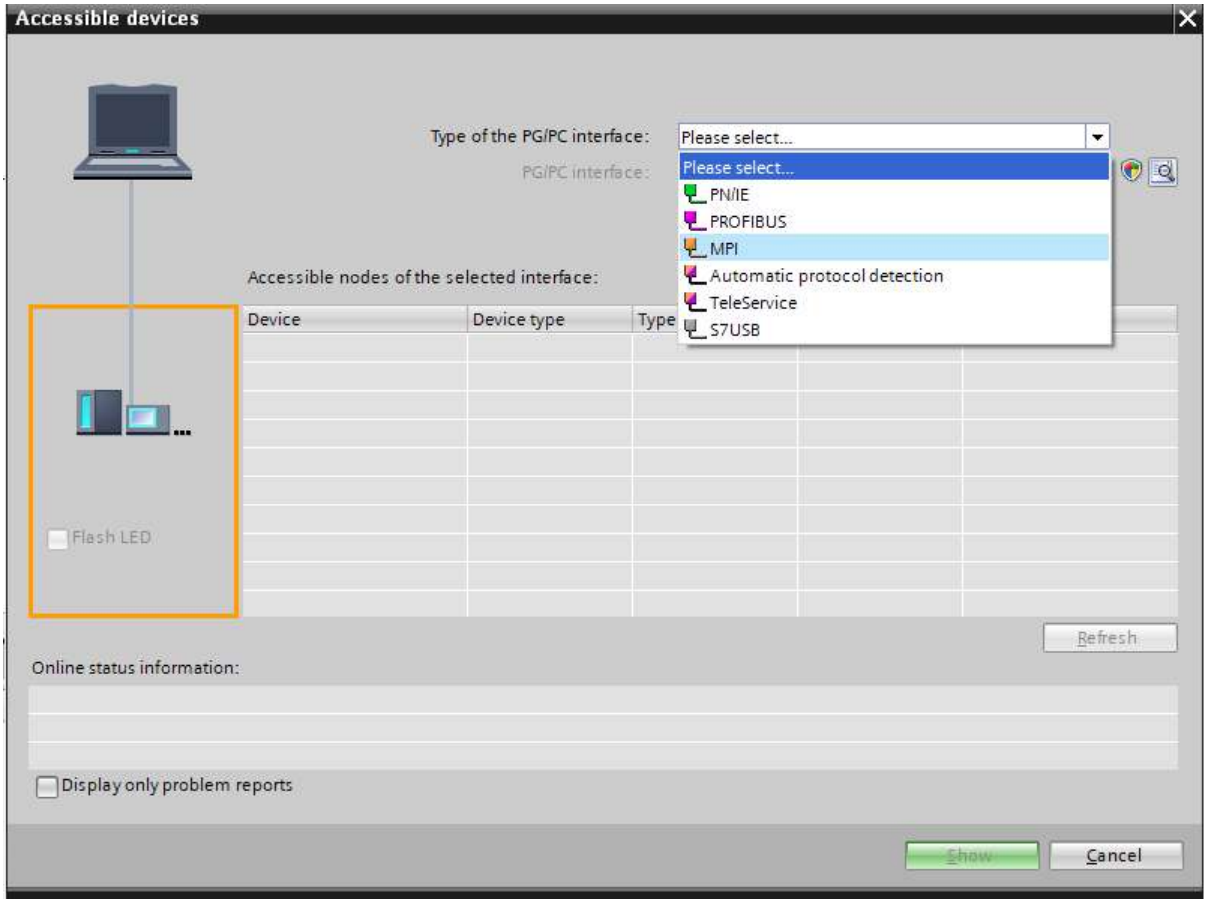


Figure 22 Choix de connexion

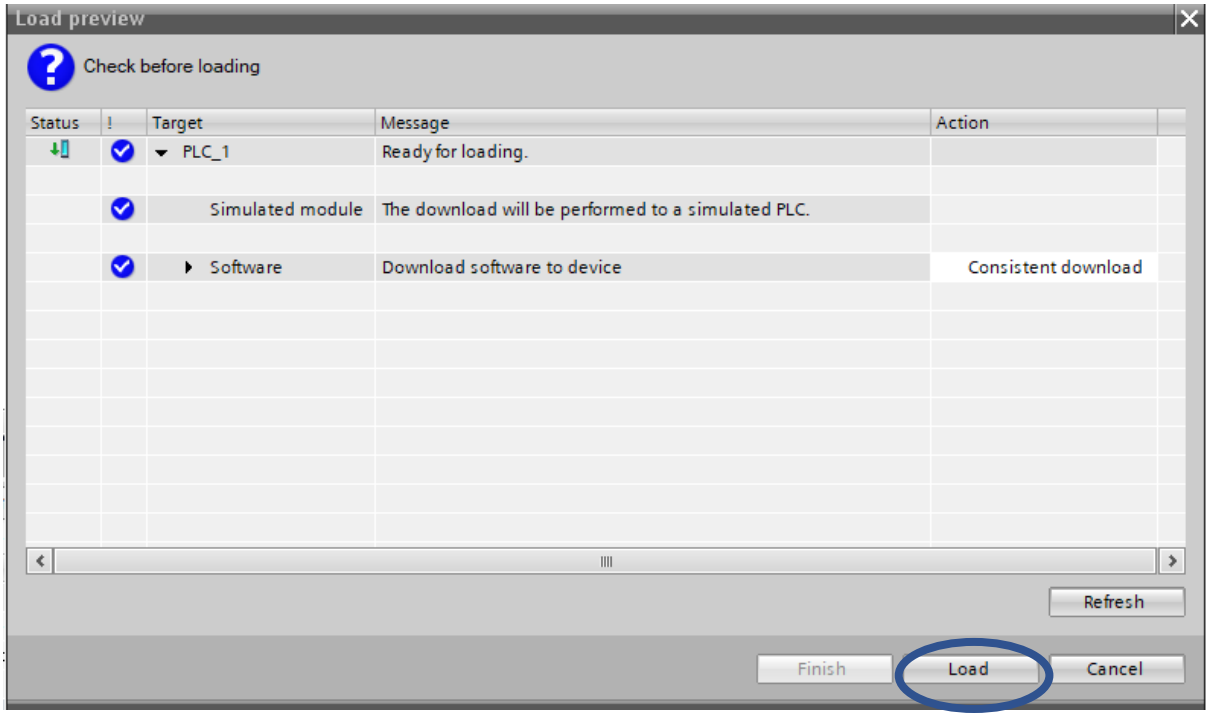


Figure 23 Chargement d'un programme

Pour une première connexion ou pour charger l'adresse IP désirée dans la CPU, il est plus facile de choisir le mode de connexion MPI et de relier le PC à la CPU via le ***« PC Adapter ».***

Si le programme trouve un appareil, il figure dans la liste en bas de la fenêtre. La touche ***« Flash. LED »*** permet de faire clignoter une LED sur la face avant de l'appareil afin de s'assurer que l'on est connecté à l'appareil désiré.[7]

9.10. Création d'une table de variables

Afin de faciliter la programmation, il est intéressant de créer une table de variables.

Pour créer une variable, il suffit d'indiquer le nom de la variable et son adresse.



Figure 24 Table des variables

9.11. Types de variables

• Identificateurs d'opérande :

– Entrée (I)

– Sortie (Q)

– Mémento (M)

• Tail de chaque type de variables :

– Bool (1 bit) 1 ou 0

– Byte (8 bits)

– Word (16 bits)

– DWord (32bits) – …

9.12. Blocs d'organisation

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement du programme cyclique et des programmes déclenchés par alarmes, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs **[6]**.

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

L'exécution d'un OB peut être interrompue par l'appel d'un autre OB (les OB de priorité plus élevée interrompent les OB de priorité plus faible). Les blocs d'organisations les plus prioritaires sont ceux de la mise en route (OB100, OB101 et OB102) et le moins prioritaire est le cycle en arrière-plan (OB90).

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

1. Cycle libre (OB1) : il s'exécute d'une façon continue. Son traitement constitue le traitement normal pour les automates programmables. Le système d'exploitation appelle l'OB1 cycliquement et déclenche ainsi le traitement cyclique du programme utilisateur.

2. Alarmes horaires (OB10 à OB17) : elles peuvent être déclenchées une seule fois à un moment donné (indication de temps absolue avec date) ou périodiquement avec indication du commencement et de la fréquence de répétition (par exemple, toutes les minutes, toutes les heures, tous les jours.

3. Alarmes temporisées (OB20 à OB23) : elles permettent l'exécution retardée de certaines parties du programme utilisateur. Ce retard doit être définit précédemment.

4. Alarmes cycliques (OB30 à OB38) : elles sont déclenchées à des intervalles de temps précis. La période de déclenchement est toujours un multiple entier de la période de base de 1 ms.

5. Alarmes de processus (OB40 à OB47) : elles réagissent à des signaux provenant des modules (SM, CP ou FM). Elles sont déclenchées lorsqu'un module de signaux pouvant générer des alarmes de processus, avec validation d'alarme de processus paramétrée, transmet un signal de processus reçu à la CPU ou lorsqu'un module de fonction de la

CPU signale une alarme.

6. Alarme DPV1 (OB55 à OB57) : les esclaves DPV1 (modules décentralisés) peuvent déclencher des alarmes de diagnostic, de processus et de débrochage.

7. Alarme multiprocesseur (OB60) : le mode "multiprocesseur" correspond au fonctionnement simultané de plusieurs CPU (quatre au maximum) dans un châssis central. Lors du traitement de l'alarme multiprocesseur, le programme utilisateur émetteur ainsi que les programmes utilisateur s'exécutant dans les autres CPU vérifient s'ils connaissent la tâche et réagissent ensuite selon la programmation.

8. Alarmes de synchronisme d'horloge (OB61 à OB64) : elle permet la synchronisation du programme utilisateur contenu dans une CPU avec d'autres programmes répartis dans un réseau.

9. Erreur de redondance (OB70 et OB72) : elle est appelée lorsqu'une perte de redondance se produit sur le réseau PROFIBUS DP, perte de redondance des CPU, erreur de synchronisation ou erreur dans un module SYNC.

10. Erreurs asynchrones (OB80 à OB87) : elles sont appelées lorsqu'apparaît :

a. une erreur de temps : le dépassement du temps de cycle maximal, le saut d'alarmes horaires parce que l'heure a été avancée, un retard excessif pour le traitement d'une classe de priorité.

b. une défaillance dans l'appareil de base ou dans un appareil d'extension : de la tension d'alimentation 24 V, d'une pile, du système de sauvegarde entier.

c. débrochage : erreur sur l'interface au réseau MPI, erreur d'exécution du programme (l'OB correspondant ne démarre pas) ou erreur de communication.

11. Cycle en arrière-plan (OB90) : Il permet d'exécuter des processus à durée non critique et ainsi d'éviter des temps d'attente.

12. Mise en route (OB100 à 102) : On distingue entre les modes de mise en route suivants :

a. redémarrage (n'existe pas pour les S7-300 et S7-400H) : OB101

b. démarrage à chaud (le seul possible pour S7-300) : OB100

c. démarrage à froid : OB102

13. Erreurs synchrones (OB121 et OB122) : Elles sont appelées lorsqu'apparaît une erreur de programmation (des temporisations adressées manquent, un bloc appelé n'est pas chargé…) ou lorsqu'une opération STEP 7 accède à une entrée ou une sortie d'un module de signaux à laquelle aucun module n'était associé lors du dernier démarrage, par exemple : erreur en cas d'accès direct à la périphérie (module défaillant ou manquant), accès à une adresse de périphérie inconnue de la CPU.[6]

9.13. Fonctions et blocs fonctionnels

On peut programmer chaque bloc d'organisation en tant que programme structuré en créant des fonctions (FC) et des blocs fonctionnels (FB) **[6].**

• Les blocs fonctionnels (FB) sont des blocs de code associés à des blocs de données d'instance, dans les quels sont sauvegardés les paramètres effectifs et les données statique des blocs fonctionnels.

• Les fonctions (FC) sont des blocs de code sans rémanence, c'est-à-dire qu'ils ne sont pas associés à des blocs de données, les paramètres effectifs ne sont pas sauvegardés automatiquement.

De plus il existe les blocs fonctionnels système (SFB) et les fonctions système (SFC), qui sont des fonctions préprogrammées. Ils peuvent être appelés à partir du programme utilisateur. On trouve des fonctions système pour la copie de blocs de données, le contrôle du programme utilisateur, la gestion des alarmes horaires et temporisées [6]

9.14. Bloc de données

Les blocs de données (DB) servent à l'enregistrement de données utilisateur. Les blocs de données globaux servent à l'enregistrement de données qui peuvent être utilisées par les autres blocs. Les blocs de données d'instances sont affectés à des blocs fonctionnels **[6]**.

Les différents blocs cités ci-dessus peuvent être édités avec l'application 'CONT LIST LOG'.

9.14.1. Le langage de programmation des blocs

TIA PORTAL dispose de plusieurs modes de représentation, selon les goûts et l'état des connaissances, le programme peut être conçu sous forme de liste d'instruction puis converti en un autre mode de représentation. En représentent certaines règles sur le schéma suivent :

CONT : le schéma à contacts s'apparente aux schémas de circuits électriques, il fait appel à des symboles, tels que les contacts et bobines. Ce mode de représentation convient particulièrement à tous les familiers de technologie des contacteurs.

LIST : la liste d'instructions se compose d'opérations TIA PORTAL. List permet de programmer librement (et parfois de manière complexe). Ce mode de représentation est plus

particulièrement destiné aux programmeurs qui maîtrisent déjà d'autres langages de programmation.

LOG : le programme est un langage de programmation graphique qui utilise des boites logiques. Le signe placé dans la boite symbolise la fonction (par exemple & opération logique ET). Ce mode de représentation permet à des non programmeurs, par exemple des spécialistes des procédés industriels, d'accéder à la programmation.

GRAPH : Le langage de programmation graphique optionnel GRAPH vous permet de programmer des commandes séquentielles. Ceci implique la création d'une succession d'étapes, la définition des actions associées à chaque étape et celle des transitions indiquant les possibilités d'évolution entre deux étapes successives. Pour définir les actions associées aux étapes, vous utilisez un langage de programmation spécial (similaire à LIST), alors que pour déterminer les conditions de réceptivité des transitions, vous utilisez une représentation sous forme de schéma à contacts (langage de programmation CONT restreint).

GRAPH permet la représentation très claire de séquences même complexes, ce qui favorise une programmation et une recherche d'erreurs efficaces.

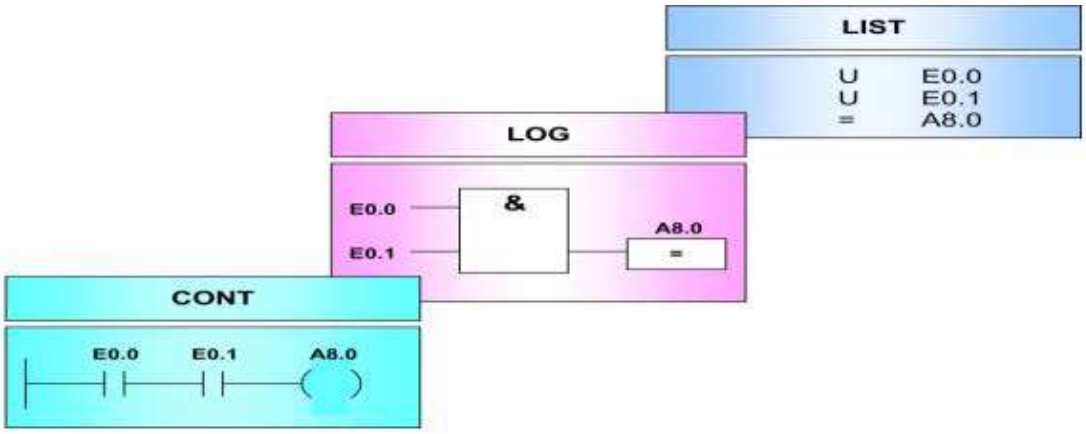


Figure 25 Langage de programmation des blocs

9.14.2. Le langage Ladder

Le langage à contact est adapté à la programmation de traitements logiques, il utilise le schéma développé. Nous retrouvons : La fonction ET en utilisant des contacts en série La fonction OU en utilisant des contacts en parallèle

9.14.3. Symbolique Ladder

Les symboles utilisés dans ce mode de programmation sont :

- ┤ ├ : variable d'entrée ou contact à fermeture
- ┤/├ : variable d'entrée complémentée ou contact à ouverture,
- -( )- : variable de sortie (Q124,2 sortie automate, etc..),
- -(S)- : sortie mise à un mémorisée (S = set),
- -(R)- : sortie mise à zéro mémorisée (R = reset),
- ── : connexions horizontales,
- │ │ : connexions verticales.

9.15. Le simulateur des programmes PLCSIM :

L'application de simulation de modules S7-PLCSIM permet l'exécution et le test du programme utilisateur destinés aux CPU S7-300.

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme, figure 4.9 (par exemple, activer ou désactiver des entrées) [26].

En outre, S7-PLCSIM possède les fonctions suivantes :

• On peut créer des "fenêtres" dans lesquelles on a la possibilité d'accéder aux zones de mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par adressage symbolique (il faut juste charger la table des mnémoniques dans 'options', puis sur 'outils' 'insérer mnémoniques').

• On peut sélectionner l'exécution automatique des temporisations ou encore les définir et les réinitialiser manuellement.

• On a la possibilité de changer l'état de fonctionnement de la CPU (STOP, RUN et RUNP) comme pour une CPU réelle. De plus, on dispose d'une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du programme.

Bien que l'AP de simulation soit essentiellement logiciel, TIA Portal le considère comme une réelle composante matérielle, à quelques différences près :

• Contrairement à ce qui se passe avec une CPU réelle lors de la mise à l'arrêt de la CPU, l'état des sorties ne change pas.

• La CPU n'attend pas le début ou la fin du cycle pour actualiser une donnée qu'on a modifiée. Toute modification dans une fenêtre entraîne l'actualisation immédiate du contenu de l'adresse en mémoire.

• Les options d'exécution permettent de choisir le mode d'exécution du programme par la CPU :

o La commande Cycle unique exécute un cycle du programme, puis attend qu'on démarre l'exécution du cycle suivant.

o La commande Cycle continu exécute le programme de la même manière que dans un AP réel : elle démarre un nouveau cycle aussitôt que le cycle précédent est terminé.

• On peut déclencher manuellement les OB d'alarme (aller dans 'exécution' puis 'déclenchement OB Erreur').

9.15.1. Exécution du programme

Une fois le programme est chargé dans la CPU, on peut exécuter le programme. Mais Il faut s'assurer d'abord que l'exécution il est on mode cycle continu sélectionné comme mode. Pour sélectionner le mode d'exécution continu du programme, on choisit la commande « Exécution Mode Cycle continu », ou bien clique sur le bouton correspondant dans la barre d'outils, et pour mettre la CPU en mode RUN et démarrer l'exécution du programme, on clique sur la case à cocher RUN (Marche) ou RUN P dans la fenêtre « CPU ». Dans le simulateur S7-PLCSIM, on a représenté l'ensemble des variables d'entrées, et de sorties sous forme de fenêtres. Pour visualiser le fonctionnement de l'automate, on suit les étapes de fonctionnement de la machine avec des cliques sur les entrées pour visualisation des sorties.

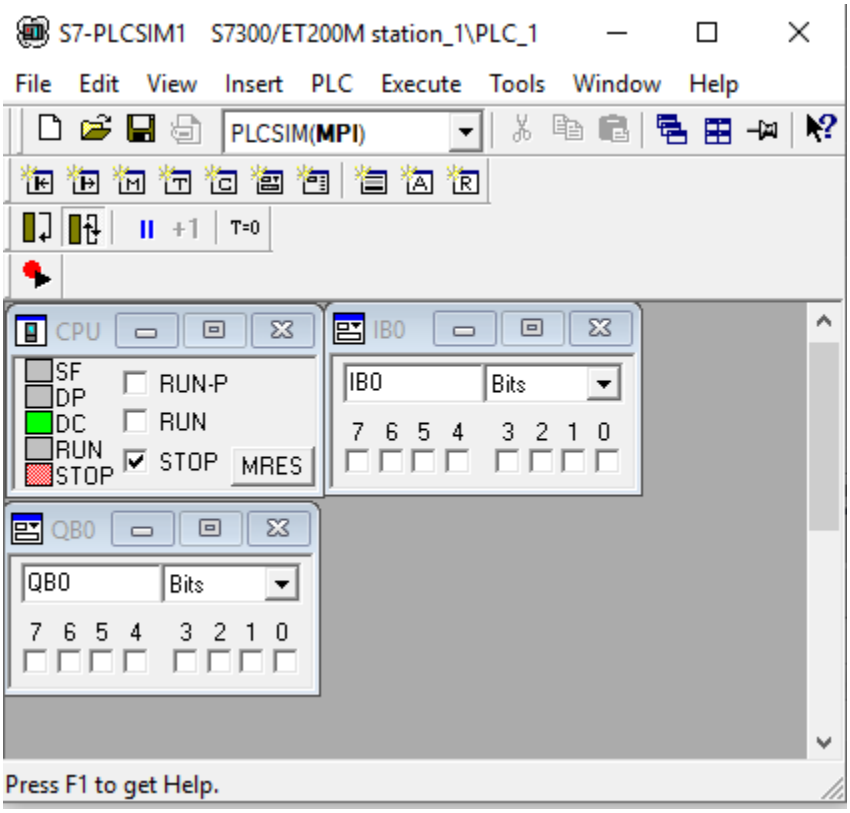


Figure 26 Fenêtre de configuration du simulateur

9.15.2. État de fonctionnement de la CPU

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

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

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

9.16. SIMATIC WinCC

Le SIMATIC WinCC dans le Totally Integrated AutomationPortal (TIA Portal) fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement d'ingénierie homogène pour la programmation et la configuration de solutions de commande, de visualisation et d'entraînement. Ce concept d'ingénierie est une avancée fondamentale dans le développement de logiciels et représente le développement continu et conséquent de la philosophie TIA.

WinCC dans le TIA Portal est le logiciel pour toutes les applications IHM allant de solutions de commande simples avec des Basic Panels aux applications SCADA pour systèmes multipostes basés sur PC. La gamme de solutions offerte par le prédécesseur de SIMATIC WinCC flexible s'en trouve considérablement élargie.

Pour les applications très complexes avec des solutions Plant Intelligence ou des architectures redondantes, il existe également SIMATIC WinCC V7, tandis que WinCC Open Architecture s'adresse aux applications à fort besoin d'adaptation aux spécificités du client, ainsi qu'aux plateformes non-Windows.

9.17. Interface Homme-Machine (IHM)

L'interface homme-machine (IHM) est l'interface utilisateur qui relie l'opérateur au dispositif de commande d'un système industriel.

Les systèmes de contrôle industriel intègrent des équipements et des logiciels conçus pour surveiller et contrôler le fonctionnement des machines-outils et des appareils associés dans les environnements industriels, Une IHM comprend des composants électroniques de signalisation et de contrôle des systèmes d'automatisation.

9.17.1. Création de station HMI

Pour crée un HMI il suffit seulement à Double-cliquez sur "Ajouter un appareil" et sélectionnez le pupitre IHM.

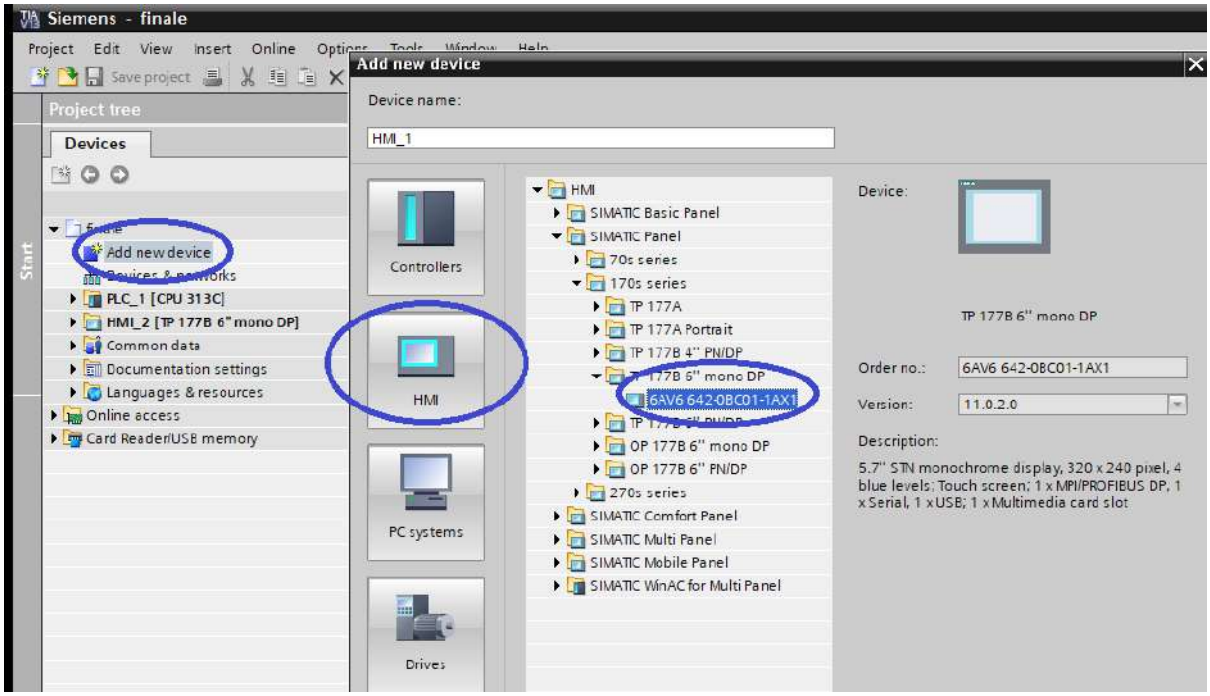


Figure 27 Création d'un HMI

9.17.2. Création de la liaison entre le projet IHM et l'API

La communication entre les pupitres opérateur et les automates SIMATICS7 peut être réalisée via les réseaux suivants :

- ✓ MPI (Multi point Interface) : Le pupitre opérateur est connecté à l'interface MPI de l'automate.
- ✓ PROFIBUS (Process Field Bus)
- ✓ Ethernet.



Figure 28 Cable Profibus

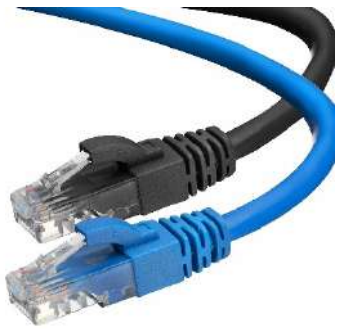


Figure 29 Cable Ethernet

La connexion qu'on a utilisé dans notre projet va être réalisée via l'interface MPI

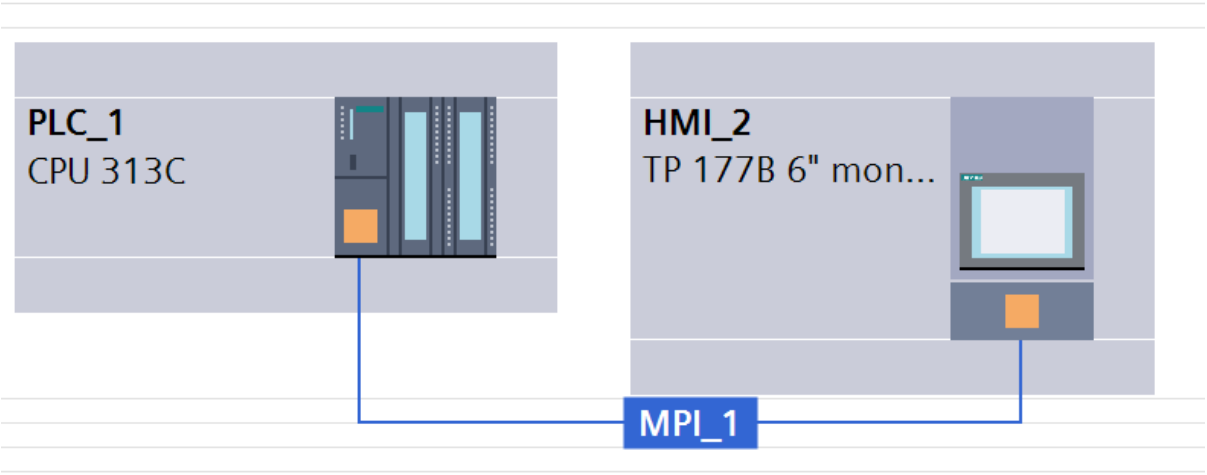


Figure 30 Liaison entre IHM et API

9.17.3. ***Programmation des Boutons et des voyants***

On ajoute les boutons et les voyants dans notre vue à partir de la bibliothèque puis on configure le nom et adresse, enfin en compile pour voir le bon fonctionnement de ces derniers. Pour les boutons dans Évènement en ajoute la fonction ou la variable à accomplir. Pour les voyants on configure dans l'Animation les états à afficher pour la variable associée.

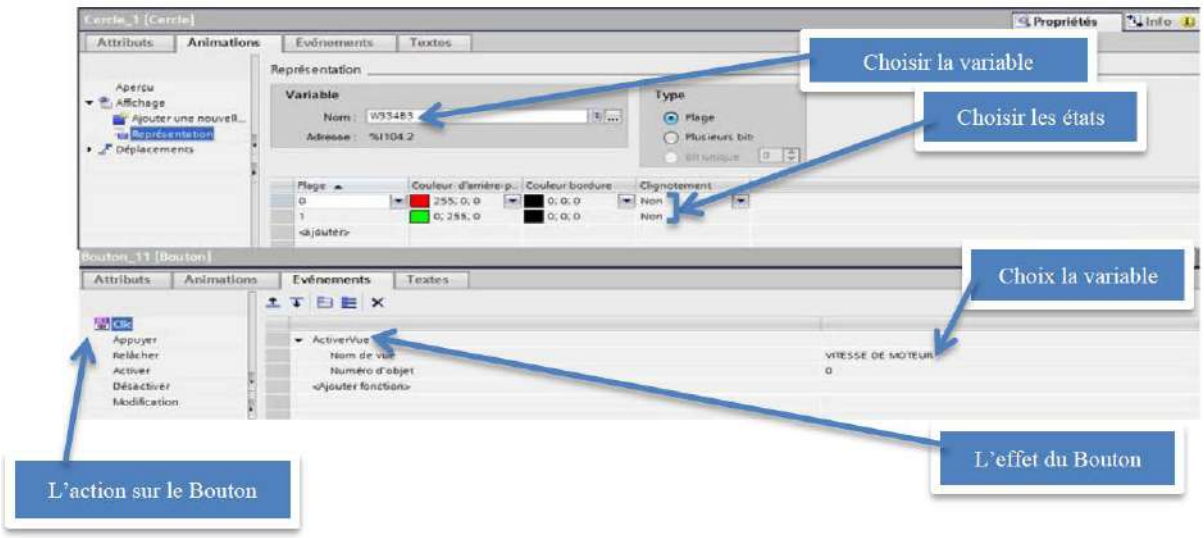


Figure 31 : Programmation bouton et voyants

10. Conclusion

Ce chapitre est consacré à la description des systèmes automatises et les automates programmables industriels d'une manière générale.

Nous avons présente les caractéristiques de l'API S7 300 de la firme SIEMENS qui est l'automate choisit pour contrôler le moteur asynchrone.

Comme on a donné une présentation pour le logiciel TIA PORTAL et quelques notions de base sur le simulateur PLCSIM et le logiciel de supervision WinCC qui nous permet une commande facile du processus à partir d'une interface compacte à écran tactile où les organes de commande de toutes sortes (boutons poussoirs, interrupteurs, élément d'affichage, graphe…) Un simple clic sue un bouton permet de démarrer/arrêter un dispositif.

CHAPITRE II : MOTEUR ASYNCHRONE TRIPHASE

1. Introduction

La machine asynchrone, connue également sous le terme « anglo-saxon » de machine à induction, 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. [8] Du fait de sa simplicité de construction, d'utilisation et d'entretien, de sa robustesse et son faible prix de revient, la machine asynchrone est très couramment utilisée comme moteur dans une gamme de puissance allant de quelques centaines de watts à plusieurs milliers de kilowatts

2. Historique Sur Les Machines Asynchrones :

Historiquement le dix-neuvième siècle fût l'époque des grandes découvertes en Electrotechnique dont les bases fondamentales ont été établies (1820--1830) par des hommes de science parmi lesquels on peut citer, OERSTED, AMPERE, BIOT, SAVART, LAPLACE, OHM, FARADAY ; Plus tard en (1873) MAXWELL formalisa les lois de l'électromagnétisme moderne dans son ouvrage fameux ;"Treatise on Eelectricity and Magnétisme". Mais ce n'est qu'à partir de (1870) que l'électrotechnique industrielle s'affirma notamment grâce à la production d`énergie électrique par les génératrices à courant continu (dynamos) de gramme et de siemens. Ensuite, dans les années (1880), furent conçus les alternateurs et les transformateurs polyphasés. Les premiers devaient concurrencer et détrôner les dynamos pour la production de l'électricité, enfin les travaux du yougoslave TESLA et de l'italien FERRARIS complétèrent les systèmes à courants alternatifs polyphasés par la conception et la construction des machines d'induction ou asynchrones en (1888). [9] Les machines asynchrones couvrent actuellement l'essentiel de besoins de la transformation d'énergie électrique en énergie mécanique, A titre d'exemple, elles sont utilisées pour la quasi-totalité des fonction auxiliaires d'une centrale de production électrique ; ils sont généralement de même dans les procédés industriels. L'étude de ces machines acquit donc une grande importance

3. Présentation

La machine se compose de deux pièces principales :

• Le stator est relié au réseau ou un variateur de vitesse

• Le rotor est constitué de conducteurs en court-circuit qui sont parcourus par des courants induits par le champ magnétique créé par les courants statoriques. Cette machine peut, selon sa construction, être reliée à un réseau monophasé ou polyphasé (généralement triphasé car c'est celui de la distribution). La machine asynchrone est la machine électrique la plus utilisée dans le domaine des puissances supérieures à quelques kilowatts car elle offre alors le meilleur rapport qualité prix. Surtout depuis l'apparition dans les années 1980 de variateurs permettant de faire varier la fréquence de rotation du moteur dans une large gamme

4. Constitution de la machine asynchrone :

Constitution des machines asynchrone : Les moteurs asynchrones, figure (32) sont pratiquement tous des moteurs triphasés. Ils sont basés sur l'entraînement d'une masse métallique par l'action d'un champ tournant.

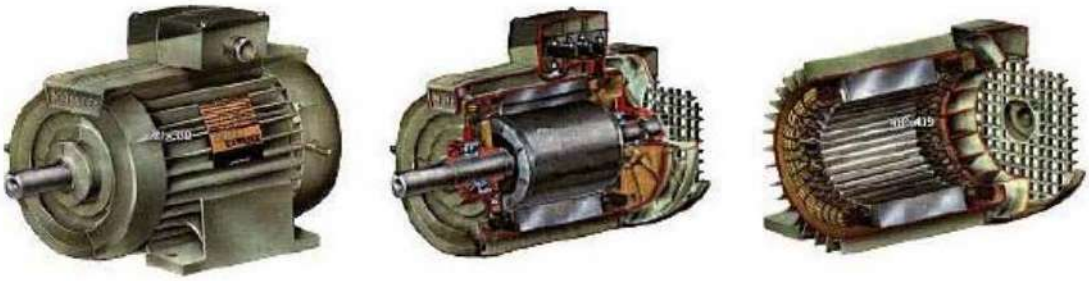


Figure 32 Moteurs asynchrones

Les moteurs asynchrones triphasé sont des convertisseurs électro-magnétique qui transforment l'énergie électrique fournie par le réseau en énergie mécanique dans la vitesse de rotation N pour une fréquence donné 'F' dépend de la charge et dans la quelle F ≠ P N. [9] Les moteurs asynchrones comportant deux armateurs coaxiaux à champ tournant, l'une est fixe (c'est le stator) et l'autre mobile (c'est le rotor). [10]

4.1. Le stator

Il est appelé inducteur ou primaire, il est fixé au bâti et former d'une carcasse ayant l'aspect général d'un cylindre creux, à l'intérieur duquel se trouve le circuit magnétique composé d'un empilage de tôles ayant la forme de couronnes circulaire a leur périphérique interne, ces tôles comportent un certain nombre d'encoches régulièrement réparties qui, par suite de l'empilage créent des rainures, ou sont logés des faisceaux du bobinage. La figure (33) représente un stator comporte un bobinage triphasé, dont chaque phase ne comporte qu'une bobine occupant deux encoches diamétralement opposées, les trois phases sont identiques mais décalées entre elle de $2\pi/3$.

Cet enroulement (bobines), est alimenté en triphasé par l'intermédiaire de la plaque aborne ce qui permet de l'alimenter en étoile ou en triangle, et possède p paires de pôles, [10]. Les courants alternatifs dans le stator créent un champ magnétique B tournant à la pulsation de synchronisme.

$$\Omega_S = \frac{w}{p}$$

Ω_S : vitesse synchrone de rotation du champ tournant en $rad.s^{-1}$.
W : pulsation des courants alternatifs en $rad.s^{-1}$. $w = 2.p.f$
P : nombre de paires de pôles



Figure 33 Stator d'une machine asynchrone

4.2. Le Rotor

Le rotor n'est lié électriquement à aucune source d'énergie, (ni continue, ni alternative) ce qui simplifie beaucoup sa construction on distingue deux types de rotor, [10].

- *Rotor à bagues*

Ce rotor à pôle lisse comporte dans ces rainures un enroulement identique à celui du stator, les trois phases sont branchées en étoile ce qui permet d'insérer un rhéostat dans leur circuit, ce rhéostat qui est mise en marche normale et permet d'assurer des meilleures conditions de démarrage comme présente la figure (34).

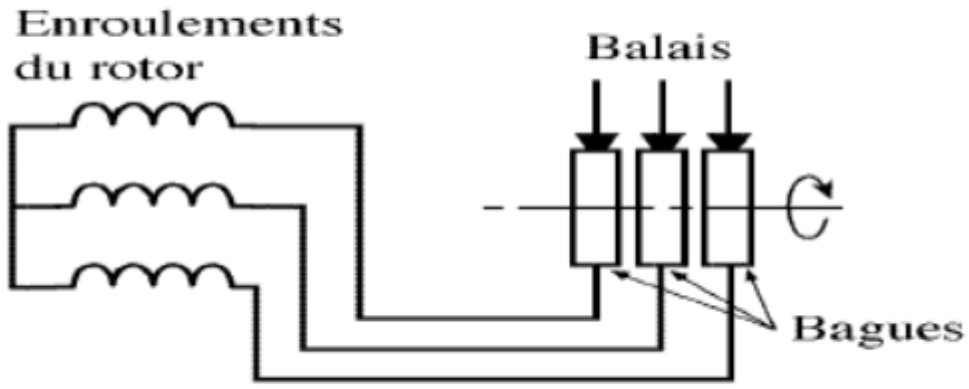


Figure 34 Principe d'un rotor bobine

Signalons que, si le nombre des pôles du rotor est obligatoirement le même, que celui de stator, le nombre de phases peut être déférent. Cependant, il n'est pas intéressant pour un moteur à bague d'augmenter le nombre des phases du rotor car il faudrait augmenter le nombre des bagues et des balais, [10].

- *Rotor à cage*

L'enroulement est remplacé par des barres de cuivre ou d'aluminium logées dans des encoches, et réunie à leurs extrémités par deux couronnes de cuivre ou d'aluminium. Ces cages comportant généralement des barreaux décalés afin de réduire les harmoniques d'encoches, il en résulte une légère diminution de la F.E.M induite par le champ tournant statorique dans ces barreaux. Un tel rotor comme présente la figure (35) est très robuste, depuis sa construction est particulièrement économique. La cage étant généralement réalisée avec l'aluminium que l'on coule dans les encoches préparées à l'avance. En effet il n'est pas nécessaire d'isoler les barres et la masse du rotor, car les courants induits s'établissent surtout dans les barres, et leur étude théorique est identique à celle des moteurs à bagues. Si N désigne le nombre des barres d'une cage, les extrémités des barres, étant en court-circuit par

les flasques. Un rotor à cage est assimilable à un rotor à bagues qui aurait q=N phases si la cage tourne dans un champ bipolaire alors qu'il a q= N / p phases si la cage tourne à un champ 2p pôles.

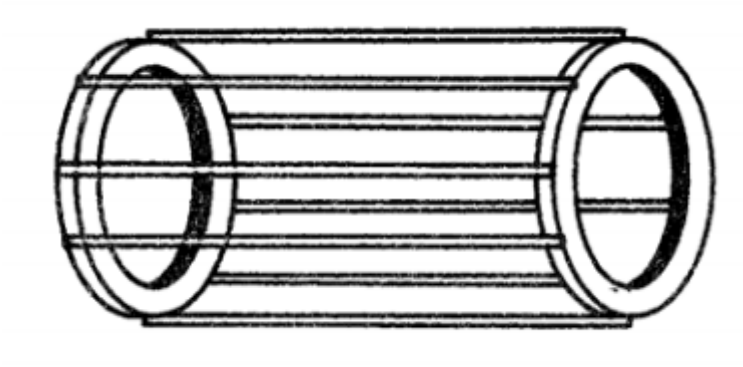


Figure **Erreur ! Utilisez l'onglet Accueil pour appliquer 0 au texte que vous souhaitez faire apparaître ici.**35 Principe d'une cage d'écureuil

Par comparaison avec les moteurs à bagues, les moteurs à cage ont l'avantage d'être robustes et du coût plus faible. Il n'est pas possible de faire varier la résistance de leur rotor. Ce qui rend défavorable les conditions de démarrage quand on l'alimente à tension et à fréquence constantes. On remédie cet inconvénient en utilisant soit des rotors à double cage, soit des rotors à une seule cage, mais à barre en la cause de leur profil, soit des moteurs à encoches très profondes : barres « lames de sabre », [10]. • Entrefer : L'entrefer est l'espace entre le stator et le rotor.

5. Symboles

La figure (36) présente les symboles moteurs à cage d'écureuil et a rotor bobiné par la suite



Figure 36 Symboles d'une machine asynchrone

6. Principe de fonctionnement du moteur à induction

Le principe de fonctionnement des moteurs asynchrones est basé sur la production d'un champ tournant. Telle que on considère un aimant tournant produit un champ électromagnétique qui traverse un cylindre métallique. Les génératrices du cylindre placées dans le champ se comporte comme des brins actifs (AB et CD). Elles sont parcourues par un courant proportionnel à F et engendrent des forces perpendiculaires qui constituent le couple, figure (37).

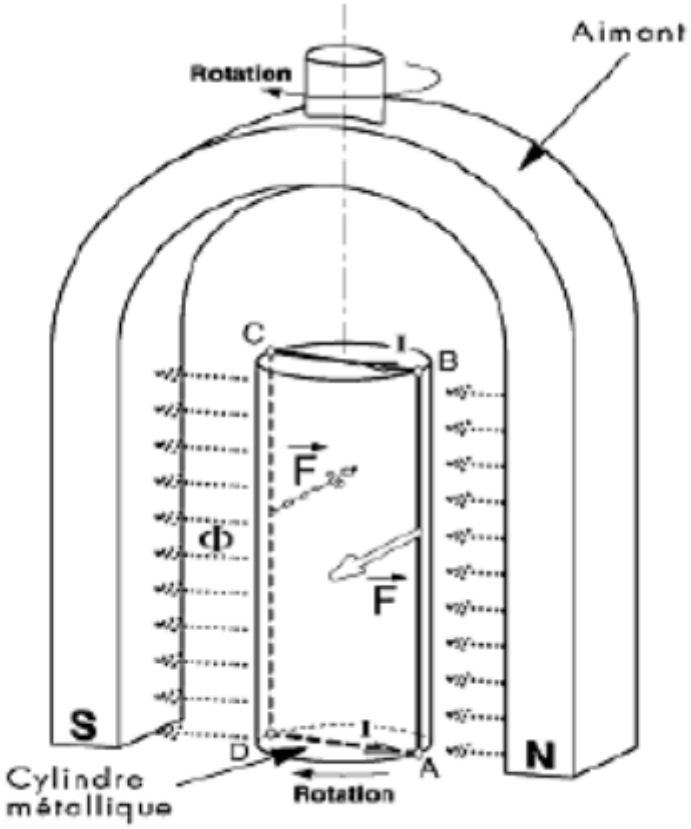


Figure 37 Principe de base d'un moteur

On constate que le cylindre suit la rotation de l'aimant avec un léger décalage appelé glissement. Sur les moteurs asynchrones triphasé. Le champ tournant est produit par trois bobinages fixes géométriquement décalés de 120o, et parcourus par des courants alternatifs présentant le même décalage électrique. Les trois champs alternatifs 20 qu'ils produisent se composent pour former un champ tournant d'amplitude constante comme est présenté sur la figure ci-dessous, [12].

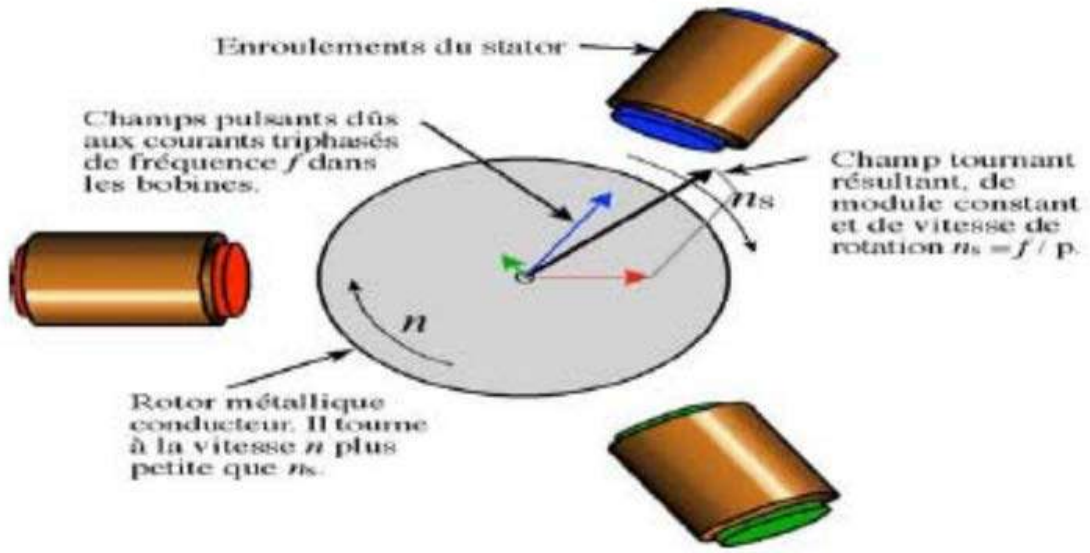


Figure 38 Principe de fonctionnement d'un moteur à induction

7. Le glissement

Le couple moteur ne peut exister que si un courant induit circule dans la spire. Ce couple est déterminé par le courant qui circule dans la spire et qui ne peut être que s'il existe une variation de flux dans cette spire. Il faut donc qu'il y ait une différence de vitesse entre la spire et le champ tournant. C'est la raison pour laquelle un moteur électrique fonctionnant suivant le principe que nous venons de décrire est appelé "moteur asynchrone".

La différence entre la vitesse de synchronisme (Ns) et celle de la spire (N) est appelée "glissement" (g) et s'exprime en % de la vitesse de synchronisme.

$$g = [(Ns - Nr) / Ns] \times 100$$

8. Vitesse de synchronisme

On a le glissement $g = (n_s - n)/n$ Alor $n = n_s(1-g)$

Et la vitesse de synchronisme $n_s = \frac{f}{P}$ avec

$$g = \frac{n_s - n_r}{n_s}$$

Donc **$n = f/p\ x(1-g)$**

g : glissement

p : nb paire de pôles

n : vitesse du rotor (tr/s)

ns : vitesse de synchronisme (tr/s)

f : fréquence de la tension (Hz)

Pour faire varier la vitesse de rotation du moteur, il existe trois méthodes :

Agir sur la constitution du moteur : couplage d'enroulements ou enroulement séparés

Agir sur le glissement : utilisation d'un rhéostat rotorique

Agir sur la fréquence de l'alimentation du moteur : variation de vitesse par MLI ou par contrôle vectoriel de flux

8.1. Action sur le nombre de pairs de pôles

Agir sur la constitution du moteur : couplage d'enroulements ou enroulements séparés,

8.1.1. Couplage d'enroulements :

Le moteur dispose de deux vitesses de rotation par couplage d'enroulements (ou encore couplage de pôles). Il possède deux bobinages par phase que l'on peut coupler en parallèle (une paire de pôles) ou en série (deux paires de pôles).

- En parallèle

En associant les deux bobinages en parallèle, leurs actions vont se superposer et ils se comportent comme un seul enroulement (une paire de pôles).

Le moteur tournera à sa vitesse maximale.



Figure 39 Bobinage en parallèle

- En série

En associant deux bobinages en série, on double le nombre de pôles (deux paires de pôles).

La vitesse du moteur est divisée par deux.

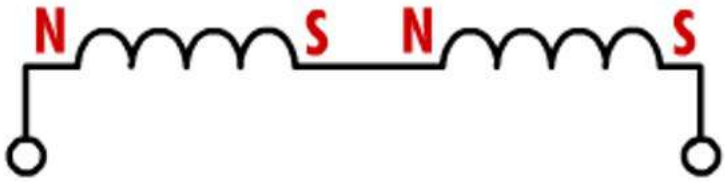


Figure 40 Bobinage en série

8.1.2. Enroulements séparés

Le moteur à enroulements séparés correspond à l'assemblage de deux moteurs à vitesses différentes dans la même carcasse.

L'encombrement d'un tel moteur est plus important que pour un moteur asynchrone à couplage d'enroulements (moteur de type Dahlander) mais le rapport des vitesses peut être quelconque.[13]

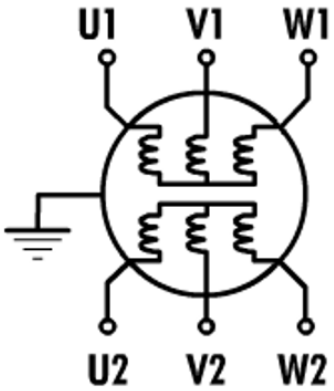
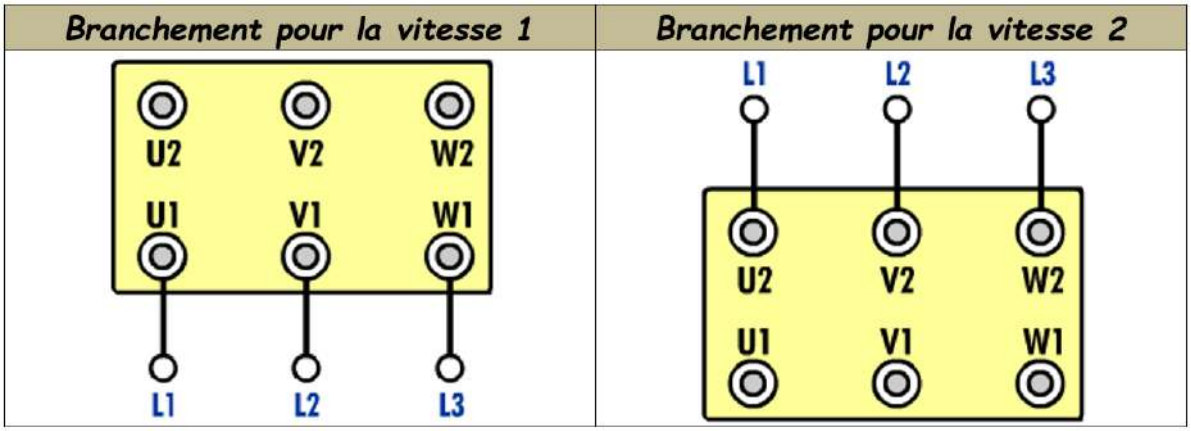


Figure 41 Enroulements séparés



8.2. Action sur le glissement

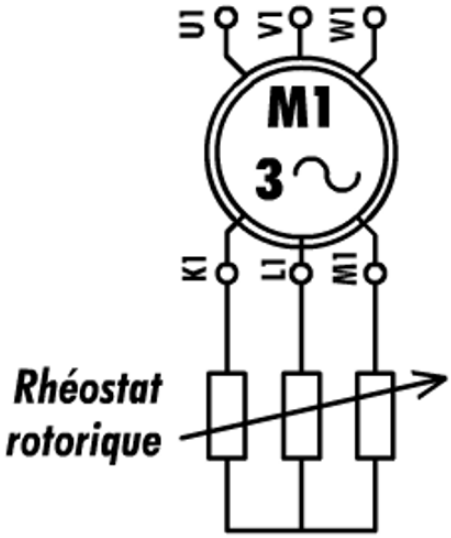
Ce procédé de variation de vitesse s'applique aux moteurs à rotor bobiné.

Les résistances variables d'un rhéostat sont placées en série avec les bobinages du rotor : le réglage du rhéostat modifie le glissement du rotor.

Augmenter le glissement fait baisser la vitesse tout en maintenant un couple moteur maximal dans toute la plage de variation de la vitesse.

Cependant, abaisser la vitesse du moteur conduit à des pertes par effet Joule très élevées. Le rendement de ce type de variateur n'est pas excellent.

Pour d'autres informations, consulter également la fiche sur le démarrage rotorique.[13]



8.3. Action sur la fréquence

Variation de vitesse par Modulation de largeur d'impulsion.

La vitesse de rotation d'un moteur asynchrone étant directement liée a la fréquence de son alimentation, pouvoir agir sur cette fréquence permet de modifier la vitesse du moteur

Il n'est cependant pas possible directement sur le réseau de distribution, c'est pourquoi on utilise la technique de modulation de largeur d'impulsion pour générer un signal électrique dont la fréquence sera contrôlée

La variation de la fréquence de la tension d'alimentation peut être obtenue avec l'utilisation d'un onduleur de tension.

Théoriquement on a $n_s = f/p(1-g)$ donc si f varie n_s varie aussi.

La variation de la fréquence de la tension d'alimentation peut être obtenue avec l'utilisation d'un onduleur de tension.

9. Couplage de la plaque à bornes

Le branchement des bobines sur le réseau se fait au niveau de la plaque à borne située sur le dessus de moteur. On dispose ainsi de 6 connexions, un pour chacune des extrémités des trois bobines. Les bornes sont reliées aux bobines selon la figure (42) ci-dessous.

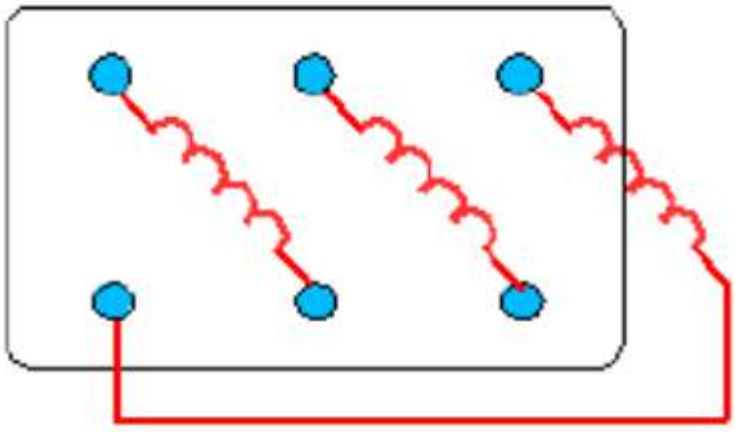


Figure 42 La plaque à bornes

10. Branchement du moteur sur le réseau triphasé

Il n'est pas toujours possible de brancher un moteur asynchrone en étoile ou en triangle. Avec le branchement étoile, la tension à la borne de chacune des bobines est 220 V. Dans le

montage en triangle, chacune des bobines est alimentée en tension nominale de réseau (380 V), [14].la figure (43) ci-dessous présente le couplage des moteurs.

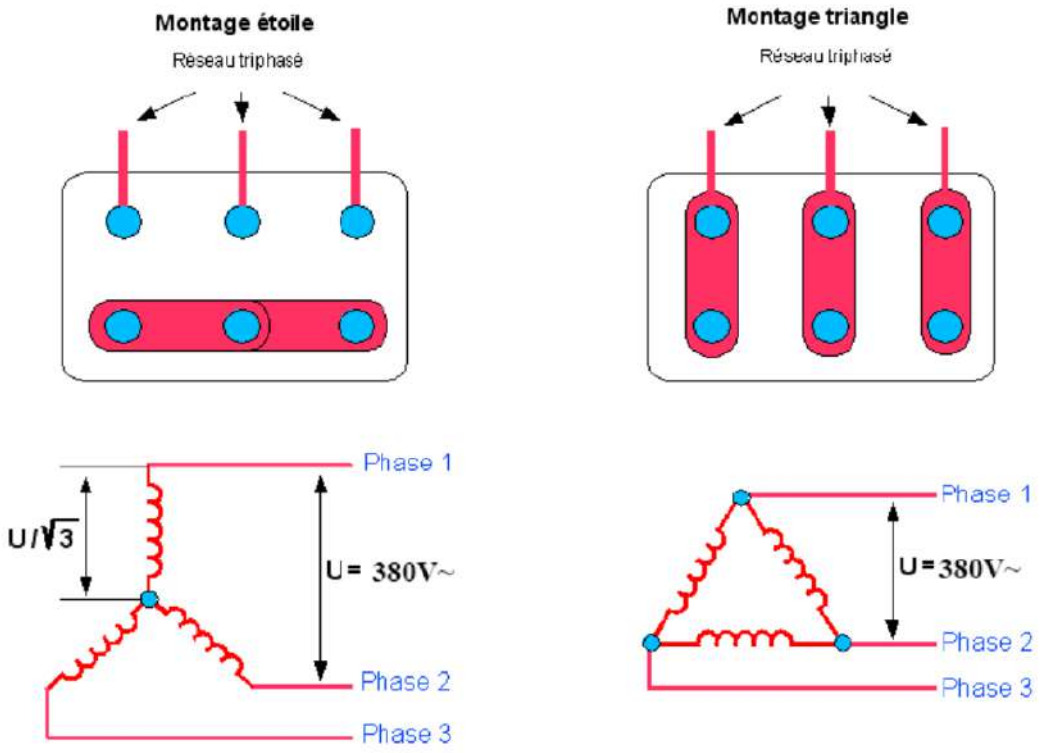


Figure 43 Couplage des moteurs

- sur une plaque signalétique d'un moteur on lit : 380 Y / Δ 220
- le réseau est en 220 / 380 V

De la plaque signalétique on déduit que la tension nominale d'une phase du moteur est de 220 V. Si on branche ce moteur en triangle, la tension aux bornes d'une phase sera de 380 V ce qui est trop élevé.

11. La plaque signalétique

La plaque signalétique d'un moteur asynchrone contient les paramètres suivants :

•Type : référence propre au constructeur

• Puissance : puissance utile délivrée sur l'arbre du moteur.

• Facteur de puissance ou cos phi : permet le calcul de la puissance réactive consommée.

• rendement : permet de connaître la puissance électrique consommée ou absorbée

•Tensions :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 connait alors La vitesse de synchronisme ns du moteur

• Température ambiante : utilisation recommandée maximum

• Fréquence : fréquence du réseau d'alimentation.

• Nombre de phases : moteur triphasé ou monophasé

12. Les avantages et les inconvénients du moteur asynchrone :

Si l'on compare le moteur asynchrone au moteur shunt à courant continu, on constate que les caractéristiques dans leurs parties stables, sont identiques, ce qui conduit aux mêmes applications industrielles. En ce qui concerne le choix d'un moteur pour une application donnée, il est intéressant de signaler les avantages et les inconvénients de ces deux moteurs.

12.1. Les avantages du moteur asynchrone

Comparé au moteur shunt, le moteur asynchrone a l'avantage d'être alimenté directement par le réseau triphasé. Son prix d'achat est moins élevé, il est beaucoup plus robuste car il ne nécessite pratiquement pas d'entretien. Ses deux qualités fondamentales (prix et solidité) résultent du fait qu'il n'a pas de collecteur. En effet, le collecteur est un organe coûteux et fragile qui nécessite un entretien fréquent : changement des balais.

12.2. Les inconvénients du moteur asynchrone

A l'exception du démarrage et de l'inversion du sens de marche que l'on peut résoudre de façon satisfaisante, le moteur asynchrone a des performances très médiocres par rapport à celles du moteur shunt. En effet jusqu'à ces dernières années, l'entraînement idéal était réalisé par le léonard formé un moteur shunt alimenté par un convertisseur de tension, [10].

13. Les variateurs de vitesse

Un variateur électronique de vitesse est un dispositif destiné à régler la vitesse d'un moteur électrique à courant alternatif en faisant varier la fréquence par convertir le signal alternative de fréquence fixe a un signal alternatif de fréquence variable.

14. 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 d'une interface utilisateur

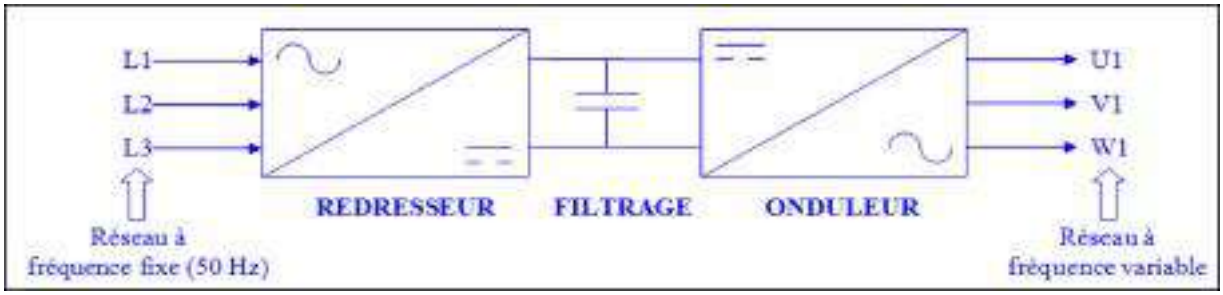


Figure 44 Structure d'un variateur de vitesse 130

- D'un redresseur qui, connecté à une alimentation triphasée (le réseau), génère une tension continue à ondulation résiduelle (le signal n'est pas parfaitement continu). Le redresseur peut être de type commandé ou pas
- D'un circuit intermédiaire agissant principalement sur le "lissage" de la tension de sortie du redresseur (améliore la composante continue). Le circuit intermédiaire peut aussi servir de dissipateur d'énergie lorsque le moteur devient générateur
- D'un onduleur qui engendre le signal de puissance à tension et/ou fréquence variables ;
- D'une électronique de commande pilotant (transmission et réception des signaux) le redresseur, le circuit intermédiaire et l'onduleur [15].

14.1. Interface utilisateur

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 a parti d'un ordinateur.

14.2. Redresseur

Les redresseurs sont les convertisseurs de l'électronique de puissance qui assurent la conversion alternative continue (AC/DC). Alimentés par une source de tension alternative, ils permettent d'alimenter en courant continu le récepteur branché à leur sortie [16]. Nous utilisons le pont triphasé à diodes alimenté par un système de tensions sinusoïdales triphasées, schématisé dans la figure. Les tensions simples des réseaux triphasés équilibrés directs sont :

$$\begin{cases} V_a = V_m \sin(\omega t - \varphi) \\ V_b = V_m \sin(\omega t - \varphi - \frac{2\pi}{3}) \\ V_c = V_m \sin(\omega t - \varphi - \frac{4\pi}{3}) \end{cases}$$

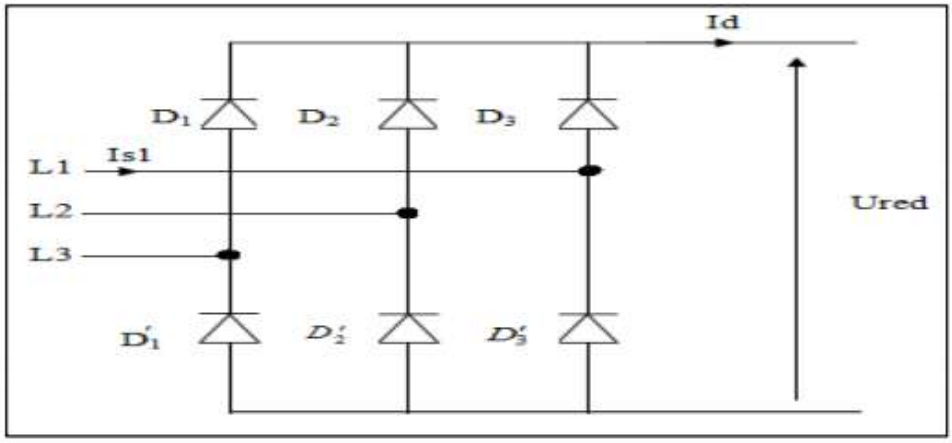


Figure 45 Redresseur triphasé à diodes

14.2.1. Etude du fonctionnement

Les trois diodes D1, D2, D3 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 [5]. La tension redressée est à tout instant la différence entre ces deux tensions, soit :

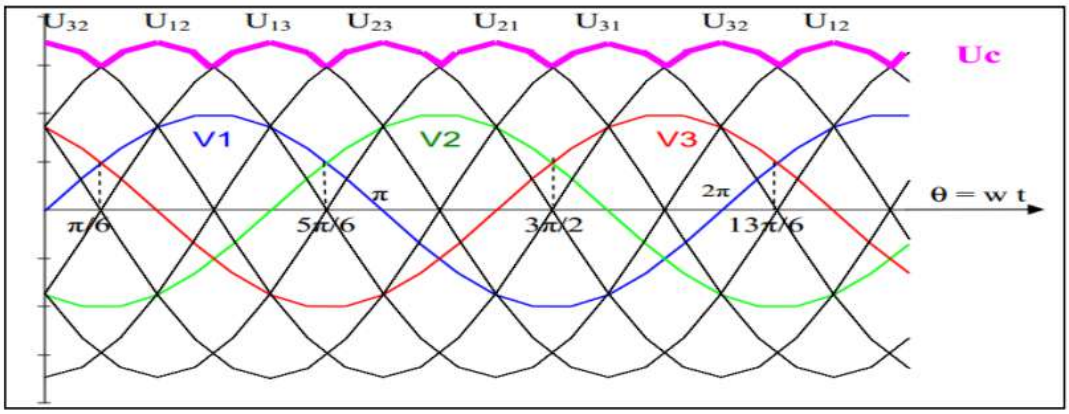


Figure 46 Tension redressée

14.3. Filtre

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

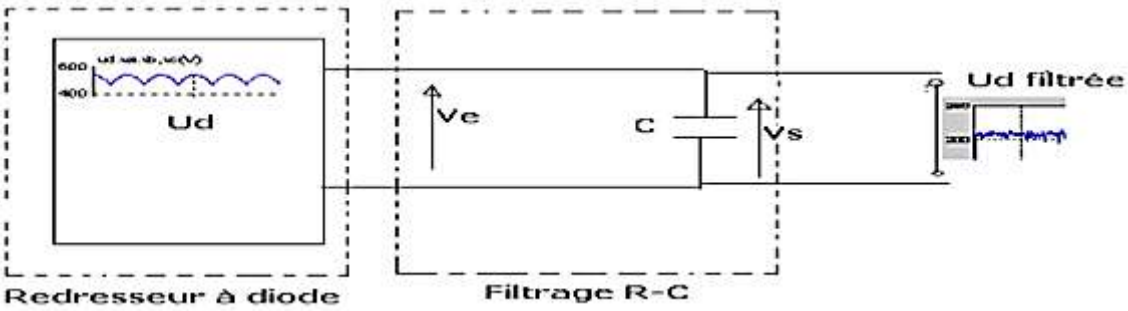


Figure 47 Schéma du principe de filtrage

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 suivants :

❖ 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. La capacité C_f du filtre correspond aussi généralement à des valeurs considérables, de plusieurs mF le cas échéant ; le condensateur utilisé peut appartenir au type électrolytique, à papier ou à film plastique. En appliquant loi des mailles au modèle de la figure on trouve :

$$U_d = R\, i_d + V_s$$

Avec :

$$i_d = \mathrm{c}\, \frac{d\, V_s}{dt}$$

14.4. L'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 [5].

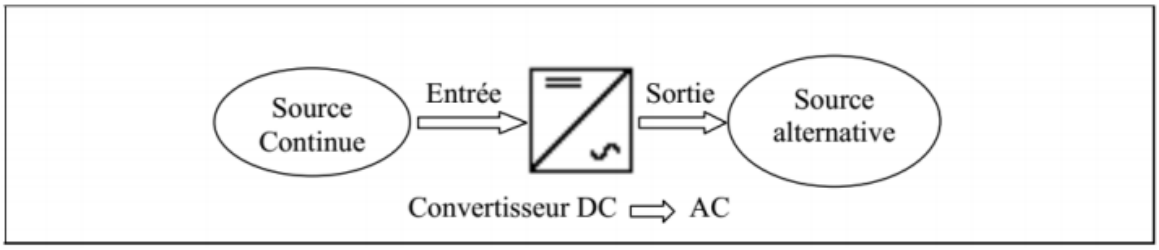


Figure 48 Schéma de principe de l'onduleur

14.4.1. Mode de fonctionnement

On distingue plusieurs modes de fonctionnement des onduleurs en fonction principalement du signal de sortie du circuit intermédiaire :

Le fonctionnement en modulation d'impulsion en amplitude *(PAM : Pulse Amplitude Modulation),*

Le fonctionnement en modulation de largeur d'impulsion *(PWM : Pulse Width Modulation).*

14.4.2. Modulation d'impulsion en amplitude

Ce type de modulation est utilisé lorsque le variateur de vitesse est à tension intermédiaire variable.

Comme on l'a vu dans le circuit intermédiaire :

Pour les variateurs avec redresseurs non-commandés, un hacheur est nécessaire pour générer une tension variable au niveau de l'onduleur.

Pour les variateurs avec redresseurs commandés, la variation de l'amplitude de la tension est générée par le redresseur lui-même.

Quel que soit le système, l'onduleur reçoit à son entrée une tension continue variable en amplitude. Dans ce cas, l'onduleur, lui, ne fait varier que la fréquence d'allumage et d'extinction des thyristor ou des transistors en fonction du niveau de la tension d'entrée pour recréer une tension sinusoïdale (dans le cas d'un moteur à courant alternatif).

14.4.3. Modulation de largeur d'impulsion (PWM) ou (MLI)

La modulation de largeur d'impulsion MLI est une technique pour générer une tension quasi sinusoïdale a partir d'un signal continue

Cette technique consiste à découper la tension d'alimentation afin d'obtenir un courant d'alimentation du moteur quasi sinusoïdal quelle que soit la fréquence d'alimentation

Le principe de commande de l'onduleur réside dans l'utilisation d'un comparateur de tensions. Ce comparateur superpose à trois tensions sinusoïdales de référence une tension de forme triangulaire. La fréquence des trois sinusoïdes de référence correspond à celle des tensions souhaitées à la sortie de l'onduleur. Les intersections entre les sinusoïdes et l'onde triangulaire déterminent l'allumage ou l'extinction des thyristors (ou des transistors de Puissance) selon le cas. Il en résulte un temps d'impulsion « passante » ou « non-passante » variable reconstituant un courant sinusoïdal en sortie du variateur de vitesse.

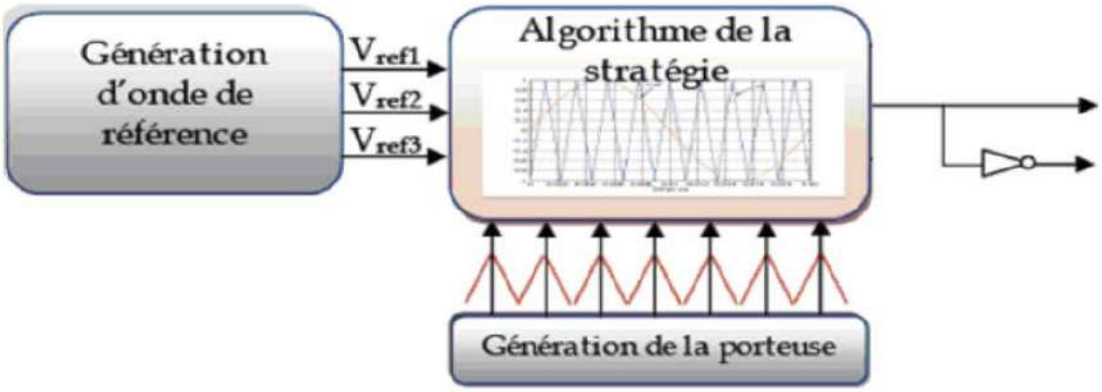


Figure 49 Principe de fonctionement

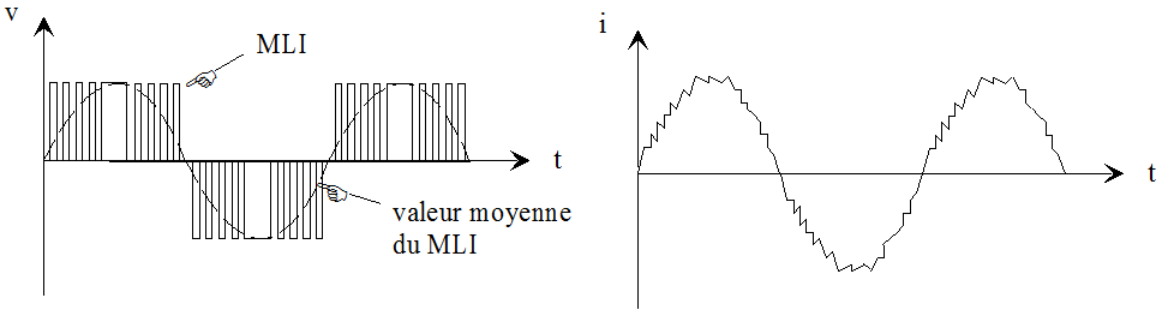


Figure 50 Commande MLI

15. Avantages et inconvénients d'un variateur de vitesse

15.1. Avantages

Le recours aux variateurs de vitesse offre plusieurs avantages :

- 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 entraine
- Diminution de la consommation d'électricité.
- Economies d'énergie.

15.2. Inconvénients

Tous les variateurs de vitesse intégrant des dispositifs de communication (diodes, thyristors, 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 résonnances harmoniques peuvent également apparaitre entre les variateurs de vitesse et les batteries des condensateurs.

Les effets néfastes peuvent se manifester par la défaillance prématurée des équipements électriques (surchauffe des moteurs, des câbles et des transformateurs) par la dégradation de l'isolation des moteurs commandes ou par l'interruption des précèdes (fusibles brules).

16. Choix d'un variateur de vitesse

Aujourd'hui, faire le bon choix d'un variateur, ne passe plus uniquement par un choix « électrique » mais par une bonne adéquation avec l'application de l'utilisateur. Le variateur électronique de vitesse se trouve au cœur de l'application et synthétise de nombreux services. Les principaux points habituels qui interviennent dans le choix d'un variateur de vitesse sont :

- Le coût de l'équipement.
- Le type de moteur entraîné
- Les caractéristiques du moteur : il s'agit notamment de la puissance, de la tension et de la vitesse de rotation. Ces trois informations permettent de choisir le calibre du variateur. Un variateur sous-dimensionné ne pourra pas atteindre la fréquence nominale requise. Surdimensionné à l'excès il induira une certaine instabilité du moteur et parfois une perte de phase.
- Le type d'application : en prêtant attention à l'application, il est possible de choisir la gamme de variateur. Par exemple, les applications de type processus industriel, manutention simple ou emballage impliquent la production d'un couple constant. Il s'agit de réalisations effectuées par des moteurs de puissance généralement inférieures à 15 kW avec des variateurs relativement standards. Entre 15 kW et 1 MW, ces applications demandent la mise en œuvre de variateurs relativement évolués.

- Le lieu d'installation : Si le variateur est placé dans un environnement poussiéreux par exemple, il faut utiliser des variateurs avec un boitier. Et s'il est placé dans un toit il lui en utiliser un avec une protection contre les UV.

17. Altivar12HU15M2 :

ALTIVAR 12 est le variateur propose pour se système à cause de sa compatibilité avec la puissance de notre moteur asynchrone et son adaptation avec tout type de réseau électrique.

La famille ALTIVAR 12 de variateurs de vitesse a fréquence réglable est utilisée pour la commande des moteurs asynchrones triphasés.[17]



Figure 51 Altivar12HU15M2

17.1. Schéma de câblage général

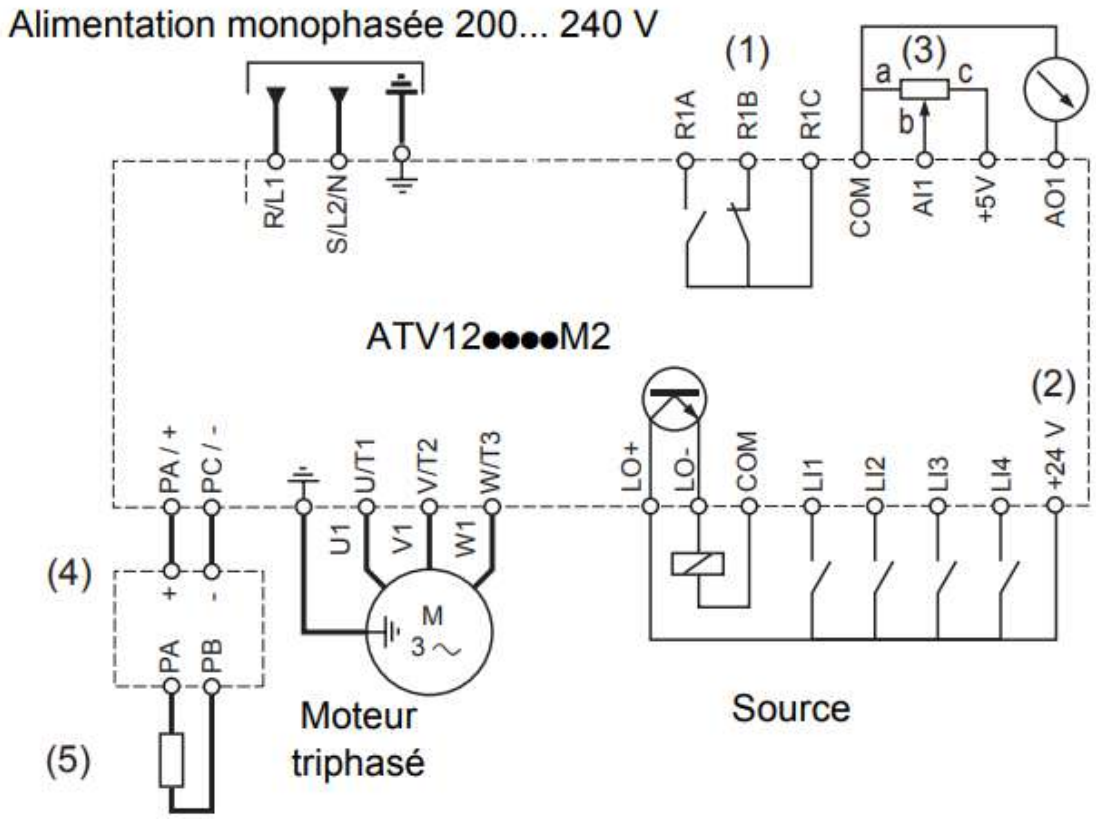


Figure 52 Schéma de câblage général

(1) Contacts de relais R1, pour signalisation à distance de l'état du variateur.

(2) Interne + 24 V c. Si une source externe est utilisée (+ 30 V au maximum), connectez le 0 V de la source sur la borne COM. N'utilisez pas la borne + 24 V c sur le variateur.

(3) Potentiomètre de référence SZ1RV1202 (2,2 kΩ) ou similaire (10 kΩ maximum).

(4) Module de freinage optionnel VW3A7005

(5) Résistance de freinage optionnelle VW3A7ppp ou autre résistance acceptable. Consultez les valeurs possibles de la résistance dans le catalogue.[17]

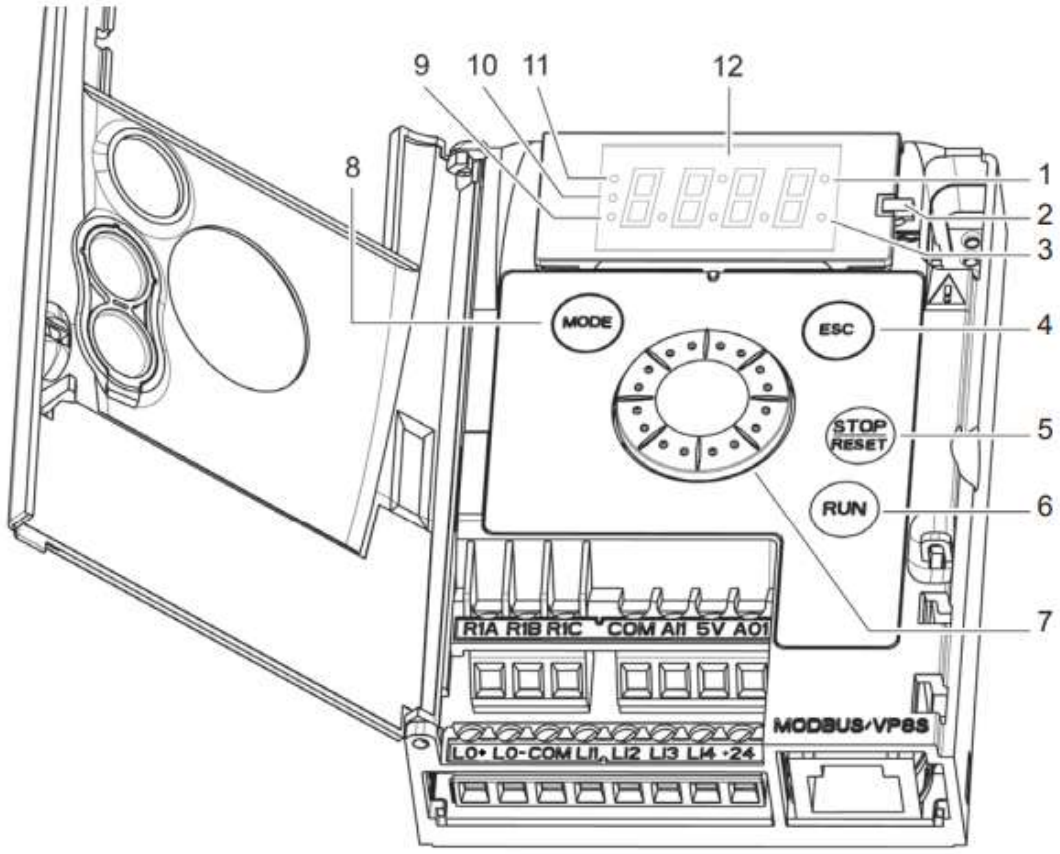


Figure 53 interface utilisateur

1. DEL de valeur (a) (b).

2. DEL de charge

3. DEL d'unité (c)

4. Bouton ESC : permet de quitter un menu ou un paramètre ou de mettre fin à la valeur affichée afin de revenir à la valeur précédente se trouvant en mémoire. En configuration locale, une pression de 2 s sur le bouton ESC commute entre les modes contrôle et programmation. 5. Bouton STOP/RESET : arrête le moteur (peut être dissimulé par la porte si la fonction est désactivée). Nota : voir les instructions sur le retrait du cache de « RUN/STOP ». Permet de réarmer un défaut.

6. Bouton RUN : lance l'exécution en configuration locale et en configuration à distance si la fonction est configurée (peut être dissimulé par la porte si la fonction est désactivée).

7. Bouton de navigation - Agit comme un potentiomètre en configuration locale et en configuration à distance si la fonction est configurée. - Molette servant à la navigation lorsqu'elle est tournée dans le sens horaire ou anti-horaire. - et à la sélection / validation par simple pression. Cette action est représentée par ce symbole [17]

8. Bouton MODE Permet de passer d'un des modes Référence, Surveillance ou Programmation à un autre. Une pression de 3 s sur le bouton MODE commute entre les configurations locales et à distance.

9. DEL du mode CONFIGURATION (b) 10. DEL du mode SURVEILLANCE 11. DEL du mode REFERENCE 12. Afficheur 4 fois 7

17.2. Bornes de contrôle

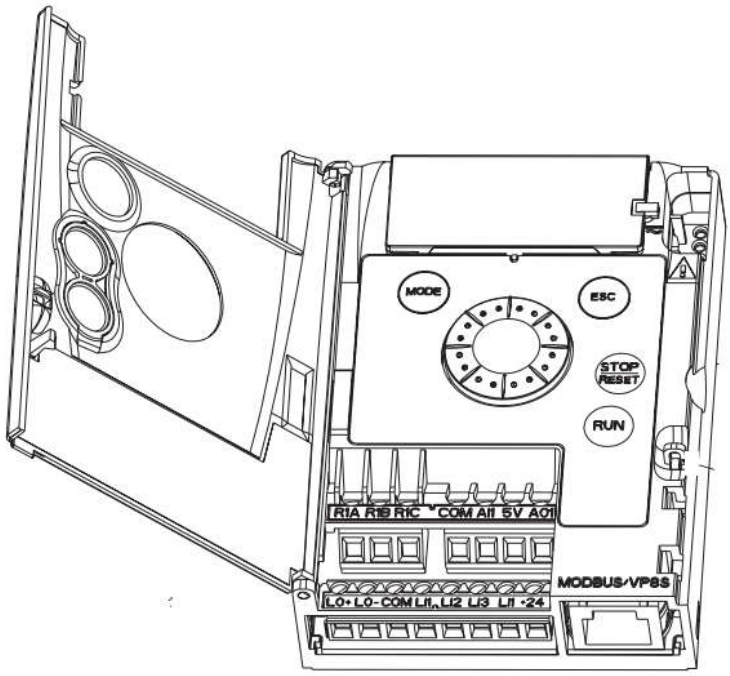


Figure 54 interface utilisateur

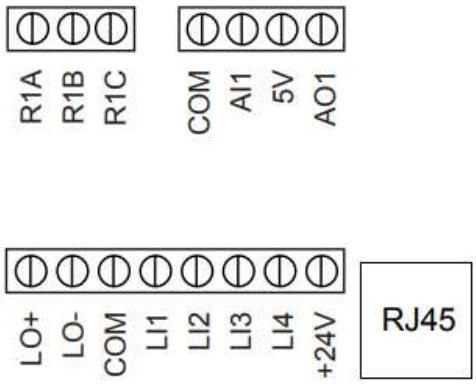


Figure 55 schéma des entrées/sorties

R1A Contact du relais Normalement Ouvert (NO)

R1B Contact du relais Normalement Fermé (NC)

R1C Borne commune du relais

COM Commun des E/S analogiques et logiques

AI1 Entrée analogique

5V Alimentation +5 Vdc fournie par le variateur

AO1 Sortie analogique

LO+ Sortie logique (collecteur)

LO- Commun pour la sortie logique (émetteur)

COM Commun des E/S analogiques et logiques

LI1 Entrée logique Entrée logique

LI2 Entrée logique Entrée logique

LI3 Entrée logique Entrée logique

LI 4 Entrée logique Entrée logique

+ 24V Alimentation +24 Vdc fournie par le variateur Connexion pour logiciel SoMove, réseau Modbus ou afficheur déporté.[17]

18. Conclusion

Dans ce chapitre on a donné une étude générale sur les moteurs asynchrones triphasés et leur constitution avec le principe de fonctionnement, ainsi les diffèrent actions pour varier la vitesse de synchronisme. Pour cela on a utilisé le variateur de vitesse ATV12 qui converti le signal alternatif d’une fréquence fix en un signal alternatif de fréquence variable.

CHAPITRE III : REALISATION

1. INTRODUCTION

Notre but est le contrôle d'un moteur asynchrone par l'automate programmable s7 300 via le variateur de vitesse ATV12, Le moteur est supervisé par un écran tactile, des boutons poussoirs, et des voyants. Dans ce chapitre nous allons établir la communication entre tous les dispositifs et faire la configuration pour le variateur de vitesse et l'IHM en suite le pilotage par l'automate programmable et les sorties TOR/Analogique.

2. Schéma structurelle di convoyeur

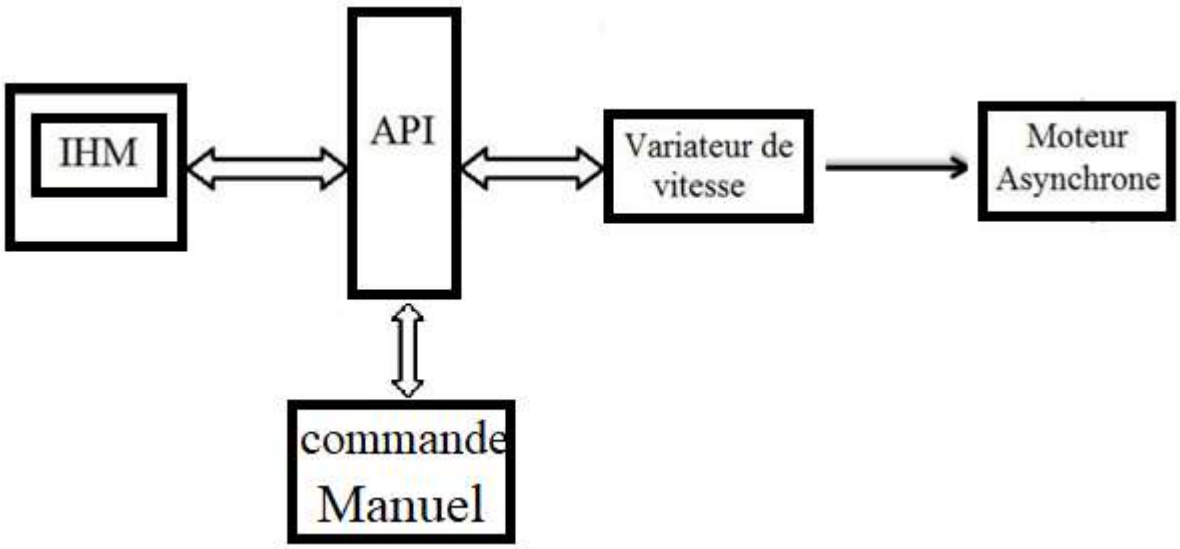


Figure 56 Schéma structurelle

L'écran tactile est lié l'automate, cette liaison permet à l'écran de recevoir des informations depuis l'automate et donner des ordres a ce dernier via le Bus de communication ?PI L'automate est relié au variateur de vitesse via les entrée TOR ou l'automate est le maitre, 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. Et cette variation est due aux consignes de l'automate. Donc c'est l'automate qui ordonne la modification de la vitesse et du sens de rotation du moteur. Tout ce qui se passe est supervisé par l'écran tactile ce qui veut dire, que l'écran affiche le sens rotation et la vitesse, et permet de commander le variateur à distance.

3. Configuration des éléments utilisée

3.1. Réglages usine du variateur

L'Altivar 12 est réglé en usine pour les conditions de fonctionnement les plus courantes :

• Afficheur : variateur prêt (RDY) moteur arrêté ou référence de fréquence du moteur en fonctionnement.

• Adaptation automatique de la rampe de décélération en cas de surtension au freinage.

• Pas de redémarrage automatique après la suppression d'un défaut détecté.

• Entrées logiques : LI1 : Sens avant

• Entrée analogique : référence de vitesse AI1 (0 à + 5 V)

• Relais R1 : le contact s'ouvre en cas de défaut détecté (ou variateur hors tension) [17]

3.2. Réglage des paramètres du moteur

Après la Consultation de la plaque signalétique du moteur pour définir les paramètres suivants.

Menu	Code	Description
CONF > FULL > drC-	bFr	Fréquence standard du moteur
	nPr	Puissance nominale du moteur donnée sur la plaque signalétique.
	nCr	Courant nominal du moteur sur la plaque signalétique du moteur (A).
CONF > FULL > FLt- > tHt-	ItH	Courant nominal indiqué sur la plaque signalétique du moteur (A).

Figure 57 configuration

3.3. Définition des paramètres de base

Menu	Code	Description	Réglage usine
CONF	ACC	Temps d'accélération (s)	3.0
	dEC	Temps de décélération (s)	3.0
	LSP	Fréquence du moteur à la référence minimum (Hz)	0.0
	HSP	Fréquence du moteur à la référence maximum (Hz)	50.0

Figure 58 Paramètre de base

3.4. *Définition des paramètres de contrôle*

Menu	Code	Description	[Configuration à distance]	[Configuration Locale]
CONF > FULL > Ctl-	Fr1	Contrôle de référence	AI1	AIU1

Figure 59 paramètre de contrôle

3.5. *Câblage de la commande choix de la configuration*

3.5.1. *Câblage de la vitesse de référence*

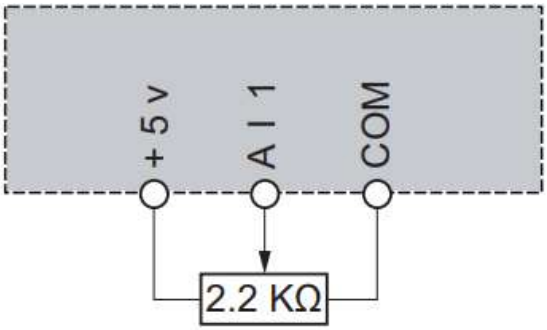


Figure 60 Câblage de potentiomètre

3.5.2. *Câblage du contrôle :*

En commande deux fils : Paramètre tCC = 2C

Voici le schéma de câblage

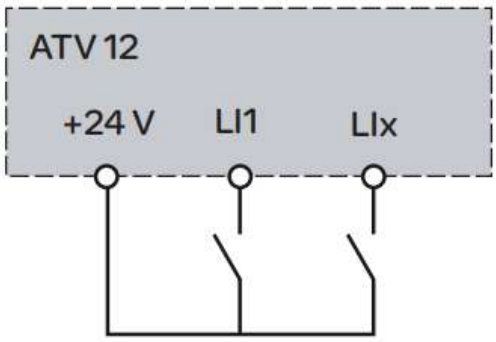


Figure 61 schéma de câblage de control 2 fils

LI1 : pour tourner en avant

LIx : pour tourner en sens inverse

En commande trois fils : Paramètre tCC = 3C

Voici son Schéma de câblage

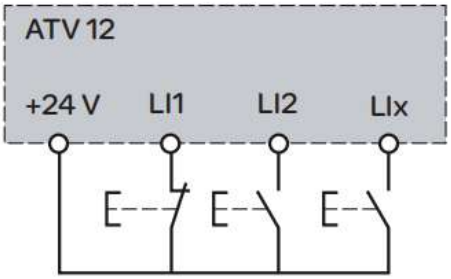


Figure 62 schéma de câblage de control 3 fils

LI1 : arrêt

LI2 : avant

LIx : arrière

Pour diminuer ou augmenter la vitesse, les rampes d'accélération et de décélération on peut modifier les paramètres :

- LSP : petite vitesse,
- HSP : grande vitesse,
- ACC : accélération,
- DEC : décélération,

4. Moteur Asynchrone

4.1. Plaque signalétique du moteur utilisé

Moteur	Asynchrone
Type	L 60
Triphasé	50Hz
Cos Phi	0.67
N° de série	348227
CH	0,5
Tr/mn	925

Tension étoile	380V
Ampérage étoile	1.3A
Tension Triphasé	220V
Ampérage Triphasé	2.3A

Tableau 1 Plaque signalétique

- Puissance 0.37 KW puissance utile délivrée sur l'arbre du moteur car 1HP = 745.7 Watt

5. Les Boutons poussoirs et les voyants

5.1. Les Boutons poussoirs

Il en existe de deux types, les boutons poussoirs à fermeture et les boutons poussoirs à ouverture. Ils servent à ouvrir ou fermer un circuit électrique, dès qu'on les relâche ils reviennent dans leur position initiale.



Figure 63 Bouton poussoir

5.2. Symbolisation

Symbole bouton poussoir normalement fermé (NC)

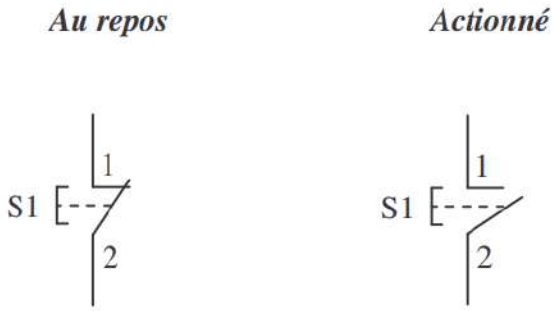


Figure 64 symbole de bouton poussoir NC

Symbole bouton poussoir normalement ouvert

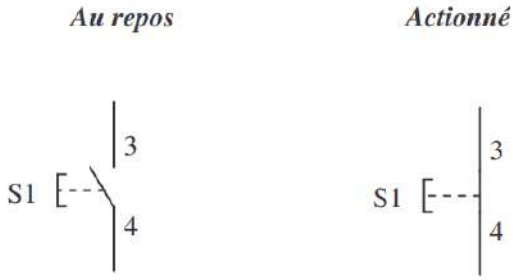


Figure 65 Symbole bouton poussoir normalement ouvert

5.3. Fonctionnement

Bouton poussoir NO

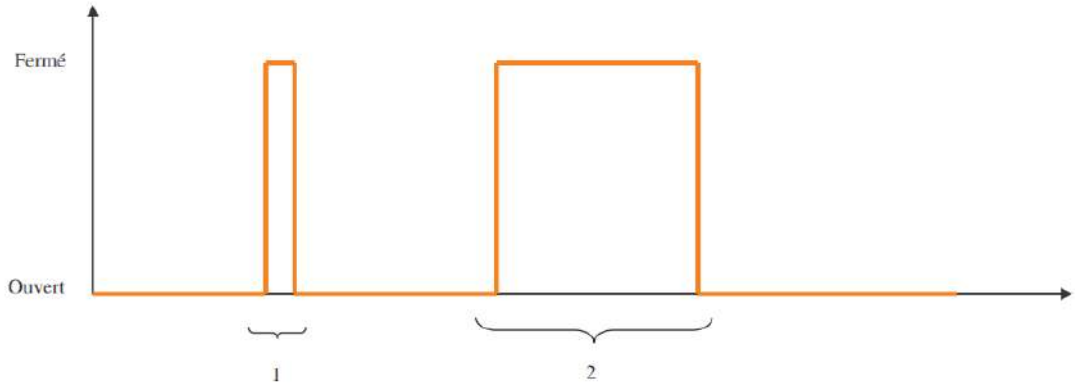


Figure 66 Fonctionnement NO

1 L'opérateur appuie sur le bouton poussoir et le relâche aussitôt

2 L'opérateur appuie sur le bouton poussoir et reste appuyer un certain temp

Bouton poussoir NC

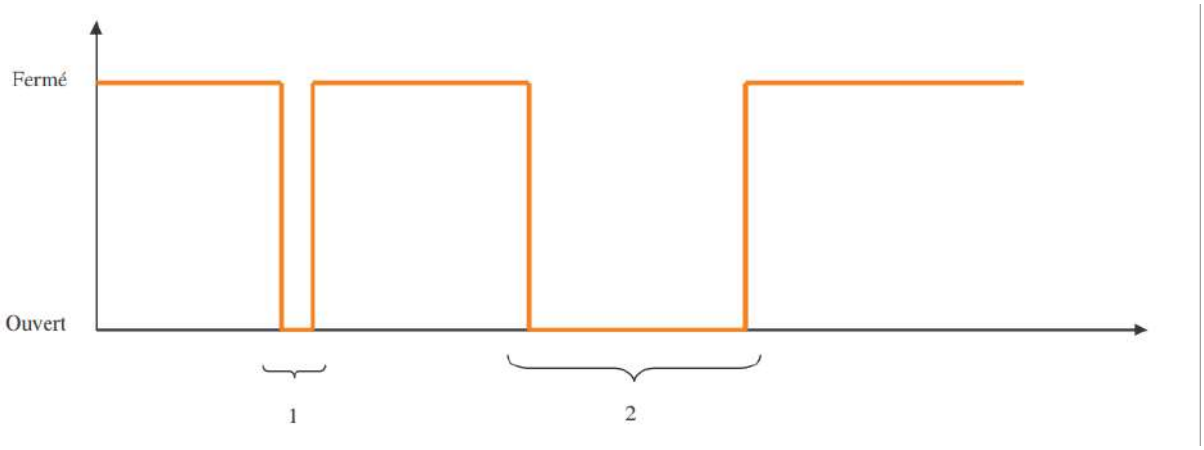


Figure 67 Fonctionnement NC

1 L'opérateur appuie sur le bouton poussoir et le relâche aussitôt

2 L'opérateur appuie sur le bouton poussoir et reste appuyer un certain temp

5.4. Les Boutons coup de poing

Ils servent à ouvrir un circuit électrique, lorsqu'ils sont actionnés ils restent dans leur position.



Figure 68 Bouton coup de poing

5.4.1. Symbolisation



Figure 69 Schéma d'un bouton coup de poing

5.4.2. Fonctionnement



Figure 70 Fonctionnement d'un boutons coup de poing

L'opérateur appuie sur l'arrêt d'urgence et donc ouvre le contact on constate qu'il reste dans cette position ouverte

Si on veut faire revenir le bouton d'arrêt d'urgence dans position initiale il faut à nouveau l'actionner la tète

5.5. Les voyants

Ils servent à donner une information sur l'état d'un système.



Figure 71 Voyant 24V

5.5.1. Symbolisation

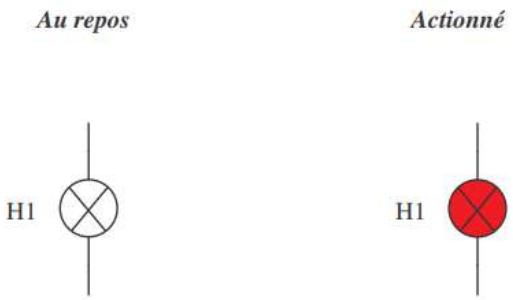


Figure 72 Symbole d'un voyant

5.5.2. Fonctionnement

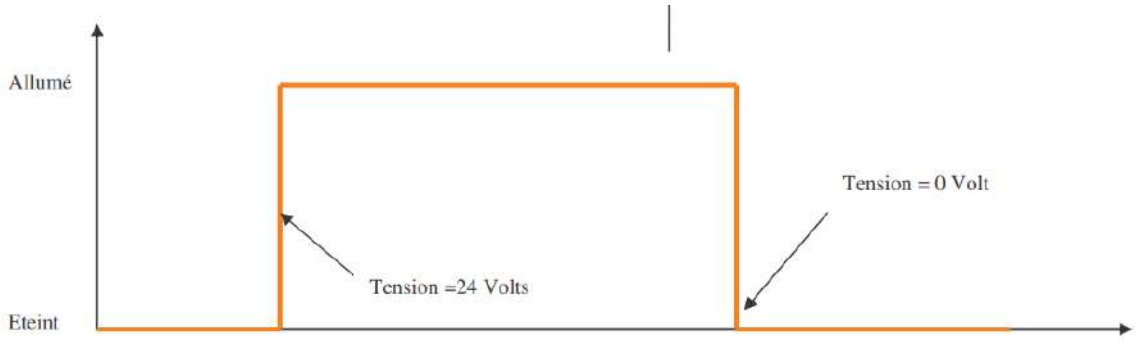


Figure 73 Fonctionnement d'un voyant

Si on le voyant qui dans un circuit est soumis a une tension inférieure ou égale a 25V Volts le voyant s'allume

6. HMI TP 177B

6.1. Présentation générale du produit :

Les pupitres de la série 177 sont issus des pupitres opérateurs déjà connus de la série 170 et présentent de nouveaux développements. Les nouveaux pupitres TP 177A, TP 177B et OP 177B permettent d'utiliser de manière encore plus efficace des projets à base de textes ou de graphiques pour des tâches simples et moyennes de contrôle-commande sur des machines et des installations. [18]

6.2. Structure du pupitre opérateur TP 177B 6"

Vue de face et vue de profile :

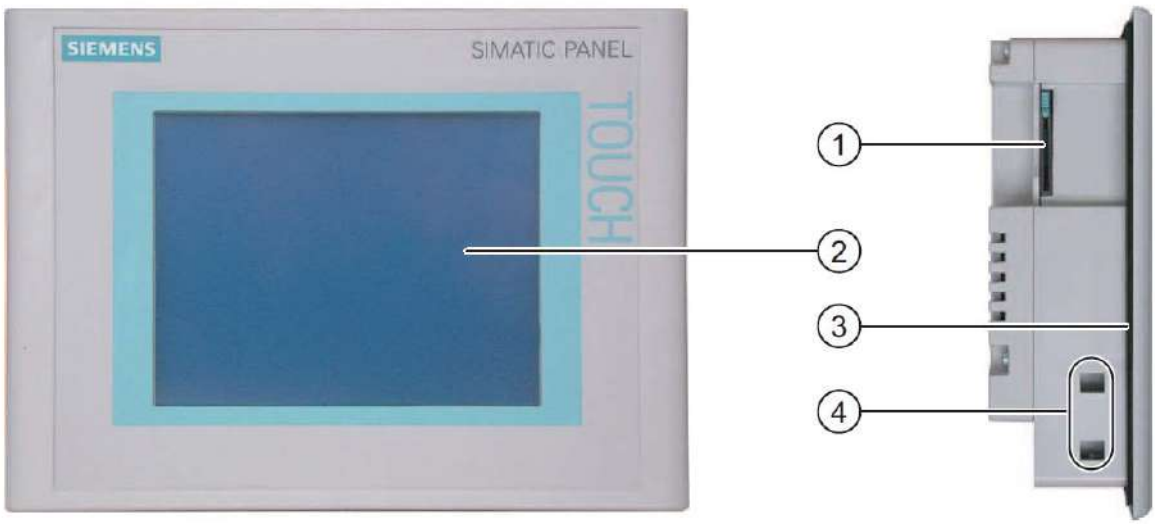


Figure 74 Vue de face et vue de profile HMI177b

1. Emplacement pour une MultiMediaCard
2. Ecran tactile
3. Joint de montage
4. Echancrure pour bride de fixation

6.3. Vue de dessous



Figure 75 Vue de dessous HMI177b

1. Contact de masse pour la liaison équipotentielle
2. Raccord pour l'alimentation 24V
3. Interface RS 422/485 (IF 1B)
4. Connecteur PROFINET (valable pour le TP 177B 6" PN/DP)
5. Connecteur USB

6.4. Vue arrière

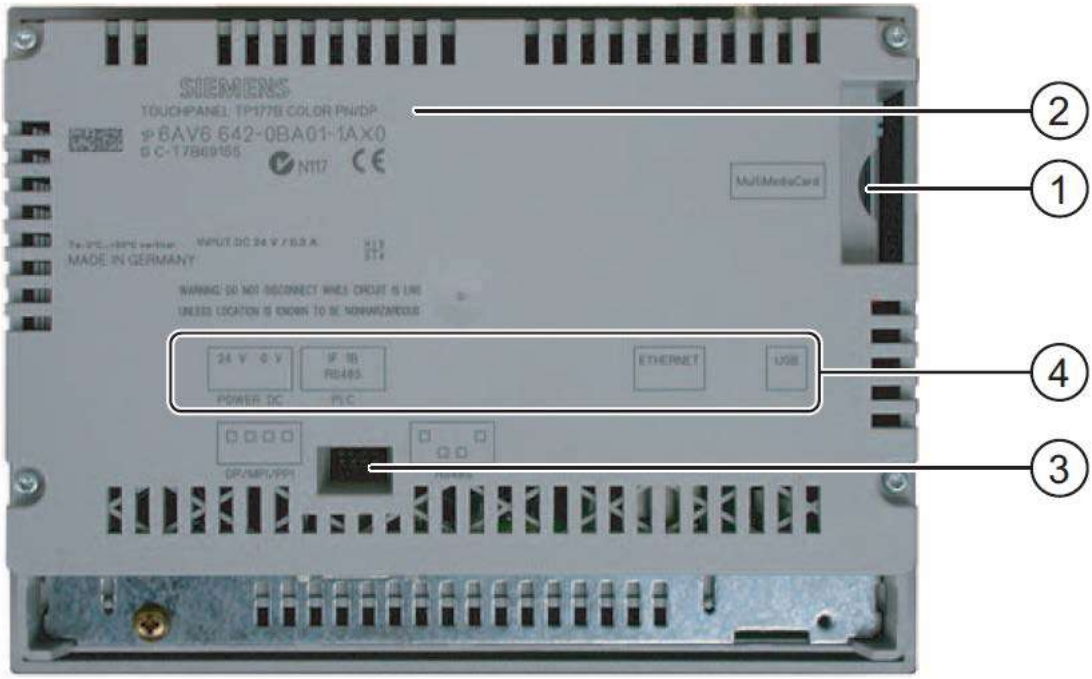


Figure 76 Vue arrière HMI177b

1. Emplacement pour une MultiMediaCard
2. Plaque signalétique
3. Commutateur multiple
4. Désignation de l'interface

6.5. Raccordement alimentation

La figure suivante montre le raccordement entre le pupitre opérateur et l'alimentation électrique.

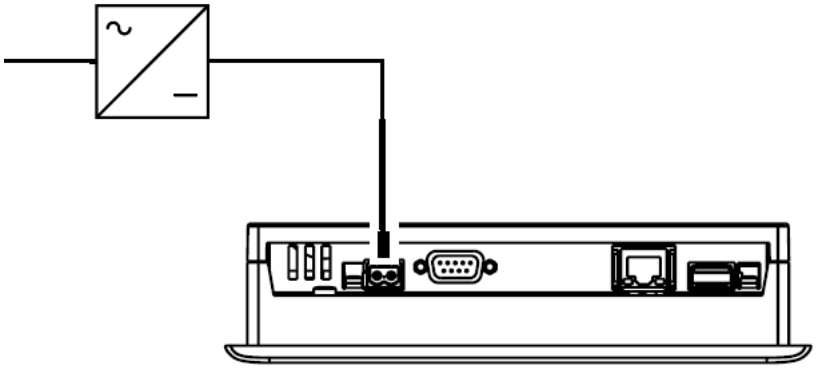


Figure 77 schéma du câblage électrique

6.6. Mettre en marche et tester le pupitre opérateur

Marche à suivre

Procédez comme suit :

1. Connectez le bornier au pupitre opérateur.

2. Mettez l'alimentation sous tension.

A la mise sous tension de l'alimentation, l'écran s'allume. Durant le démarrage, une barre de progression s'affiche.

Si le pupitre opérateur ne démarre pas, il se peut que les fils aient été inversés au niveau du bornier. Vérifiez les fils branchés et modifiez leur branchement, le cas échéant. Après le démarrage du système d'exploitation, le programme de chargement ("loader") s'affiche. [18]

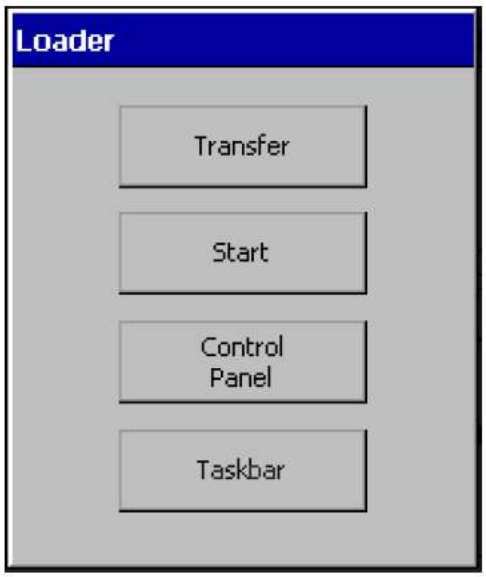


Figure 78 Menu Loader

Lors de la première mise en service, le pupitre opérateur passe automatiquement

Au mode de fonctionnement "Transfer" :

– Aucun projet ne se trouve sur le pupitre opérateur.

– Un canal de données au moins est paramétré.

Le dialogue suivant s'affiche alors :

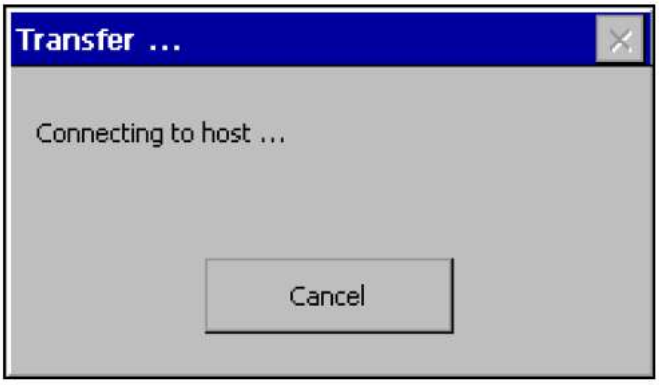


Figure 79 Menu Transfer

3. Pour annuler le transfert, appuyez sur le bouton "Cancel".

7. Raccordements des différents éléments

7.1. Raccordement du variateur de vitesse avec l'API

On a utilisé les deux sorties de l'api DQ124.0 et DQ124.1 pour contrôler le sense de rotation et la sortie analogique AQ752 pour changer la vitesse.

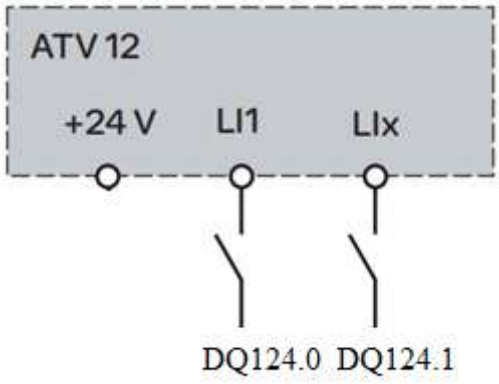


Figure 80 Schéma du raccordement TOR ATV12/API

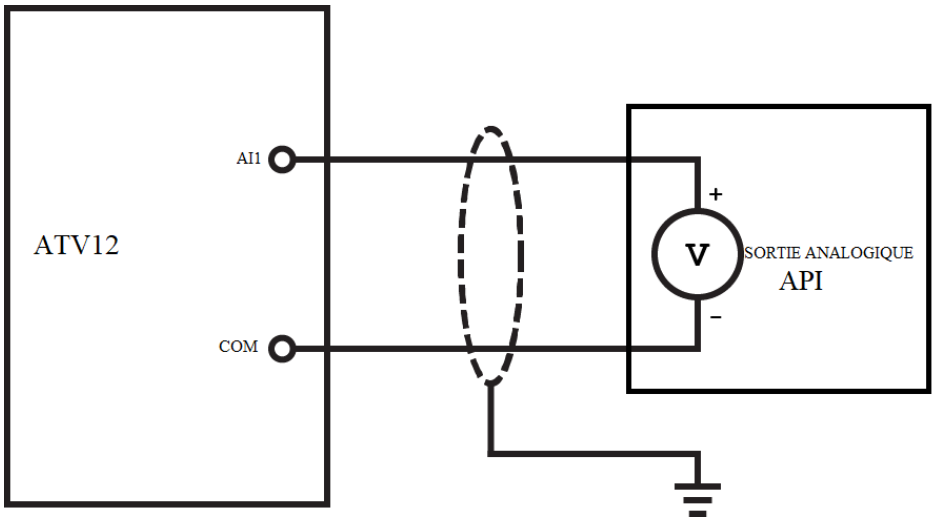


Figure 81 Schéma du raccordement analogique ATV12/API

7.2. Raccordement du variateur de vitesse avec le moteur asynchrone

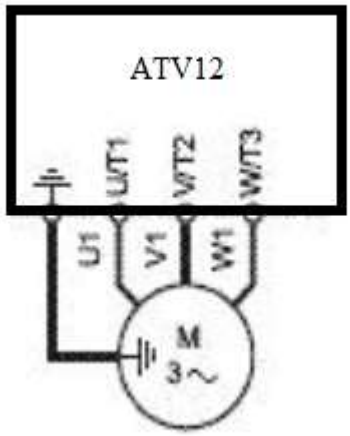
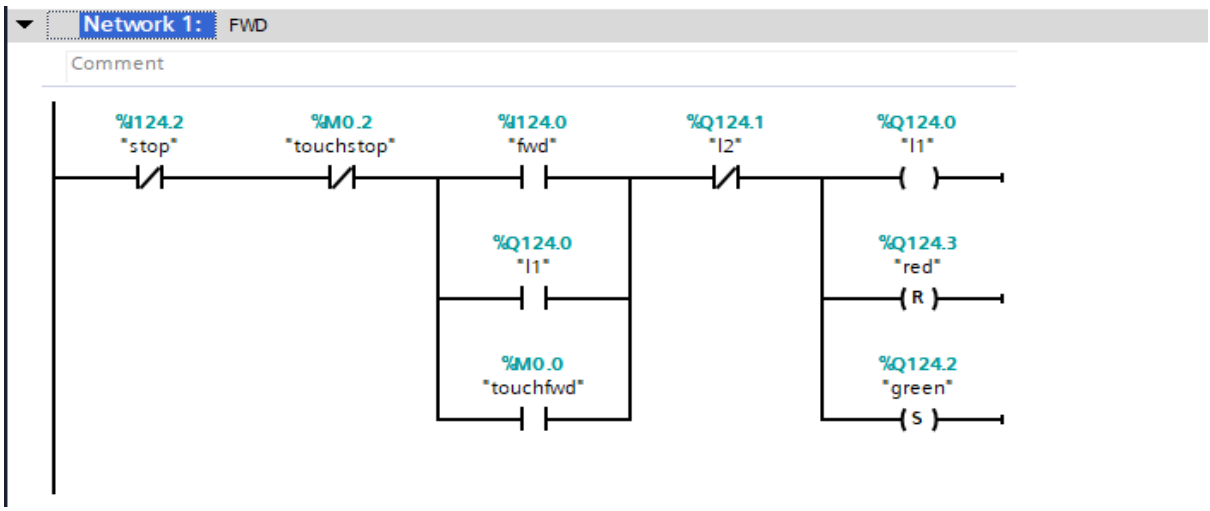
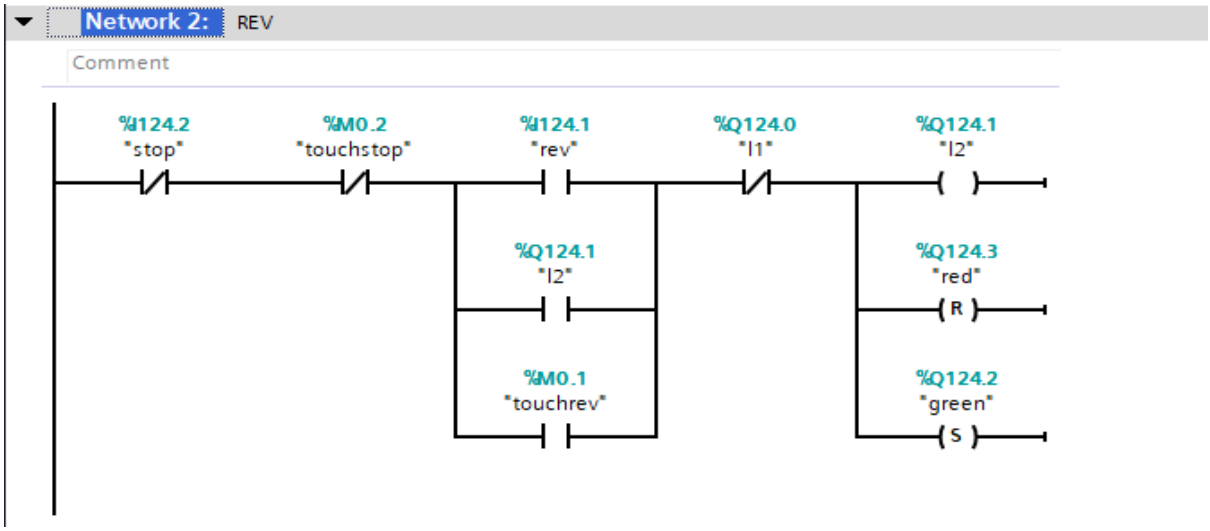


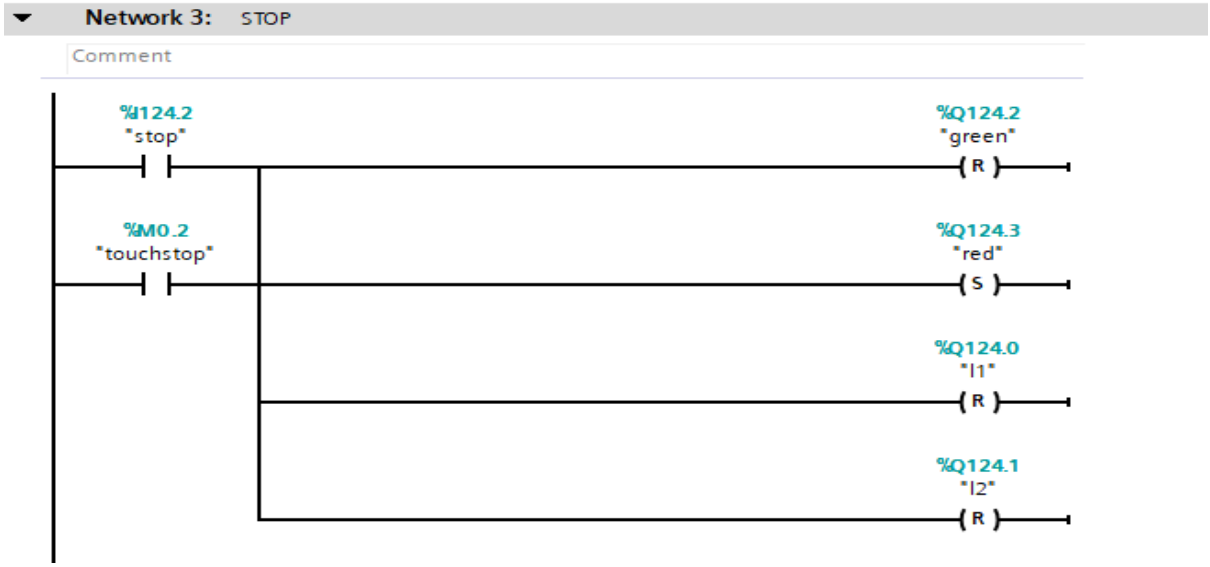
Figure 82 Schéma du câblage moteur asynchrone avec ATV12

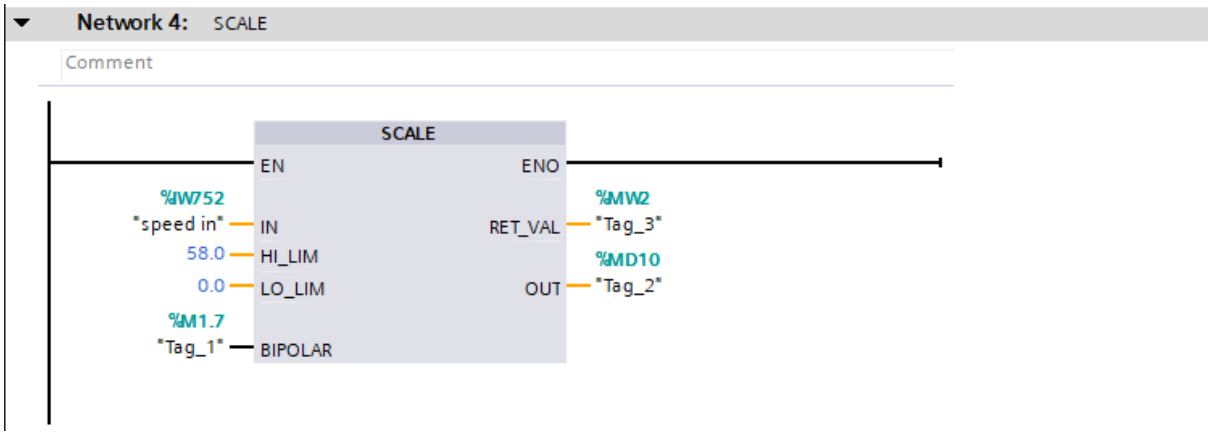
8. Pilotage du variateur de vitesse ATV312 avec l'API

Voici le programme Ladder qu'on a utilisé pour le control du moteur asynchrone :




Network 2: REV
Comment
%I124.2
"stop"
%M0.2
"touchstop"
%I124.1
"rev"
%Q124.0
"I1"
%Q124.1
"I2"
%Q124.1
"I2"
%Q124.3
"red"
R
%M0.1
"touchrev"
%Q124.2
"green"
S


Network 3: STOP
Comment
%I124.2
"stop"
%Q124.2
"green"
R
%M0.2
"touchstop"
%Q124.3
"red"
S
%Q124.0
"I1"
R
%Q124.1
"I2"
R


Network 4: SCALE
Comment
SCALE
EN
ENO
%IW752
"speed in"
IN
RET_VAL
%MW2
"Tag_3"
58.0
HI_LIM
%MD10
0.0
LO_LIM
OUT
"Tag_2"
%M1.7
"Tag_1"
BIPOLAR

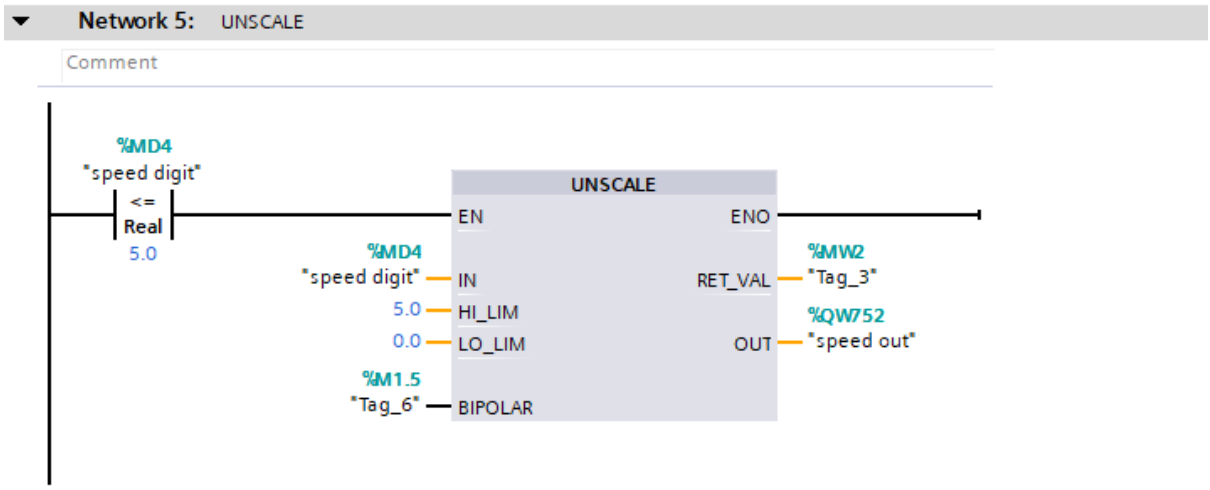


Figure 83 Le code ladder utiliser pour le control du moteur

8.1. Explication des variables utiliser avec leur adresse

Nom de variable	Description
Fwd	Variable d'entrée pour faire tourner le moteur en sens avant avec bouton poussoir son adresse est I124, le bit 0
Rev	Variable d'entrée pour faire tourner le moteur en sens inverse avec bouton poussoir son adresse est I124, le bit 1
Stop	Variable d'entrée pour faire arreter le moteur avec bouton arrêt d'urgence son adresse est I124, le bit 2
Touchfwd	Variable d'entrée memoire pour faire tourner le moteur en sens avant avec HMI son adresse est 0, le bit 0
Touchrev	Variable d'entrée memoire pour faire tourner le moteur en sens avant avec HMI son adresse est 0, le bit 0
Touchstop	Variable d'entrée memoire pour faire tourner le moteur en sens avant avec HMI son adresse est 0, le bit 0
L1	Variable de sortie pour faire tourner le moteur en sens avant son adresse est Q124, le bit 0
L2	Variable de sortie pour faire tourner le moteur en sens avant son adresse est Q124, le bit 1
Green	Variable de sortie pour faire allumer le voyant vert quand le moteur est en marche son adresse est Q124, le bit 2

Red	Variable de sortie pour faire allumer le voyant rouge quand le moteur n'est pas en marche son adresse est Q124, le bit 3
Speedin	Variable d'entrée analogique pour faire afficher la vitesse dans l'HMI I752
speedigit	Variable d'entrée et une condition pour que la valeur soi inferieur de 5V dans la fonction scale
speedout	Variable de sortie analogique pour varie la vitesse du moteur via HMI son adresse Q752
Tag1	Valeur bipolaire pour la fonction scale et unscale
Tag2	Variable de sortie pour afficher la vitesse dans HMI
Tag3	Variable pour retourner une valeur en cas derreur dexecution du fonction

Tableau 2 Liste des variables

8.2. Explication de chaque réseau

8.2.1. Network 1

Le réseau 1 est simplement pour faire tourner le moteur en sens avant en appuyant soi sur le bouton poussoir fwd soi sur le bouton touchfwd dans l'écran HMI a condition que le bouton d'urgence ne soi pas ouvert et le moteur n'est pas entrain de tourner en sens inverse.

8.2.2. Network 2

Le réseau 2 est pour faire tourner le moteur en sens inverse en appuyant soi sur le bouton poussoir inv soi sur le bouton touchinv dans l'écran HMI à condition que le bouton d'urgence ne soit pas ouvert et le moteur n'est pas en train de tourner en sens avant.

8.2.3. Network 3

Le réseau est le programme pour faire arrêter le moteur quand le bouton d'urgence soi ouvert, le voyant rouge s'allume et le voyant vert s'éteint.

8.2.4. Network 4

Le réseau 4 est seulement pour faire convertir l'entrée analogique a une valeur numérique pour faire afficher la vitesse a l'écran HMI en utilisant la fonction SCALE

8.2.5. Network 5

Le réseau 5 est pour faire convertir une entrée numérique a une sortie analogique pour contrôler la vitesse du moteur avec l'écran HMI avec une condition que la valeur d'entrée doit être inferieur de l'intervalle 0 à 5

9. Supervision

9.1. Le menu Start

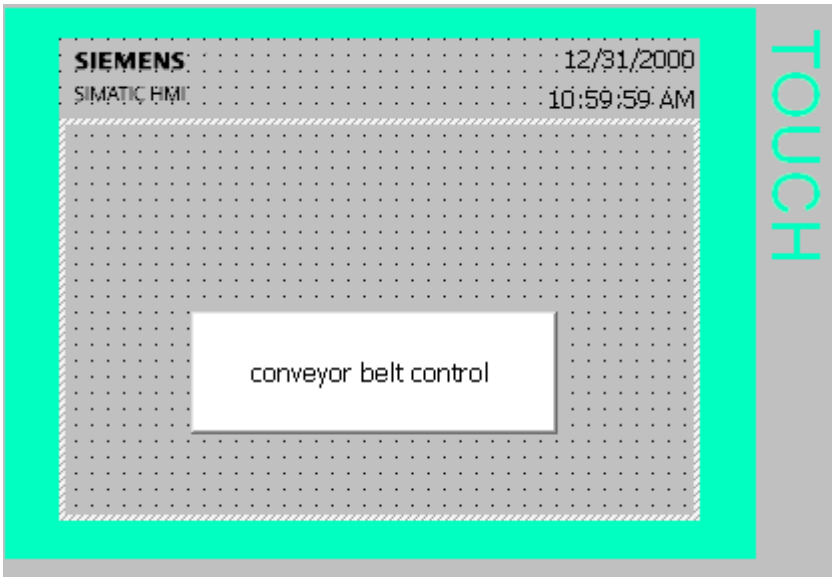


Figure 84 Le menu start

9.2. Le menu Main

Sur cet écran nous pouvons envoyer une consigne de changement de sens et de vitesse au moteur, afficher la vitesse avec laquelle il tourne, voir l'état du moteur (marche ou arrêt)

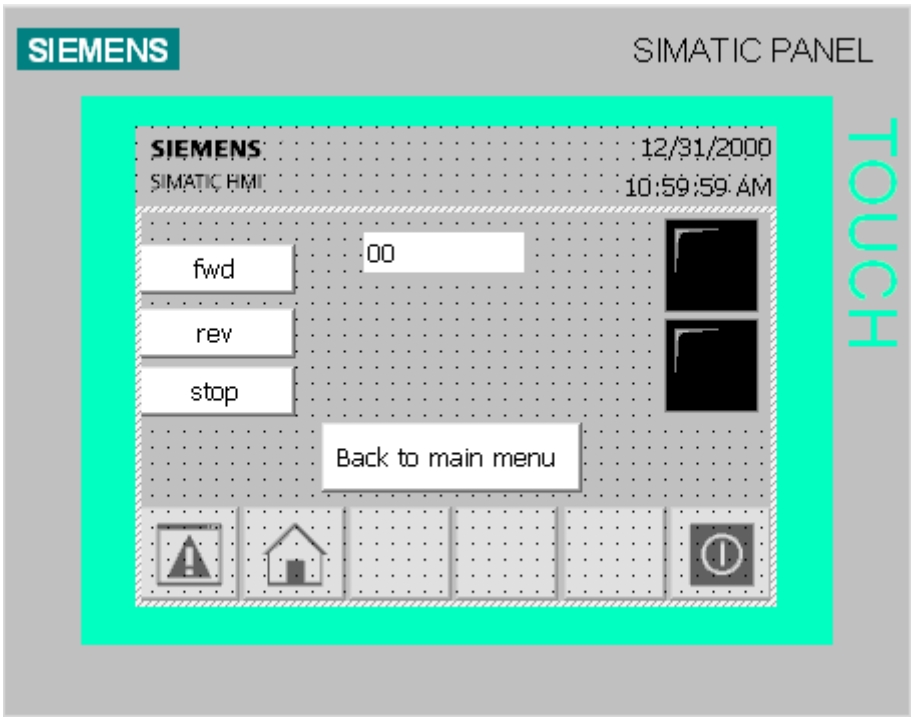


Figure 85 Le menu main

10. Conclusion

Dans ce chapitre, on montrer la configuration de variateur de vitesse et le câblage utilisés pour la réalisation. Ensuite nous avons parlé de la partie commande de notre système ou nous avons expliqué comment se fait le pilotage du variateur de vitesse, en utilisant les deux entrée logique LI1 et LI2. Nous sommes parvenus à commander le variateur de vitesse avec l'API. On a eu des difficultés pour changer la vitesse via l'écran car le voltage de la sortie du l'automate programmable est 24V et le voltage d'entrée analogique du variateur est 5 ou 10V.

CONCLUSION GENERALE

Notre projet nous a permis de connaitre le fonctionnement d'un moteur asynchrone et ses différentes méthodes pour varier sa vitesse, pour cela nous avons utilisé un variateur de vitesse, chose qui nous a permis de connaitre le fonctionnement de ce dernier et comment l`utiliser pour varier la fréquence. Enfin nous avons utilisé l`API S7 300 pour commander le moteur asynchrone via le variateur de vitesse aussi, nous avons utilisé des boutons poussoirs et une interface HMI pour superviser le comportement du moteur.

Nous avons réalisé tout le programme de command à l`aide du logiciel TIA PORTAL 13 avec le langage LADDER,

Notre programme LADDER a été simuler à l`aide du logiciel PLCSIM pour confirmer le bon fonctionnement de notre programme.

La réalisation de ce projet nous a permis aussi de faire la configuration du variateur de vitesse suite à la plaque signalétique du moteur, ainsi que le câblage du dispositif et la configuration de l`IHM.

Nous espérons que notre travaille pourra servir de support pour les prochains étudiants, qui veulent découvrir ces technologies, notamment le circuit API – variateur de vitesse, la programmation et l'utilisation du logiciel TIA PORTAL.

BIBLIOGRAPHIE ET WEBOGRAPHIE

[1] A. Simon, « Automates Programmable », L'ELANE, 1983.

[2] Informatique-automate-programmable
https://www.futura-sciences.com/tech/definitions/informatique-automateprogrammable-10525/ Apparue le 3 juin 2019

[3] https://www.automation-sense.com/blog/automatisme/pg-pc-et- Apparue le 3 juin 2019

[4]S7-300 CPU 31xC et CPU 31x : Installation et configuration
configuration-des-interfaces-pg-pc-sous-step7-siemens.html Apparue le 4 juin 2019

[5] Peter Cleaveland, « AC Adjustable Speed Drives », Control Engineering, Reed Business Information,1er novembre 2007

[6] CTIA06 - Programmation des automates S7-300 - Les blocs d'organisation (OB).pdf Apparue le 4 juin 2019

[7] CTIA10 - Introduction à la programmation Grafcet sous Siemens TIA PORTAL.pdf Apparue le 3 juin 2019

[8] Largeaud, Hubert. « Télémécanique, schématique électrotechnique», Eyrolles, 1986.

[9] C. Largeaud, Hubert «Canudas schéma éléctrique» édition eyrolles 1994
Française milsant '' machine électrique '' berti édition, 1993.

[10] J. P. Caron et J.P. Hautier, ''Modélisation et commande de la machine asynchrone''. Editions Technip, Paris, 1995.

[11] Française milsant «machine électrique '' berti édition, 1993.

[12] Physique Appliquée, terminale électrotechnique» , collection R. Mérat et R.

[13] Le Guide des Automatismes, centre académique de ressources et de maintenance informatique, Apparue le 5 juin 2019.

[14] Physique appliquée, terminale génie électrotechnique - Delva, Leclercq,
Trannoy – édition Hachette éducation 1994.

[15] Document Siemens, « Automate programmable S7-300, caractéristiques électriques et techniques des CPU SIMATIC,» Edition Siemens, 2001.

[16] MAS - le Moteur Asynchrone.odt, version du 15 avr. 2012

[17]https://www.schneider-electric.com/pdf/ATV12HU15M2 Apparue le 7 juin 2019

[18]https://cache.industry.siemens.com/dl/files/461/21084461/att_82592/v1/hmi_tp177a.pdf Apparue le 7 juin 2019