

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



Département d'ELECTROTECHNIQUE  
Laboratoire de Commande des processus

Mémoire de projet de fin d'études pour l'obtention du diplôme d'  
Ingénieur d'Etat en Electrotechnique  
Intitulé

**Réalisation d'une armoire à variateur de vitesse commandé par automate programmable.**

Présenté par

**Amar TIOURGUIOUNE et Mohamed Hocine SOUMATIA**

Sous la direction de Pr. El Madjid BERKOUK et Pr MO.MAHMOUDI

Soutenu publiquement le 26/06/2019

Membres du Jury

Présidente : H.Sahraoui Docteur à l'ENP  
Rapporteur : E.M.Berkouk Professeur à l'ENP  
Examineur : R.Tahmi professeur à l'ENP

ENP 2019



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



Département d'ELECTROTECHNIQUE  
Laboratoire de Commande des processus

Mémoire de projet de fin d'études pour l'obtention du diplôme d'  
Ingénieur d'Etat en Electrotechnique  
Intitulé

**Réalisation d'une armoire à variateur de vitesse commandé par automate programmable**

Présenté par

**Amar TIOURGUIOUNE et Mohamed Hocine SOUMATIA**

Sous la direction de Pr. El Madjid BERKOUK et Pr.MO.MAHMOUDI

Soutenu publiquement le 26/06/2019

Membres du Jury

Président : H.Sahraoui, Docteur à l'ENP  
Rapporteur : E.M.Berkouk Professeur à l'ENP  
Examineur : R.Tahmi Professeur à l'ENP

ENP 2019

## تلخيص

العمل المنجز في المذكرة يتمحور على إنشاء خزانة كهرباء بواسطة مسير صناعي مبرمج "سيمنس" ومغير السرعة "ألتيفار 312" حيث اقترحت دائرة هندسة الكهرباء للمدرسة الوطنية المتعددة التقنيات صناعة هذا المشروع في نهاية الدراسة من أجل استغلاله لغرض تعليمي.

تسير التحصيل ومعالجة المعلومات محقق من طرف هذا المسير الصناعي. لضبط المحطة استعملنا "برنامج" ستابسات" وللمراقبة استخدمنا برنامج «وينسيسيفلكسيل».

## الكلمات المفتاحية

"مسير صناعي مبرمج" سيمنس", برنامج" ستابسات", برنامج "وينسيسيفلكسيل".

## ABSTRACT

The work presented in this memory is based on realization of an electrical cabinet based on Siemens programmable controller, and an Altivar 312 speed dimmer. The Department of Electrical Technology of national polytechnic school has proposed this end-of-study project in order to exploit it for a didactic purpose.

We undertook a program of control with the STEP7 software and supervision with the WinCC flexible software.

### Keywords:

Siemens PLC S7-314 IFM, STEP7 software, WINCC software.

## RESUME :

Le travail présenté dans ce mémoire est basé sur la réalisation d'une armoire électrique à base d'automate programmable Siemens et un variateur de vitesse Altivar 312. Le Département d'Electrotechnique de l'Ecole Nationale Polytechnique a proposé ce projet de fin d'étude afin de l'exploiter dans un objectif didactique.

Nous avons entrepris la réalisation d'un programme de gestion par le logiciel STEP7, et de supervision par le progiciel WINCC flexible.

### Mots clés :

Automate programmable Siemens S7-314 IFM, logiciel STEP7, logiciel WinCC.

## REMERCIEMENTS

En préambule à ce mémoire nous remerciant ALLAH de nous avoir donné la patience, le courage et la force d'accomplir ce travail.

Nous tenons à exprimer nos vifs remerciements à notre promoteur Pr.BERKOUK pour nous avoir encadrées durant notre projet de fin d'études et nous conseillé tout le long de notre travail.

*Nous remercions également notre Co-promoteur Pr.MO.MAHMOUDI pour ses conseils et son aide.*

*Nous remercions également Pr.K.BOUGHRARA pour ses précieux conseils et son aide durant toute la période du travail.*

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptent d'évaluer notre projet.

Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Electrotechnique qui nous ont encadrées auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, pour leur disponibilité et leurs efforts.

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

# *Dédicaces*

*J'ai le grand honneur de dédier ce modeste travail  
À celle qui a donné un sens à mon existence en m'offrant une éducation digne de  
confiance.*

*À ma chère Mère sans elle je ne serais pas l'homme que je suis aujourd'hui.*

*À mon père, pour son amour et son soutien moral depuis mon enfance.*

*À ma sœur qui a su m'épauler dans les moments difficiles.*

*À ma grande mère BAYA qui a toujours été présente pour moi, m'encourager et  
qui m'a soutenu en toutes circonstances (que dieu te garde parmi nous).*

*À mes chères amis AOUALI Fatima, AMOURA Cylia et HAMIDI Youcef qui  
ont toujours été là pour moi.*

*À mon ami et binôme Amar TIOUERGVOUINE sur qui j'ai toujours pu  
compter.*

*À tous mes amis et camarades de l'École Nationale Polytechnique, et toute la  
promotion ELECTROTECHNIQUE de l'année 2019.*

*À tous ceux qui ont contribué de loin ou de prêt à notre travail.*

*SOUMATIA Mohamed Hocine*

## ***Dédicaces***

*J'ai le grand honneur de dédier ce modeste travail :*

*A la femme la plus courageuse, généreuse, la plus belle à mes yeux, à celle qui a su me donner amour et joie de vivre, à celle qui a toujours montrée affection à mon égard. A la personne qui a toujours été présente pour moi, m'encourager et me pousser de l'avant.*

*Celle qui m'a soutenu en toutes circonstances. A la femme de ma vie,*

*Ma mère que j'aime.*

*À ma sœur qui a toujours été là pour moi.*

*À mes grands-parents qui étaient et sont toujours là pour moi avec leurs conseils et soutien ; (que Dieu vous garde parmi nous).*

*A mon ami, mon binôme Hocine SOUMATIA pour tout le travail qu'on a accompli.*

*À mes oncles et tantes, mes cousins qui ont répondu présents, et mes cousines pour leurs encouragements.*

*À tous mes amis de l'Ecole Nationale Polytechnique avec qui j'ai eu d'agréable moments et appris beaucoup de choses.*

*A tous ceux qui ont contribués de loin ou de prêt à notre travail.*

***TIOURGUIOUINE Amar.***

# Sommaire

Liste des figures	
Liste des tableaux	
Liste des abréviations	
Introduction Générale .....	13
Chapitre I : Les automates Programmables industriels SIEMENS.....	15
I.1 Historique .....	16
I.2 Définition.....	16
I.3 Architecture des automates programmables industriels .....	17
I.4 Les différentes variantes dans la gamme SIMATIC. ....	18
I.4.1 Les modules LOGO.....	18
I.4.2 SIMATIC S7 .....	18
I.4.3 SIMATIC C7.....	20
I.5 Choix d'un API :.....	21
I.6 La gamme SIMATICS7-300 :.....	21
I.6.1 Description de la gamme SIMATIC S7 300 : .....	21
I.6.2 Avantage de la gamme SIMATIC S7-300 :.....	22
I.6.3 Structure d'un S7 300 .....	22
I.7 STEP 7, logiciel de programmation des API Siemens. ....	22
I.7.1 PRESENTATION.....	22
I.7.2 Fonctions du logiciel de base step7 : .....	23
I.7.3 Applications disponibles .....	23
I.7.4 SIMATIC Manager.....	24
I.7.5 Editeur des mnémoniques. ....	25
I.7.6 Configuration matérielle. ....	26
I.7.7 Langage de programmation pour API.....	27
I.7.8 Réquisition de la "Configuration matérielle". ....	27
I.7.9 Paramétrage de l'interface PG-PC.....	28
I.7.10 Blocs dans le programme utilisateur.....	28



I.7.11	Les différentes étapes pour l'implémentation d'un projet sous STEP7 .....	29
CHAPITRE II : Le logiciel de conception des interfaces Homme/Machine WinCC flexible .....		34
II.1	Présentation du système WinCC flexible. ....	35
II.2	Eléments d'interface logicielle de WinCC flexible.....	36
II.2.1	Menus et barres d'outils .....	36
II.2.2	Zone de travail.....	36
II.2.3	Fenêtre des propriétés .....	37
II.2.4	Boîte à outils .....	37
II.2.5	Bibliothèque .....	37
II.3	Editeur de Variables .....	37
II.4	Création des vues. ....	38
II.4.1	Navigation.....	38
II.4.2	Etapes.....	38
II.5	Liaison de WinCC flexible avec STEP7 .....	38
II.6	Exemple d'application d'un projet step 7, "WinCC " Doseur avec déplacement d'un convoyeur". ....	40
CHAPITRE III : Variateur de vitesse. ....		48
III.1	Principe de fonctionnement d'un variateur de vitesse.....	49
III.1.1	Redresseur :.....	50
III.1.2	Circuit intermédiaire :.....	51
III.1.3	L'onduleur :.....	52
III.2	Alimentation d'un moteur asynchrone triphasé a fréquence variable : .....	53
III.3	Variateur de vitesse Altivar 312.....	55
III.3.1	Description .....	55
III.3.2	Schéma de raccordement : .....	57
III.3.3	Consigne de vitesse : .....	57
III.3.4	Paramétrages du Variateur de vitesse ATV 312 :.....	58
III.3.5	Les menus du variateur ATV312 : [13] .....	59
III.3.6	Paramétrages du Variateur de vitesse ATV 312 :.....	61

CHAPITRE IV : La réalisation de l'armoire électrique.....	65
IV.1    Structure de la Tâche d'automatisation .....	66
IV.2    Description de l'installation de déplacement du chariot : .....	66
IV.3    Les éléments de commande et de supervision utilisés .....	67
IV.4    Mise en œuvre de l'application : « Entraînement d'un chariot » .....	68
IV.4.1    Description des cahiers de charge.....	68
IV.4.2    Grafcet et définition des cahiers de charge des 3 manipulations : .....	69
IV.4.3    Développement du projet S7 pour la maquette : .....	73
IV.5    Présentation de l'installation :.....	77
IV.5.1    Organisation électrique de l'armoire de commande.....	77
IV.5.2    Schéma de connexion des entrées et des sorties .....	79
IV.5.3    L'armoire de commande. ....	80
IV.5.4    Les éléments de manœuvre et de signalisation.....	80
IV.6    L'interface homme machine HMI configurée par le logiciel « Win CC flexible » .....	81
IV.6.1    Création de la station HMI :.....	81
IV.6.2    Paramétrage du pupitre : .....	83
IV.6.3    L'édition des variables :.....	84
IV.6.4    Configuration de la connexion entre l'API et le WinCC Advanced RT :.....	85
IV.6.5    Simulation du programme STEP 7 dans PLCSIM : .....	85
IV.6.6    Chargement dans le système cible à partir de la PG/PC .....	85
IV.7    Les tests en ligne : .....	86
Conclusion générale.....	88
Bibliographie :.....	89
ANNEXE A : Le programme de l'application « mise en œuvre d'un chariot ».....	91
Annexe B : Description des éléments utilisés. ....	101
ANNEXE C : Support TP. ....	104

# Liste des figures

Figure 1: Automate programmable Siemens s7-300. [2].....	17
Figure 2 : Automate programmable logo [4].....	18
Figure 3 : Automate programmable Siemens S7-200 [4].....	18
Figure 4 : Automate programmable Siemens S7-300 [4].....	19
Figure 5 : Automate programmable Siemens S7-400 [4].....	19
Figure 6 : Automate programmable Siemens S7-1200 [4].....	20
Figure 7: Automate programmable Siemens S7-1500 [4].....	20
Figure 8: Automate programmable Siemens C7 [4] .....	21
Figure 9: différentes applications du logiciel STEP 7. [5] .....	24
Figure 10: Le gestionnaire de projet STEP 7.....	24
Figure 11: Configuration matérielle pour la station S300. ....	26
Figure 12: Configuration matérielle pour la station S300. ....	26
Figure 13: Les types de bloc. [5] .....	28
Figure 14 : Création du programme S7 .....	29
Figure 15: Choix de la station SIMATIC 300. ....	30
Figure 16 : La procédure à suivre pour la configuration matérielle. ....	30
Figure 17: Edition des Mnémoniques. ....	31
Figure 18: Edition des programmes. ....	32
Figure 19: logiciel de simulation PLC-SIM. ....	33
Figure 20: La fenêtre principale de WinCC flexible [8] .....	36
Figure 21: Editeur de Variables WinCC. ....	37
Figure 22: Création d'une liaison de la station WinCC avec le niveau automatisation.....	38
Figure 23 : Zone de travail WinCC flexible. ....	39
Figure 24: Reprise des variables du projet STEP 7 déjà crée. ....	40
Figure 25 : Organigramme. ....	41
Figure 26: Configuration du matérielle.....	42
Figure 27 : Table de mnémoniques .....	42
Figure 28 : Transformation d'une entrée analogique a un intervalle du poids de la matière. ....	43
Figure 29 : Calcul de la vitesse selon des valeurs de poids de la matière et de la demande. ....	43
Figure 30 : Vitesse fixe à 1400 tr/m à niveau max. ....	43
Figure 31: Transformation la vitesse en sortie analogique. ....	44
Figure 32 : Arrêt d'urgence.....	44
Figure 33 : Alarme en cas un problème. ....	44
Figure 34 : l'appel des fonctions FC1, FC2. ....	45
Figure 35: Vue globale dans la station SIMATIC HMI .....	46
Figure 36: Exécution du programme de simulation Le système 'Runtime'. ....	47
Figure 37 : variateur de vitesse.....	50
Figure 38 : Redresseur triphasé à diodes.....	50
Figure 39 : tension redressée .....	51
Figure 40 : schéma du principe de filtrage.....	51
Figure 41 : schéma de principe d'un onduleur. ....	52

<b>Figure 42: circuit de puissance d'un onduleur de tension</b> .....	53
<b>Figure 43: Couple du moteur pour une fréquence variable</b> .....	54
<b>Figure 44 : Tension du moteur en fonction de la fréquence</b> .....	54
<b>Figure 45: variateur de vitesse Altivar 312 [13]</b> .....	55
<b>Figure 46: schéma de raccordement d'un Altivar 312 [10]</b> .....	57
<b>Figure 47 : Réglage du variateur de vitesse</b> .....	59
<b>Figure 48 : Synoptique du prototype</b> .....	67
<b>Figure 49: Diagramme des deux manipulations.</b> .....	69
<b>Figure 50: Grafcet de la manipulation 1 « Cahier de charge n°1 »</b> .....	69
<b>Figure 51: Chronogramme de fonctionnement</b> .....	71
<b>Figure 52 : Grafcet de la manipulation 2 « Cahier de charge n°2»</b> .....	71
<b>Figure 53: Chronogramme de fonctionnement.</b> .....	73
<b>Figure 54: Configuration matériel pour la CPU314 IFM.</b> .....	74
<b>Figure 55: blocs programme dans SIMATIC Manager</b> .....	75
<b>Figure 56: Tableau de mnémonique</b> .....	77
<b>Figure 57: l'organisation électrique de l'armoire.</b> .....	78
<b>Figure 58 : Schéma de connexion des entrées/sorties</b> .....	79
<b>Figure 59 : Schéma de câblage et l'emplacement des composants dans l'armoire</b> .....	79
<b>Figure 60 : les éléments de manœuvre et de signalisation.</b> .....	80
<b>Figure 61 : vue principale crée pour la HMI</b> .....	81
<b>Figure 62 : La vue des trois cahiers de charge</b> .....	82
<b>Figure 63: Vue cahier de charge n°1</b> .....	82
<b>Figure 64 : Vue cahier de charge n°2</b> .....	83
<b>Figure 65: Vue cahier de charge n°3</b> .....	83
<b>Figure 66 : paramétrage du pupitre</b> .....	84
<b>Figure 67 : l'édition des variables</b> .....	85
<b>Figure 68 : Configuration de la liaison MPI.</b> .....	85
<b>Figure 69 : visualisation de l'état zéro du premier cahier de charge</b> .....	86
<b>Figure 70 : visualisation de l'état 1 du premier cahier de charge.</b> .....	87

## Liste des tableaux

<b>Tableau 1 : Type de données</b> .....	25
<b>Tableau 2: Principaux menus du variateur ATV312 [11]</b> .....	60
<b>Tableau 3: principaux paramètres du Set- [11]</b> .....	60
<b>Tableau 4: principaux paramètres du drC-[11]</b> .....	60
<b>Tableau 5: principaux paramètres du CtL- [11]</b> .....	61
<b>Tableau 6: Adressage des entrées /sorties du processus</b> .....	75
<b>Tableau 7 : la fonction principale des composants de l'armoire</b> .....	78

## Liste des abréviations

<b>API</b>	Automate Programmable Industriel
<b>AI</b>	Analog Input
<b>AO</b>	Analog output
<b>CPU</b>	Central Processing Unit
<b>DI</b>	Digital Input
<b>DO</b>	Digital Output
<b>FB</b>	Fonctionnel Bloc
<b>FC</b>	Fonction
<b>MPI</b>	Multi Point Interface
<b>OB</b>	Organisationnel Bloc
<b>S7</b>	Step 7
<b>RAM</b>	Random Access Memory
<b>ROM</b>	Programmable Read Only Memory
<b>PC</b>	Personnel Computer
<b>PG</b>	La console de programmation sur le terrain
<b>PLC</b>	Programmable Logic Controller
<b>PROFIBUS</b>	Process Field Bus
<b>PROM</b>	Programmable Read Only Memory

# Introduction Générale

Dans le monde moderne, les machines et les équipements électriques font désormais partie intégrante de toutes les industries. L'automatisation des processus industriels est désormais d'une importance primordiale, pour l'amélioration de la production et de la qualité du produit.

L'automate programmable industriel A.P.I est aujourd'hui le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie car il répond à des besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

De plus, l'utilisation des machines électriques, en particulier les moteurs à induction et la variation de vitesse de ces derniers est d'une utilité indispensable dans le domaine industriel ; car les équipements utilisent de plus en plus l'entraînement à vitesse variable, ceci pour donner au dispositif entraîné la vitesse optimale pour chacune des phases d'un processus, mais surtout pour permettre l'automatisation qui nécessite la possibilité d'asservir la vitesse des moteurs.

Pour cela, on y fait appel non seulement à l'électronique de puissance, mais aussi aux théories de réglage et de commande pour ce type d'entraînements et l'application des M. A.S est devenue possible grâce à l'évolution de l'électronique de puissance dans le domaine des convertisseurs de fréquence statiques.

De ce fait, le variateur de vitesse a été d'un apport considérable dans ce domaine, il a donné une grande souplesse dans l'adaptation des montages à des applications diverses.

Le but de ce projet est de réaliser une armoire électrique à base d'API Siemens S7 300 et de variateur de vitesse ATV 31 qui assure la commande du moteur à induction triphasé sans capteur, utilisé pour l'entraînement d'un chariot. Ces derniers contrôlent les paramètres opérationnels en fonction de la vitesse programmée et assure le bon fonctionnement du système, de ce fait on a décrit les différentes étapes de conception, la mise en œuvre du matériel et des logiciels de configuration.

Suite à cela, le présent rapport comporte quatre grands chapitres :

- le premier chapitre est consacré à la présentation des automates programmables et le logiciel de programmation Step7.
- Le deuxième chapitre a servi à présenter le logiciel de conception des interfaces Homme-Machine appelé « SIMATIC WinCC ». Ce dernier est un système de supervision de processus modulables qui offre des fonctions performantes de surveillance des automatismes.
- Le troisième chapitre sera consacré à la description de la gamme des variateurs de fréquences ATV 31 de Schneider, une étude sur les variateurs de vitesse électroniques, le principe de fonctionnement, les différents composants et la configuration de ce dernier seront détaillées.
- En fin, le quatrième chapitre sera consacré à la description de l'armoire avec ses différents composants, ainsi que la simulation et la supervision du système choisi, par ailleurs nous y détaillerons l'aspect mécanique de la partie opérative du système.

A la fin nous présenterons la conclusion de cette étude ainsi que les perspectives envisageables pour la poursuite de ce travail.



**Chapitre I : Les automates**  
**Programmables industriels SIEMENS**

## **Introduction**

L'automate programmable industriel (API) en anglais « Programmable LogicController » (PLC), est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels.

L'automate programmable industriel (API) est le constituant principal de toute les chaines de production, son rôle principal c'est de récolter les informations via ses interfaces d'entrées venantes des capteurs, les traiter pour prendre une décision qui sera transmise à la fin vers les actionneurs à travers les interfaces de sorties.

### **I.1 Historique**

Les automates programmables industriels sont apparus à la fin des années soixante aux Etats Unis, à la demande de l'industrie automobile américaine (General Motors en leader), qui réclamait plus d'adaptabilité de ses systèmes de commande. Ce n'est qu'en 1971 qu'ils firent leur apparition en France.

Les années soixante-dix connaissent une explosion des besoins industriels dans le domaine de l'automatique, de la flexibilité et l'évolutivité des Systèmes Automatisés de Production (SAP).

Siemens AG est un groupe allemand. Fondé en 1847 par Werner Von Siemens, il réalise des équipements électroniques et électrotechniques. Son siège est à Munich, et c'est l'une des plus grosses entreprises européennes.

### **I.2 Définition**

L'Automate Programmable Industriel 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 issues des actionneurs [1].

« Appareil électronique qui comporte une mémoire programmable par un utilisateur automatique (et non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme par exemple :

- Logique séquentiel et combinatoire
- Temporisation, comptage, décomptage, comparaison
- Calcul arithmétique
- Réglage, asservissement, régulation ...etc.

### I.3 Architecture des automates programmables industriels

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

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM et EEPROM nécessaire pour stocker les programmes, les données, et les paramètres de configuration du système.
- Un module d'alimentation qui, à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues  $\pm 5V$ ,  $\pm 12V$  ou  $\pm 15V$ .
- Un ou plusieurs modules d'entrées 'Tout ou Rien' ou analogiques pour l'acquisition des informations provenant de la partie opérative (procédé à conduire).
- Un ou plusieurs modules de sorties 'Tout ou Rien' (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties [1].
- Un ou plusieurs modules de communication comprenant :
  - Interfaces série utilisant dans la plupart des cas comme support de communication, les liaisons RS-232 ou RS422/RS485 ;
  - Interfaces pour assurer l'accès à un bus de terrain ;
  - Interface d'accès à un réseau Ethernet. [2]

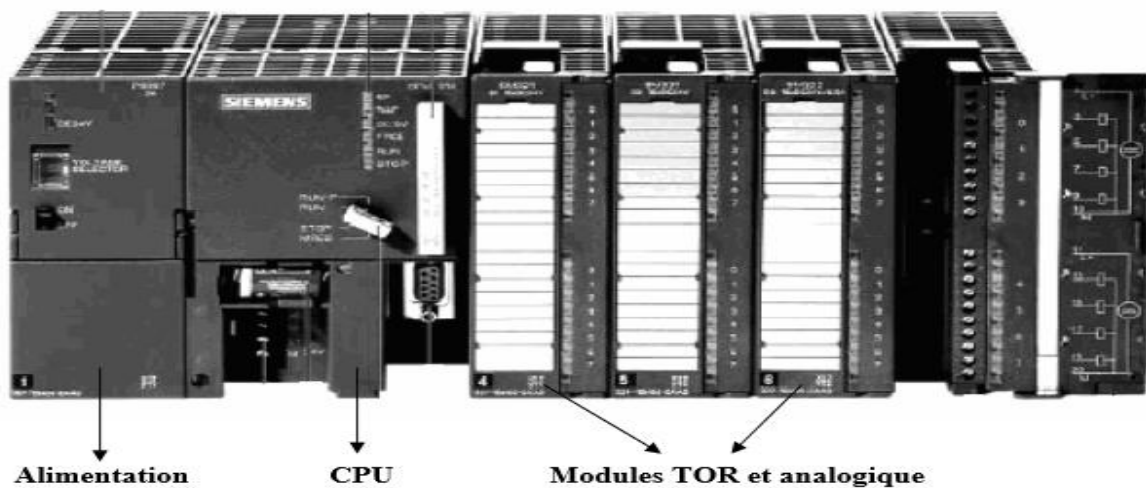


Figure 1: Automate programmable Siemens s7-300. [2]

#### I.4 Les différentes variantes dans la gamme SIMATIC.

##### I.4.1 Les modules LOGO

Ce sont des mini automates utilisés pour les petites applications d'automatisation telles que la domotique ou sur des petites machines. Les modules LOGO sont très flexibles, l'utilisateur a la possibilité d'ajouter des E/S supplémentaires via des modules d'extension.



Figure 2 : Automate programmable logo [4].

##### I.4.2 SIMATIC S7

Cette gamme d'automates comporte plusieurs automates :

- **S7 200 :**

C'est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.



Figure 3 : Automate programmable Siemens S7-200 [4].

- **S7-300**

C'est un automate modulaire haut gamme très utilisé dans les industries de transformation. Il est fourni sous différentes versions (CPU standard, CPU Fail-safe).



**Figure 4 : Automate programmable Siemens S7-300 [4]**

- **S7-400**

C'est un automate de haute gamme très utilisé dans les industries de processus. Les CPU de la gamme S7-400 disposent d'une mémoire de travail ; et d'un bon temps d'exécution répondant aux exigences des industries de processus.



**Figure 5 : Automate programmable Siemens S7-400 [4].**

- **S7-1200**

L'automate SIMATIC S7-1200 est un automate compact qui se programme avec le logiciel TIA PORTAL de Siemens. L'automate S7-1200 est compatible avec PROFINET IO, et peut être utilisé avec des appareils PROFINET IO pour résoudre diverses applications.



**Figure 6 : Automate programmable Siemens S7-1200 [4]**

- **S7-1500**

C'est la dernière gamme d'automates Siemens. Il se programme sous TIA PORTAL et dispose d'un petit écran de façade permettant de faire quelques configurations basiques.



**Figure 7: Automate programmable Siemens S7-1500 [4]**

### **I.4.3 SIMATIC C7**

Le SIMATIC C7 combine automate programmable et panneau opérateur dans une seule unité. L'automate compte la CPU, les modules d'entrées/sorties, et le panneau opérateur qui est utilisé comme une interface Homme/Machine HMI.

Le C7 permet la visualisation des états de fonctionnement, des valeurs actuelles du processus et des anomalies.



**Figure 8: Automate programmable Siemens C7 [4]**

### **I.5 Choix d'un API :**

Le choix d'un API est fonction de la partie commande à programmer. On doit tenir compte de plusieurs critères :

- Nombres d'entrées/sorties intégrés.
- Temps de traitement (scrutation).
- Capacité de la mémoire.
- Nombre de compteurs.
- Nombre de temporisateurs.
- Les langages de programmation.
- Le nombre de voies analogiques.
- La capacité de traitement arithmétique.
- La bibliothèque fonctionnelle.

### **I.6 La gamme SIMATIC S7-300 :**

#### **I.6.1 Description de la gamme SIMATIC S7 300 :**

Le S7-300 est l'automate conçu pour des solutions dédiées au système manufacturier et constitue à ce titre une plate-forme d'automatisation universelle pour les applications avec des architectures centralisées et décentralisées.

Le S7-300 offre une gamme échelonnée de 24 CPU ; des CPUs standard parmi lesquelles la première CPU avec interface Ethernet/PROFINet intégrée, des CPUs de sécurité, des CPU compacts avec fonctions technologiques et périphérie intégrées et CPU technologiques pour la gestion des fonctions motion control.

Le S7-300 offre également une très large palette de modules d'E/S TOR et analogiques pour la quasi-totalité des signaux avec possibilité de traitement des interruptions et du diagnostic.

### **I.6.2 Avantage de la gamme SIMATIC S7-300 :**

Une construction compacte et modulaire, libre de contraintes de configuration. Une riche gamme de modules adaptés à tous les besoins du marché est utilisable en architecture centralisée ou décentralisée, qui réduit grandement le stock de pièces de rechange. Une large gamme de CPU adaptée à toutes les demandes de performances pour pouvoir d'obtenir des temps de cycle machines courts, certaines étant dotées de fonctions technologiques intégrées comme par ex. le comptage, la régulation ou le positionnement. Une économie d'ingénierie en utilisant les outils orientés application et normalisés CEI 1131-3 tels que les langages évolués SCL ou des logiciels exécutifs orientés technologie pour le contrôle des mouvements.

### **I.6.3 Structure d'un S7 300**

Un automate S7-300 est constitué des éléments suivants :

- Contrôleur compact avec fonction de positionnement intégrée (1 support x 1 codeur incrémental).
- (1) interface RS-485 (MPI), mémoire de 32 Ko pour le programme utilisateur des - données,
- (16) Entrées TOR 24Vd.c
- (16) sorties TOR 24Vd.c
- (4) Entrées analogiques tension/courant
- (1) sortie analogiques tension/courant
- Alimentation 24Vd.c intégrée, avec connecteurs (2 x 40 broches)

## **I.7 STEP 7, logiciel de programmation des API Siemens.**

### **I.7.1 PRESENTATION**

STEP 7 est le progiciel de base utilisé pour la configuration et la programmation des automates programmables. Il fait partie l'industrie logicielle SIMATIC. Il existe plusieurs versions du logiciel de base STEP 7 :

- STEP 7 Micro / Win pour des applications autonomes plus simples sur SIMATIC S7-200.
- STEP 7 pour les applications sur SIMATIC S7-300 / S7-400, SIMATIC M7-300 avec un large éventail de fonctions :
- Possibilité d'extension du logiciel SIMATIC avec d'autres logiciels



## Chapitre I : Les automates programmables industriels SIEMENS.

-Possibilité d'affecter des paramètres aux modules fonctionnels et aux processeurs de communication.

- Mode de forçage et fonctionnement multiprocesseur.
- Communication globale de données.
- Transfert de données événementiel à l'aide de blocs fonction de communication.
- Configuration des connexions.

### **I.7.2 Fonctions du logiciel de base step7 :**

Le logiciel vous assiste dans toutes les phases du processus de création d'une tâche d'automatisation, telles que :

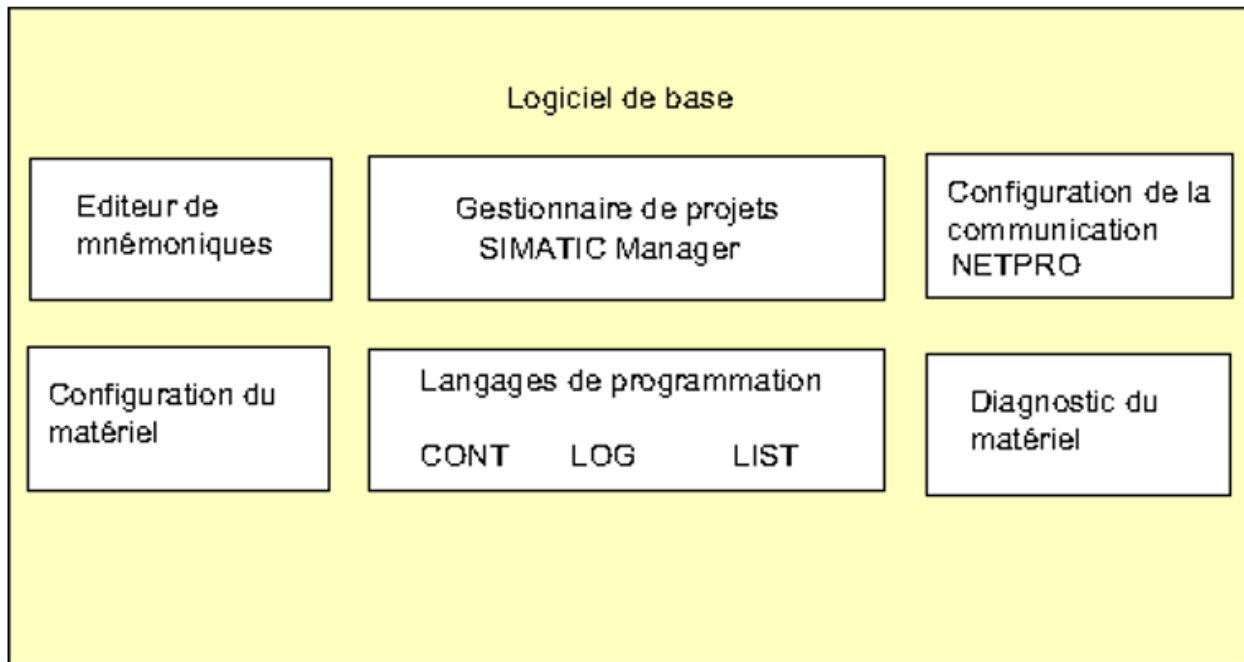
- Création et gestion de projets.
- Configuration et affectation du matériel et de la communication.
- Gestion des mnémoniques.
- Création de programmes, par exemple, pour automates S7
- Chargement de programmes sur des automates programmables
- Test du système d'automatisation
- Diagnostic des défaillances de l'installation.

L'interface utilisateur du logiciel STEP 7 a été conçue pour répondre aux exigences d'ergonomie les plus récentes.

### **I.7.3 Applications disponibles**

STEP 7 comporte les quatre sous logiciels de base suivants :

Le progiciel STEP 7 Standard fournit une série d'applications (outils) dans le logiciel:

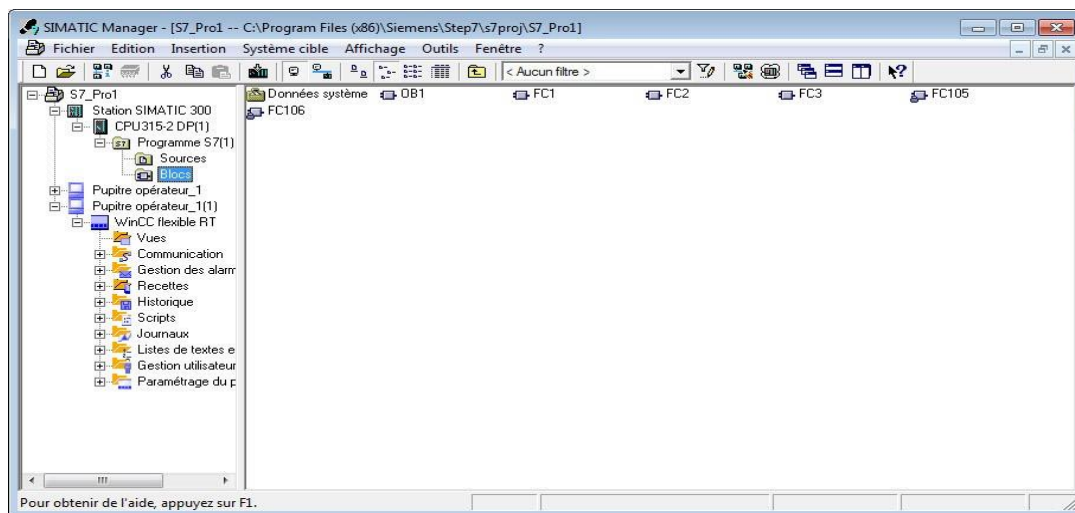


**Figure 9: différentes applications du logiciel STEP 7. [5]**

Vous n'avez pas besoin d'ouvrir les outils séparément ; ils sont lancés automatiquement lorsque vous sélectionnez la fonction correspondante ou ouvrez un objet.

#### **I.7.4 SIMATIC Manager.**

SIMATIC Manager gère toutes les données appartenant à un projet d'automatisation, quel que soit le système de contrôle programmable (S7 / M7 / C7) pour lequel elles ont été conçues. Les outils nécessaires à la modification des données sélectionnées sont lancés automatiquement par SIMATIC Manager. [6]



**Figure 10: Le gestionnaire de projet STEP 7.**

### I.7.5 Editeur des mnémoniques.

L'éditeur des mnémoniques permet de gérer toutes les variables globales. Les fonctions disponibles sont :

- Définition de noms symboliques et de commentaires pour les signaux de processus (entrées / sorties), les mémentos et les blocs.
- Fonctions de tri.
- Importation / exportation vers / depuis d'autres programmes Windows.

La table des mnémoniques créée est mise à disposition de toutes les applications.

Toute modification des propriétés d'un paramètre d'une mnémonique est donc automatiquement reconnue par toutes les applications.

#### Opérandes et types de données autorisés dans la table des mnémoniques :

Anglais	Allemand	Désignation	Type de données
I	E	Bit d'entrée	BOOL
IB	EB	Octet d'entrée	BYTE, CHAR
IW	EW	Mot d'entrée	WORD, INT, S5TIME, DATE
ID	ED	Double mot d'entrée	DWORD, DINT, REAL, TOD, TIME
Q	A	Bit de sortie	BOOL
QB	AB	Octet de sortie	BYTE, CHAR
QW	AW	Mot de sortie	WORD, INT, S5TIME, DATE
QD	AD	Double mot de sortie	DWORD, DINT, REAL, TOD, TIME
M	M	Bit de memento	BOOL
MB	MB	Octet de memento	BYTE, CHAR
MW	MW	Mot de memento	WORD, INT, S5TIME, DATE
MD	MD	Double mot de memento	DWORD, DINT, REAL, TOD, TIME
PIW	PEW	Mot de périphérie d'entrée	WORD, INT, S5TIME, DATE
PQW	PAW	Mot de périphérie de sortie	WORD, INT, S5TIME, DATE
PID	PED	Double mot de périphérie d'entrée	DWORD, DINT, REAL, TOD, TIME
PQD	PAD	Double mot de périphérie de sortie	DWORD, DINT, REAL, TOD, TIME
T	T	Temporisation	TIMER
C	Z	Compteur	COUNTER
FB	FB	Bloc fonctionnel	FB
OB	OB	Bloc d'organisation	OB
DB	DB	Bloc de données	DB, FB

Tableau 1 : Type de données

En affectant des noms symboliques aux adresses absolues, les programmes deviennent plus lisibles, faciles à corriger et à mettre à jour.

### I.7.6 Configuration matérielle.

Par "configuration", on entend dans ce qui suit la disposition de profilés support ou châssis, de modules, d'appareils de la périphérie décentralisée et de cartouches interface dans une fenêtre de station. Les profilés support ou châssis sont représentés par une table de configuration, dans laquelle l'on peut enficher un nombre défini de modules, tout comme dans les profilés support ou châssis "réels".

STEP 7 affecte automatiquement une adresse à chaque module dans la table de configuration. Vous pouvez modifier les adresses des modules d'une station, à condition que la CPU permette l'adressage libre.:[7]

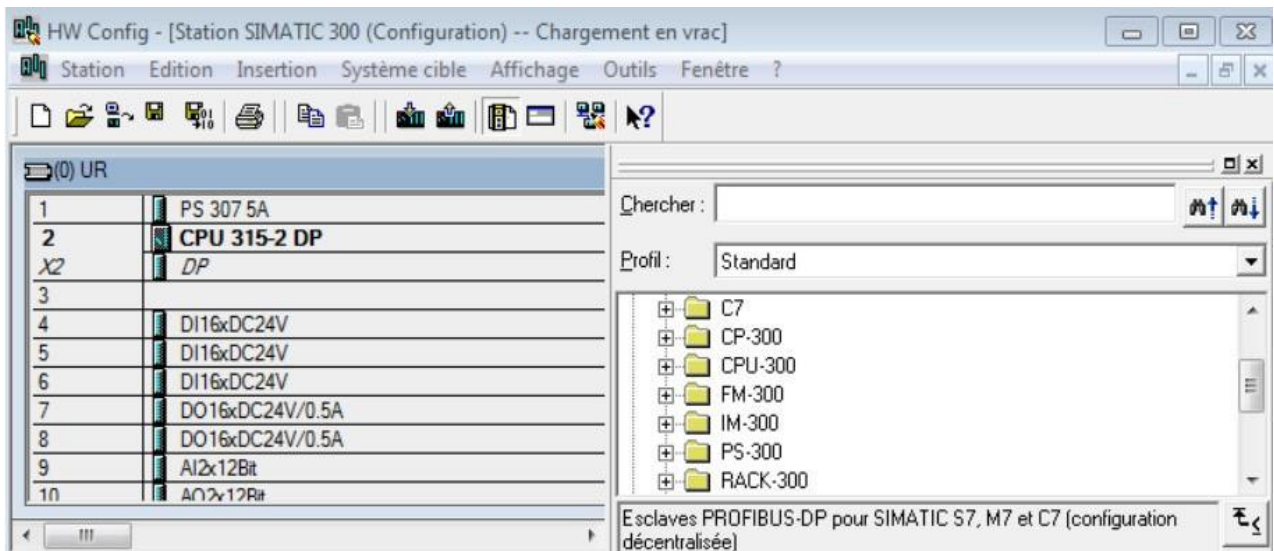


Figure 11: Configuration matérielle pour la station S300.

### Règles pour la disposition des modules (SIMATIC 300) [7]

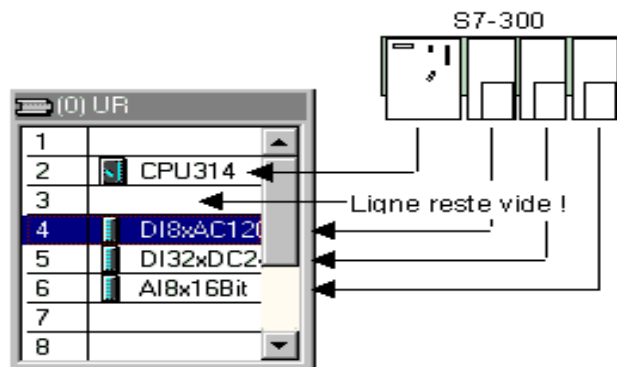


Figure 12: Configuration matérielle pour la station S300.

### **I.7.7 Langage de programmation pour API**

Les programmes utilisés avec les API peuvent être écrits dans différents formats. Pour que les ingénieurs ayant peu de connaissances en programmation puissent élaborer des programmes pour les API. La norme IEC 1131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels :

- Le langage LD (Ladder Diagram) :

C'est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie).

- Le langage IL (Instruction List) :

Un langage textuel de bas niveau. Il est particulièrement adapté aux applications de petites tailles. L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est stocké à son tour dans le résultat courant.

- Le langage FBD (Function Block Diagram) :

C'est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.

- Le langage ST (StructuredText) :

Un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques.

- Le langage SFC (SequentialFunction Chart), ou GRAFCET :

C'est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. [3]

### **I.7.8 Réquisition de la "Configuration matérielle".**

Les paramètres des automates programmables S7 et des modules ont été prédéfinis de sorte à ne pas nécessiter de configuration dans bien des cas. La configuration est obligatoire :

- lorsque vous souhaitez modifier les paramètres prédéfinis d'un module (par exemple, valider l'alarme de processus pour un module).
- lorsque vous souhaitez configurer des liaisons de communication.
- pour les stations avec une périphérie décentralisée (PROFIBUS DP ou PROFINET IO).
- pour les stations S7-400 avec plusieurs CPU (multiprocesseur) ou châssis d'extension.
- pour les automates hautes disponibilité.

### I.7.9 Paramétrage de l'interface PG-PC.

Cet outil sert à paramétrer l'adresse locale des PG/PC, la vitesse de transmission dans le réseau MPI ou PROFIBUS en vue d'une communication avec l'automate et le transfert du projet.

### I.7.10 Blocs dans le programme utilisateur.

Le logiciel de programmation STEP 7 permet de structurer le programme utilisateur, c'est-à-dire de le subdiviser en différentes parties autonomes.

Il en résulte les avantages suivants :

- Écrire des programmes importants mais clairs.
- Standardiser certaines parties du programme.
- Simplifier l'organisation du programme.
- Modifier facilement le programme.
- Simplifier le test du programme, car vous pouvez l'exécuter section par section.
- Faciliter la mise en service.

#### Types de bloc :

Nous pouvons utiliser différents types de bloc dans un programme utilisateur S7 :

Bloc	Brève description de la fonction
Blocs d'organisation (OB)	Les OB déterminent la structure du programme utilisateur.
Blocs fonctionnels système (SFB) et fonctions système (SFC)	Les SFB et SFC sont intégrés à la CPU S7 et vous permettent de réaliser quelques fonctions systèmes importantes.
Blocs fonctionnels (FB)	Les FB sont des blocs avec "mémoire" que vous programmez vous-même.
Fonctions (FC)	Les FC contiennent des routines de programmes pour les fonctions fréquemment utilisées.
Blocs de données d'instance (DB d'instance)	Les DB d'instance sont affectés au bloc FB/SFB appelé. Ils sont générés automatiquement lors de la compilation.
Blocs de données (DB)	Les DB sont des zones de données dans lesquelles l'on enregistre les données utilisateur. Outre les données affectées respectivement à un bloc fonctionnel, vous pouvez définir des données globales utilisables par tous les blocs.

Figure 13: Les types de bloc. [5]

### I.7.11 Les différentes étapes pour l'implémentation d'un projet sous STEP7

Avant d'utiliser STEP 7, on doit planifier notre solution d'automatisation depuis la division du processus en tâches individuelles jusqu'à la réalisation d'un schéma de configuration. En utilisant les blocs mis à notre disposition par STEP 7, nous transposons les tâches décrites lors de la conception de notre solution d'automatisation en structure de programme.

Les étapes suivantes vont nous permettre de mettre en place une solution d'automatisation complète :

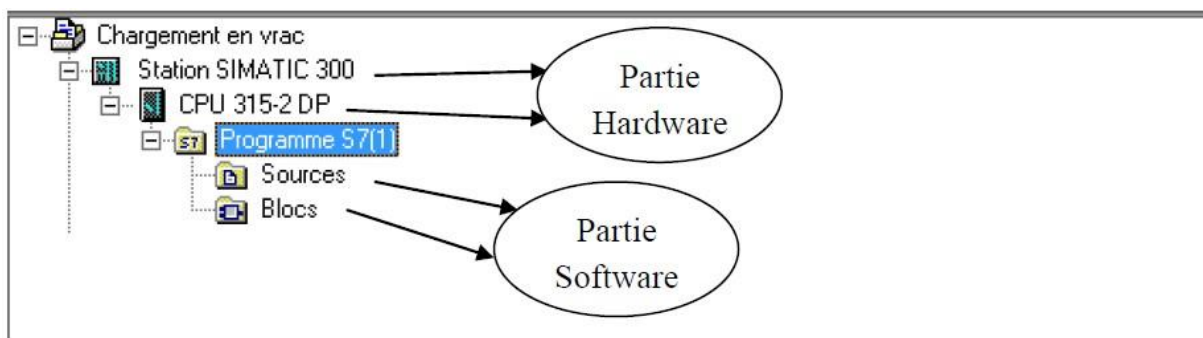
#### 1. Création du projet SIMATIC Step7 à l'aide de l'assistant.

Le plus simple pour créer un nouveau projet, c'est d'avoir recours à l'assistant "Nouveau projet".

Pour l'appeler, il faut choisir la commande **Fichier > Assistant "Nouveau projet"**.

Il nous posera les questions nécessaires dans des boîtes de dialogue et créera le projet pour nous. Outre la station, la CPU, les dossiers Programmes, Sources et Blocs ainsi que l'OB1, nous pouvons déjà y sélectionner les OB de traitement d'erreurs et d'alarmes.

La figure suivante illustre un projet créé à l'aide de l'assistant.



**Figure 14 : Création du programme S7**

Dans un projet, la station représente la configuration matérielle de l'automate programmable et contient les données pour la configuration et le paramétrage des divers modules.

Les nouveaux projets créés par l'assistant "Nouveau projet" contiennent déjà une station, nous pouvons également créer la station en choisissant la commande **Insertion > Station**. nous pouvons choisir les stations suivantes :

- Station SIMATIC 300.
- Station SIMATIC 400.
- Station SIMATIC PC:représente un PC contenant des composants.), ou une carte CPU enfichée dans le PC.



- PC/PG:c'est une console de programmation compatible avec le milieu industriel.
- SIMATIC S5.
- Autre station, c'est-à-dire SIMATIC S7/M7 ou bien des appareils d'autres fabricants.

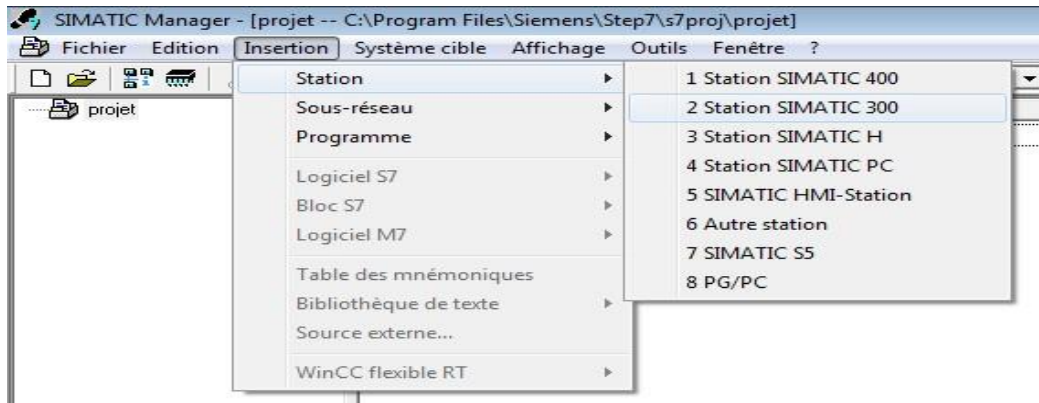


Figure 15: Choix de la station SIMATIC 300.

## 2. Configuration du matériel

- **Marche à suivre.**

Pour configurer et paramétrer une installation, vous devez procéder aux étapes suivantes :

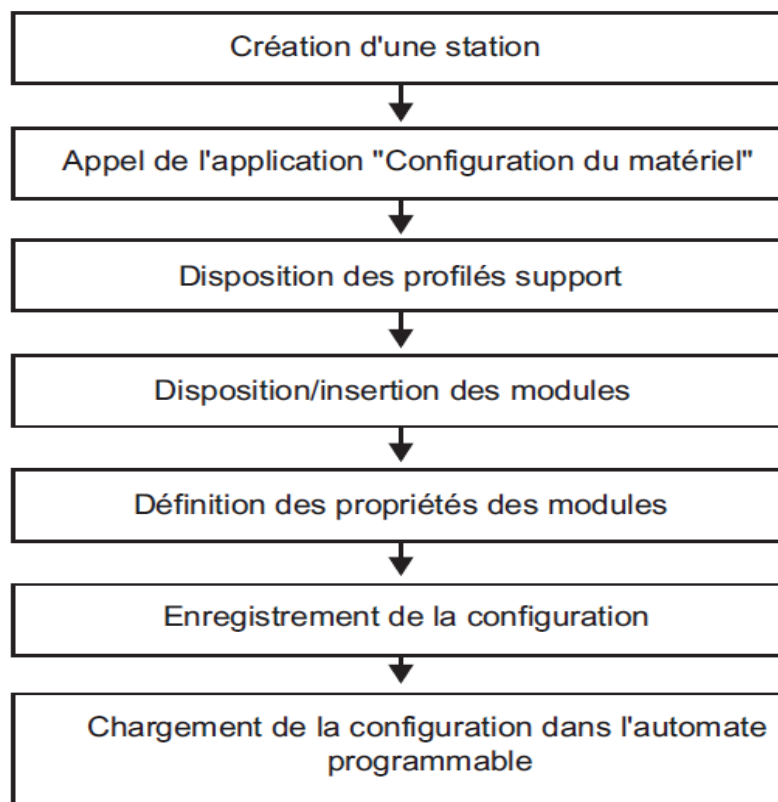


Figure 16 : La procédure à suivre pour la configuration matérielle.



### 3. Définition des mnémoniques

Dans un programme STEP 7, nous utilisons des opérandes comme des signaux d'E/S, des mémorisés, des compteurs, des temporisations, des blocs de données et des blocs fonctionnels. Nous pouvons accéder à ces opérandes par adressage absolu dans notre programme. Toutefois, la lisibilité de nos programmes sera grandement améliorée si nous faisons plutôt appel à des mnémoniques (par ex : marche, arrêt). Il est alors possible d'accéder aux opérandes de notre programme utilisateur via ces mnémoniques.[1]

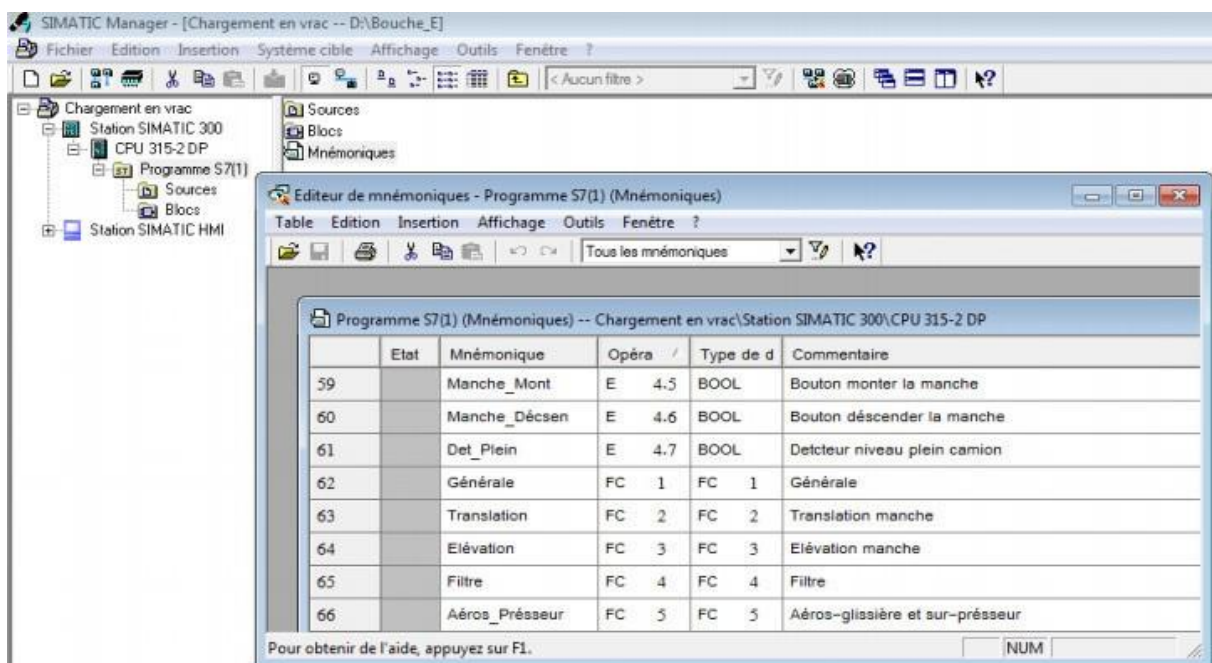
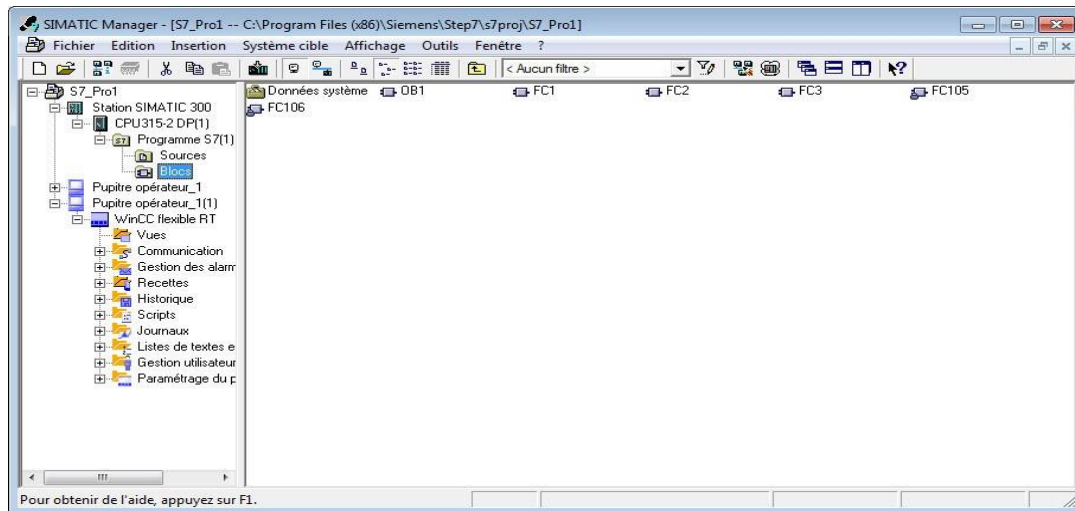


Figure 17: Edition des Mnémoniques.

### 4. Edition des programmes

Dans la section 'bloc' du SIMATIC Manager, on trouve par défaut le bloc d'organisation 1 'OB1' qui représente le programme cyclique. On peut rajouter d'autres blocs à tout moment par un clic droit dans la section Bloc de SIMATIC Manager [6].



**Figure 18: Edition des programmes.**

## 5. Chargement du programme dans la CPU

Une fois la configuration, le paramétrage et la création du programme terminés, on peut transférer le programme utilisateur dans le système cible. La CPU contient déjà le système d'exploitation.

## 6. Test des programmes avec logiciel de simulation S7-PLCSIM

Le logiciel optionnel de simulation permet d'exécuter et de tester le programme dans un système d'automatisation que nous simulons dans notre ordinateur ou dans notre console de programmation.

La CPU S7 simulée nous permet de tester les programmes et de remédier à d'éventuelles erreurs.

Cette application dispose d'une interface simple vous permettant de surveiller et de modifier les différents paramètres utilisés par le programme (comme par exemple d'activer ou de désactiver des entrées).

### **S7-PLCSIM possède aussi 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.

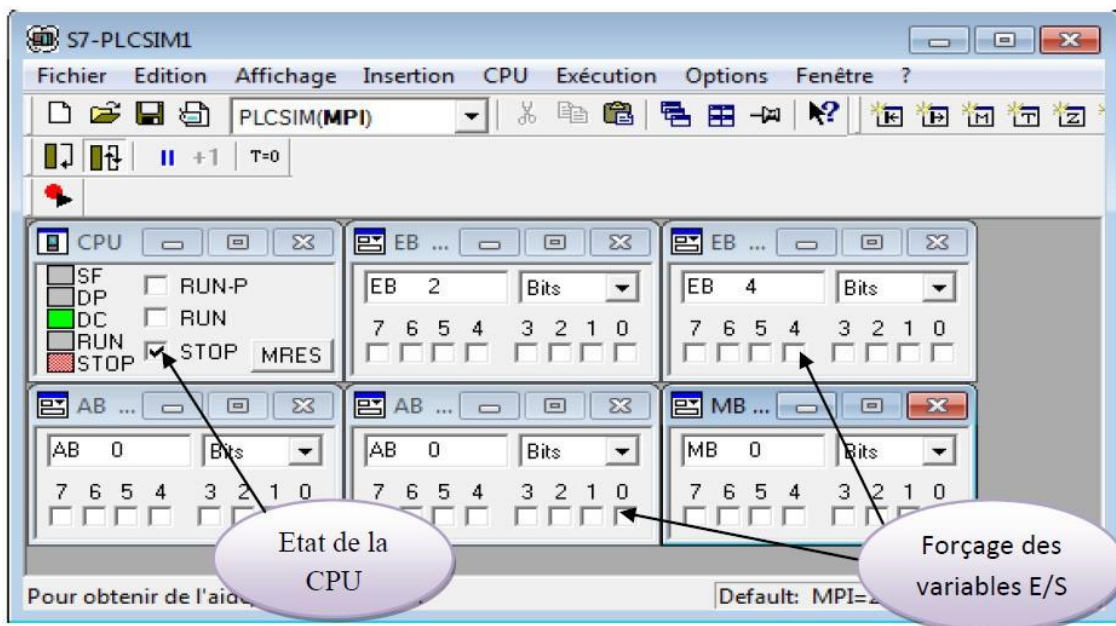


Figure 19: logiciel de simulation PLC-SIM.

### Conclusion :

Dans ce chapitre nous avons présenté l'historique de l'automate programmable industriel, son principe de fonctionnement, son architecture et par la suite une description de l'automate programmable S7-300 et le logiciel de programmation STEP7, comme on a détaillé les étapes de la création et de la configuration d'un projet. A la fin, une description des différentes étapes pour une structure d'automatisation complète a été présentée en détails.

**CHAPITRE II : Le logiciel de conception  
des interfaces Homme/Machine WinCC  
flexible**

## **Introduction à SIMATIC HMI**

Lorsque la complexité des processus augmente et que les machines et installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme-Machine (IHM).

SIMATIC HMI offre une gamme complète permettant de couvrir toutes les tâches de contrôle-commande. SIMATIC HMI vous permet de maîtriser le processus à tout instant et de maintenir les machines et installations en état de marche.

WinCC flexible est le logiciel IHM pour la réalisation, par des moyens d'ingénierie simples et efficaces, de concepts d'automatisation évolutifs, au niveau machine.

WinCC flexible réunit les avantages suivants:

- Simplicité.
- Ouverture.
- Flexibilité.

### **II.1 Présentation du système WinCC flexible.**

- **Éléments de WinCC flexible.**
  - **WinCC flexible Engineering System.**

WinCC flexible Engineering System est le logiciel avec lequel vous réalisez toutes les tâches de configuration requises. L'édition WinCC flexible détermine les pupitres opérateurs de la gamme SIMATIC HMI pouvant être configurés.

- **WinCC flexible Runtime.**

WinCC flexible Runtime est le logiciel de visualisation de processus. Dans Runtime, vous exécutez le projet en mode processus.

Au Runtime, l'opérateur peut réaliser le contrôle-commande du processus. Les tâches suivantes sont alors exécutées :

- Communication avec les automates.
- Affichage des vues à l'écran.

Commande du processus, exemple : spécification de consignes ou ouverture et fermeture de vannes.

## II.2 Eléments d'interface logicielle de WinCC flexible.

WinCC flexible se compose des éléments suivants :

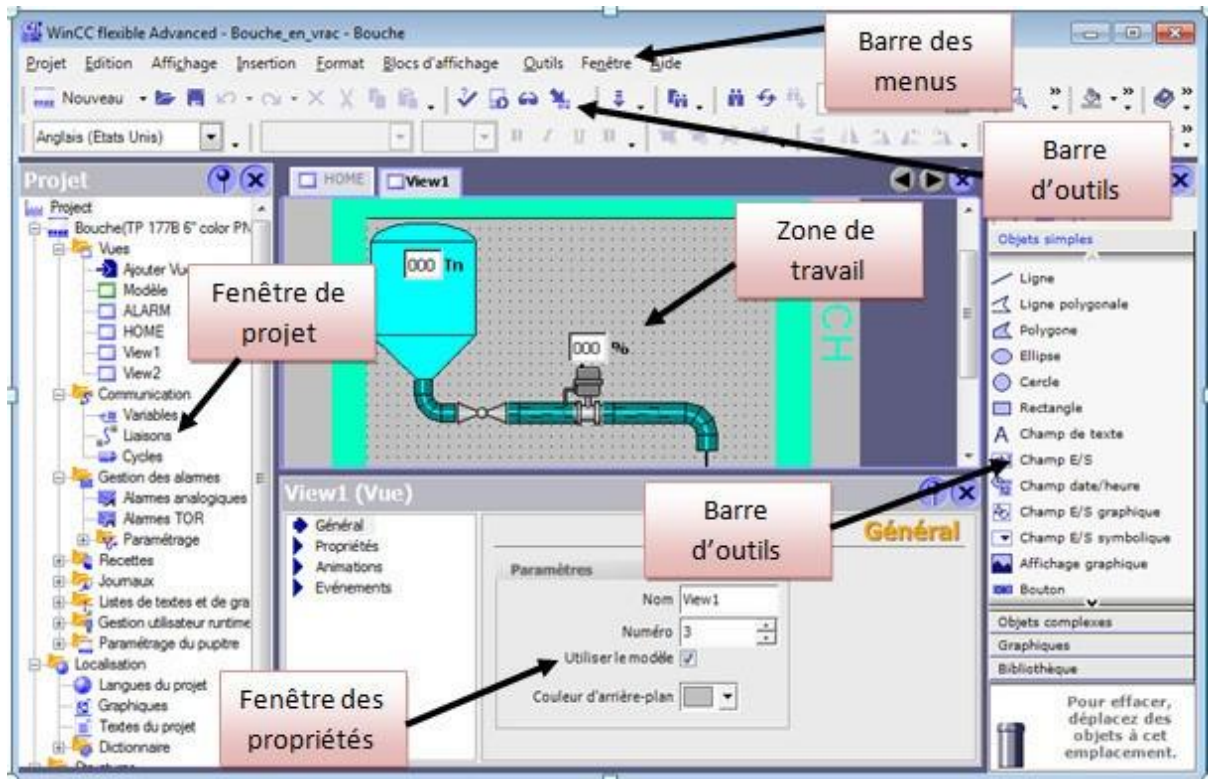


Figure 20: La fenêtre principale de WinCC flexible [8]

### II.2.1 Menus et barres d'outils

Les menus et barres d'outils nous donnent accès à toutes les fonctions disponibles sous WinCC flexible. Lorsqu'on positionne le pointeur de la souris sur une fonction, on obtient une info-bulle.

### II.2.2 Zone de travail

La zone de travail sert à éditer les objets du projet. Tous les éléments de WinCC flexible sont disposés autour de la zone de travail.

#### Fenêtre de projet

Tous les éléments et tous les éditeurs disponibles d'un projet sont affichés sous forme d'arborescence dans la fenêtre du projet et peuvent être ouverts à partir de cette fenêtre. Sous chaque éditeur se trouvent les dossiers, dans lesquels un stockage structuré des objets est possible.

### II.2.3 Fenêtre des propriétés

La fenêtre des propriétés nous permet d'éditer les propriétés des objets, p. ex. la couleur des objets de vue. Elle n'est disponible que dans certains éditeurs.

### II.2.4 Boîte à outils

La fenêtre d'outils nous propose un choix d'objets que nous pouvons insérer dans les vues, p. ex. des objets graphiques et éléments de commande. La fenêtre d'outils contient aussi des bibliothèques d'objets et collections de blocs d'affichage prêts à l'emploi.

### II.2.5 Bibliothèque

La bibliothèque fait partie de la fenêtre d'outils. La bibliothèque vous donne accès aux objets de vue préconfigurés. Les objets de la bibliothèque permettent d'augmenter la quantité d'objets de vue disponibles et d'améliorer votre productivité lors de la configuration par la réutilisation d'objets préconfigurés. La bibliothèque est le lieu central d'enregistrement des objets fréquemment utilisés tels que les objets graphiques et variables.

## II.3 Editeur de Variables

On crée et configure des variables dans l'éditeur Variables par suite toutes les variables d'un dossier seront affichées

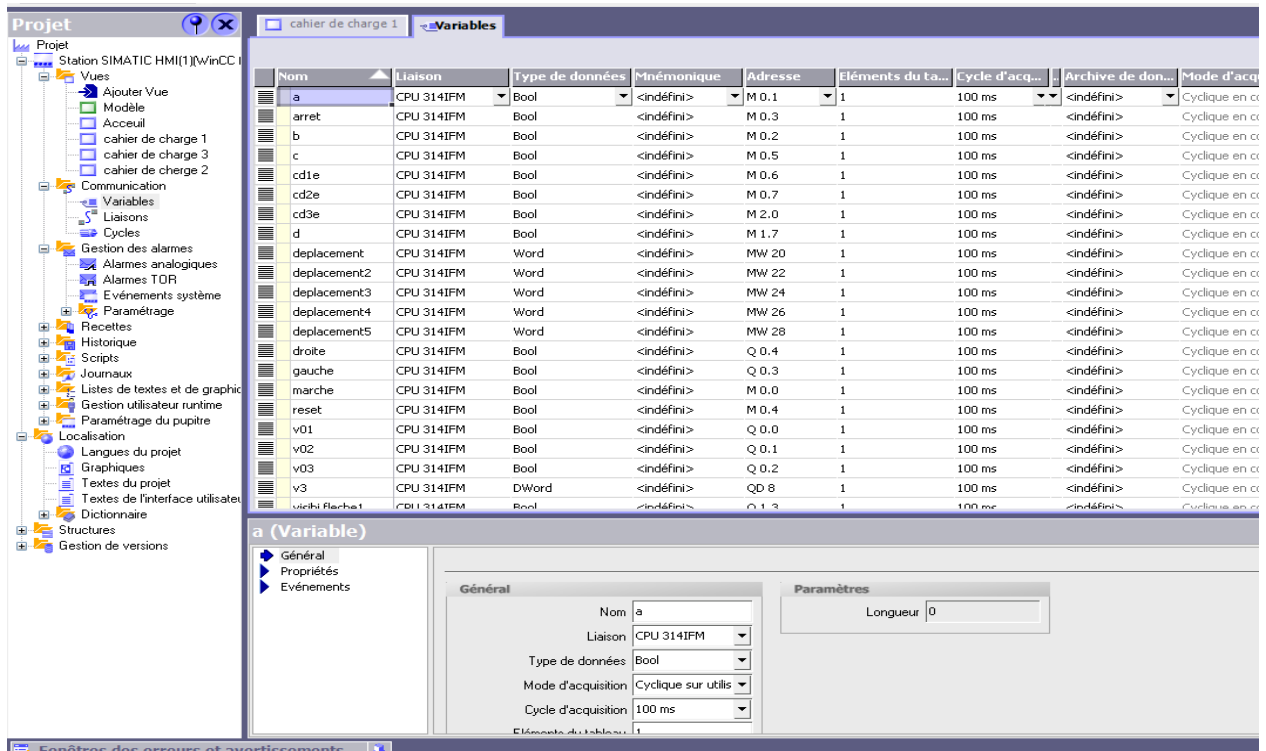


Figure 21: Editeur de Variables WinCC.



## II.4 Création des vues.

Dans WinCC flexible, on crée des vues pour le contrôle-commande de machines et d'installations. On dispose d'objets prédéfinis permettant de représenter notre installation, d'afficher des procédures et de définir des valeurs de processus.

### II.4.1 Navigation.

Afin que l'opérateur puisse appeler une vue en Runtime sur le pupitre opérateur, vous devez intégrer chaque vue configurée dans la séquence de vues. Pour cela on utilise l'éditeur de vues, afin de configurer, dans les vues, des boutons et des touches de fonction permettant d'appeler d'autres vues.

### II.4.2 Etapes.

Les principales étapes ci-dessous sont nécessaires à la création de vues :

- Planifiez la structure de la représentation du processus (le nombre de vues et la hiérarchie nécessaires).

Exemple : les processus partiels peuvent être représentés dans des vues séparées, puis regroupés en une vue principale.

- Planifiez la navigation au sein d'une vue et entre les différentes vues.
- Adaptez le modèle.

## II.5 Liaison de WinCC flexible avec STEP7

### II.5.1 Configuration de liaisons

Des liaisons de communication sont nécessaires pour que WinCC flexible puisse échanger des données avec le niveau automatisation.

### II.5.2 Utilisation de WinCC flexible

On peut créer des liaisons ou les éditer. Dans les projets intégrés, l'éditeur de configuration de liaisons propose les colonnes additionnelles "Station", "Partenaire" et "Nœud".

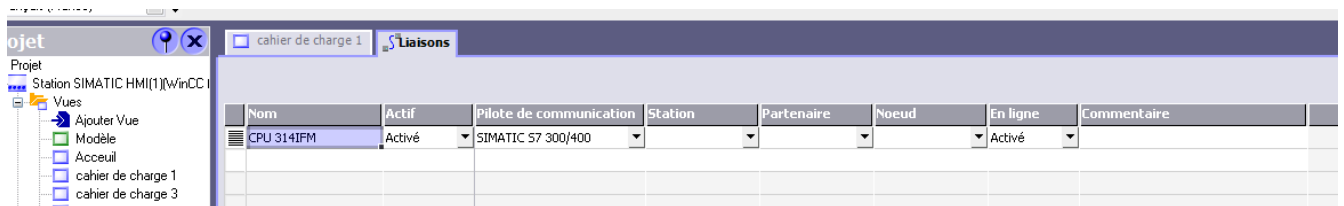
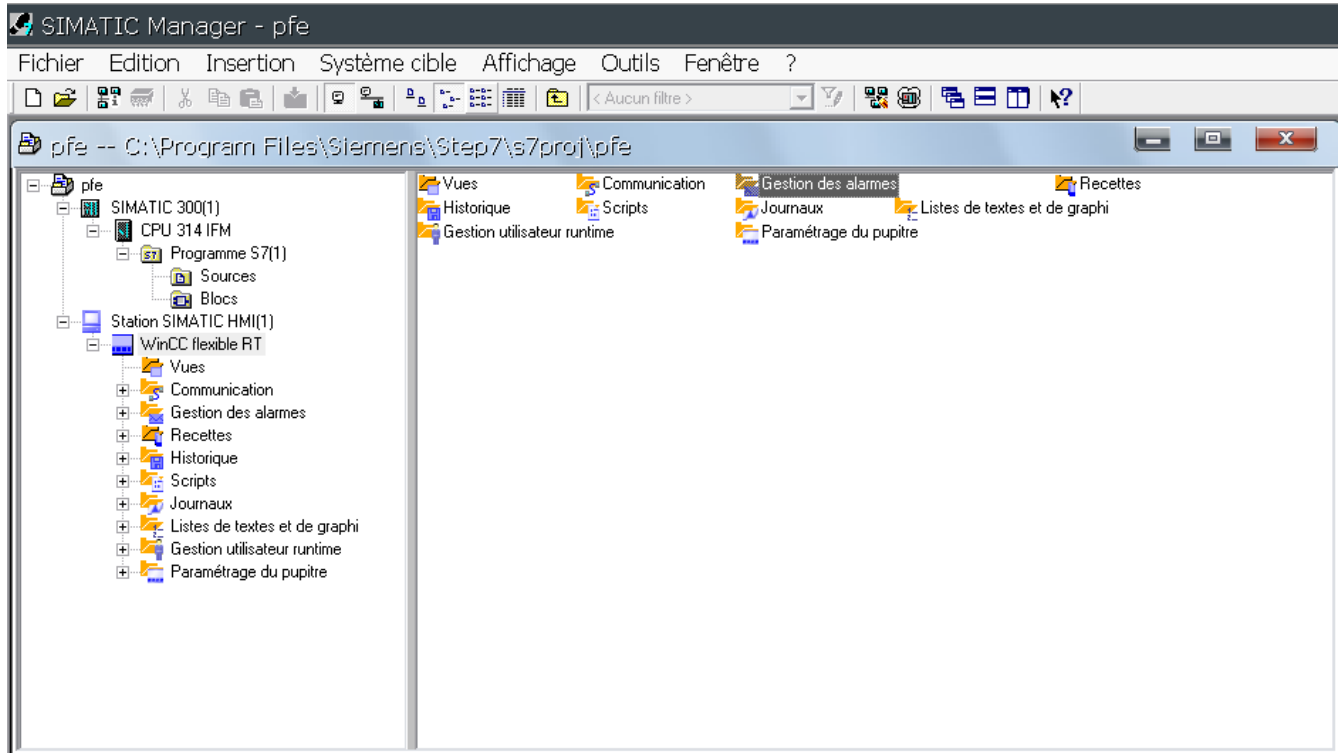


Figure 22: Création d'une liaison de la station WinCC avec le niveau automatisation.



La création d'une station HMI dans le SIMATIC Manager génère toujours un nouveau projet WinCC flexible.

Après intégration d'un projet WinCC flexible dans STEP 7, ce projet est affiché dans la fenêtre de projet du SIMATIC Manager.



**Figure 23 : Zone de travail WinCC flexible.**

On peut y ouvrir les objets disponibles ou y créer de nouveaux objets et WinCC flexible est automatiquement démarré pour l'édition de l'objet.

### **II.5.3 Reprise de variables de STEP 7.**

Pour reprendre des variables de STEP 7, on ouvre l'éditeur de variables sous WinCC flexible. On ajoute une nouvelle variable puis on navigue jusqu'au programme S7 de l'automate voulu, et on sélectionne le mnémonique requis dans la liste des mnémoniques ou dans un bloc de données.

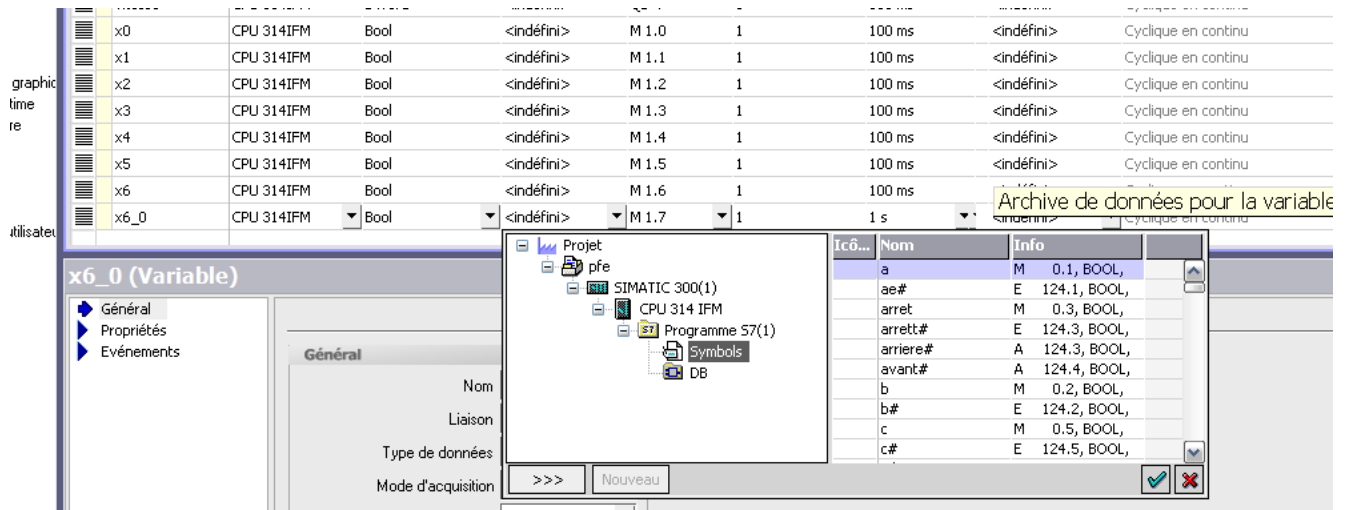


Figure 24: Reprise des variables du projet STEP 7 déjà créée.

## II.6 Exemple d'application d'un projet step 7, "WinCC " Doseur avec déplacement d'un convoyeur".

### 1. Cahier de charge

La méthode de travail du doseur se présente par le calcul du poids de la matière (Kg), qui se trouve au-dessus du convoyeur du doseur à travers la cellule de pesage.

La maîtrise de son utilisation est selon la demande (Ton/heure), à travers la vitesse du moteur (tr/m) par le variateur de vitesse.

D'autre part le calcul de la vitesse se fait par le capteur de rotation, le volume et la surface de la matière sont stables.

Donc les étapes de travail sont :

- La demande de dosage de matière (T/ h)
- Présence de matière dans le convoyeur
- Calcul du poids de matière à chaque moment
- Calcul de la vitesse de moteur du convoyeur selon la demande et le poids de la matière
- Changement de la vitesse du moteur par le variateur si le poids de la matière ou la demande seront changés

## 2. Organigramme correspondant au cahier de charge

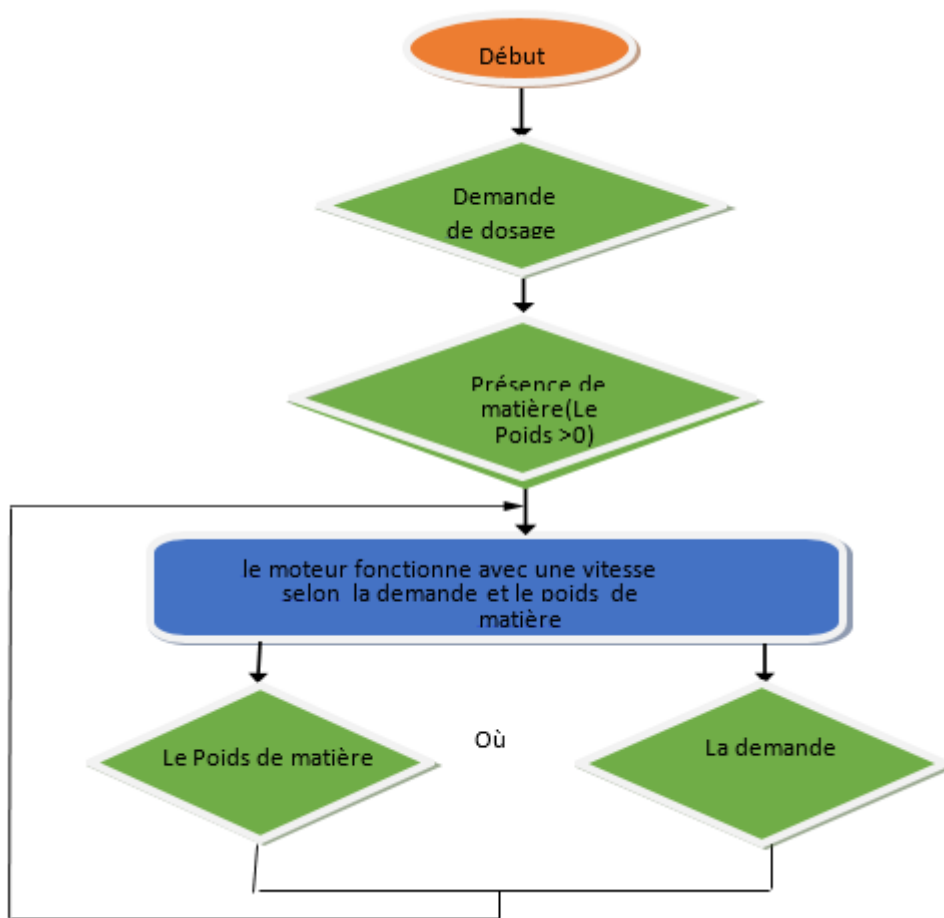


Figure 25 : Organigramme.

#### 4. Configuration du matérielle.

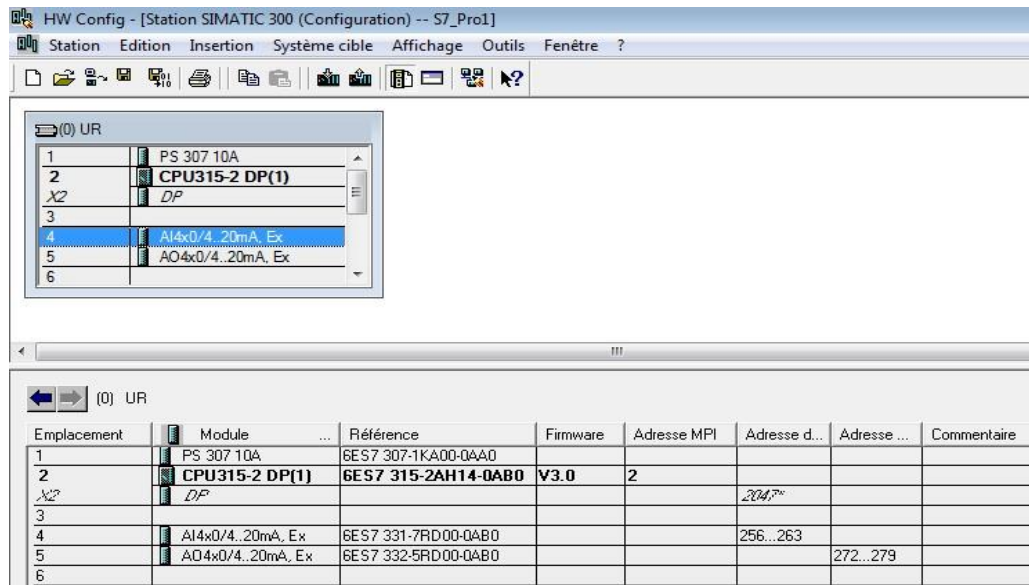


Figure 26: Configuration du matérielle.

#### 5. Table de mnémoniques.

La création d'un bloc d'organisation suit la phase de configuration matérielle où se fait le choix du langage utilisé et la définition des variables au niveau d'une table de mnémoniques.

	Etat	Mnémonique /	Opérande	Type de d	Commentaire
1		Alarme	A 0.0	BOOL	
2		bouton marche	M 0.1	BOOL	
3		capteur de rotation	E 0.0	BOOL	
4		Cycle Execution	OB 1	OB 1	
5		la dommande	MD 30	REAL	
6		la vitesse de mot...	MD 15	REAL	
7		niveau	MD 55	REAL	
8		poids de tapie 1	MD 10	REAL	
9		SCALE	FC 105	FC 105	Scaling Values
10		UNSCALE	FC 106	FC 106	Unscaling Values

Figure 27 : Table de mnémoniques

#### 6. Programmation en step7.

✓ FC1

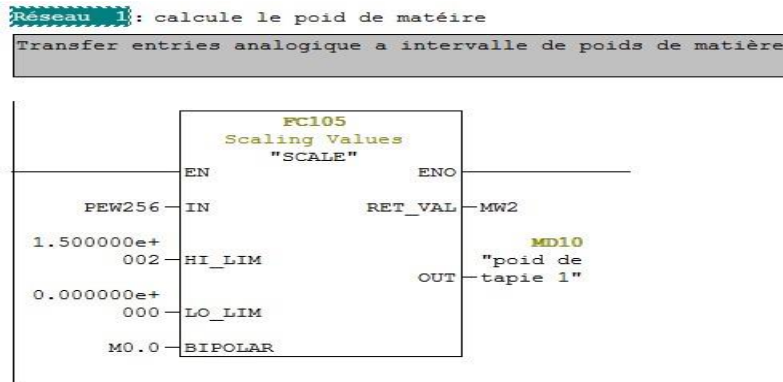


Figure 28 : Transformation d'une entrée analogique a un intervalle du poids de la matière.

✓ FC2

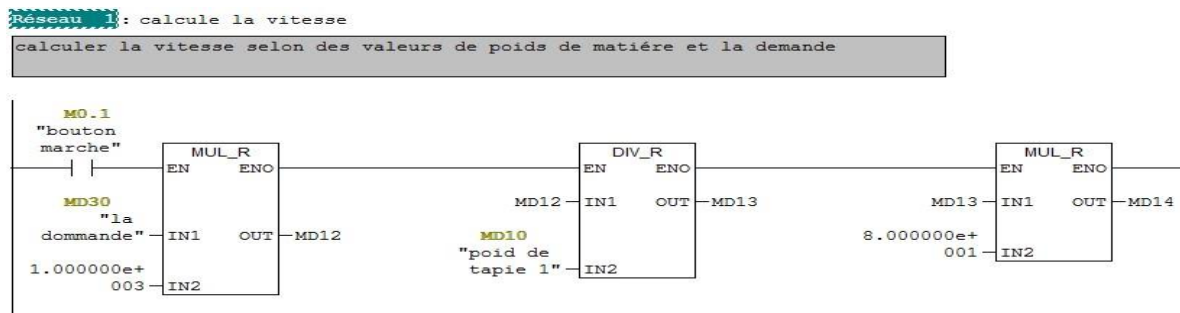


Figure 29 : Calcul de la vitesse selon des valeurs de poids de la matière et de la demande.

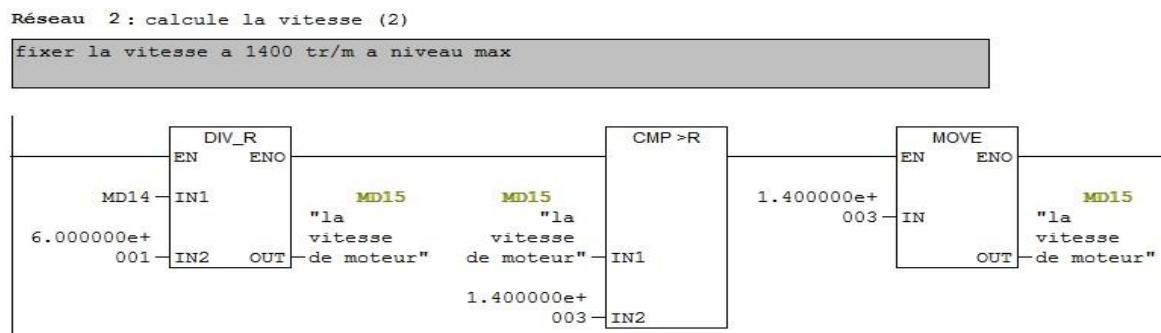


Figure 30 : Vitesse fixe à 1400 tr/m à niveau max.

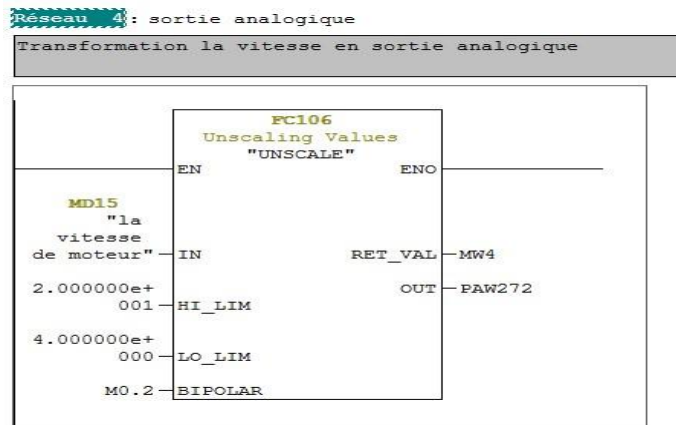


Figure 31: Transformation la vitesse en sortie analogique.

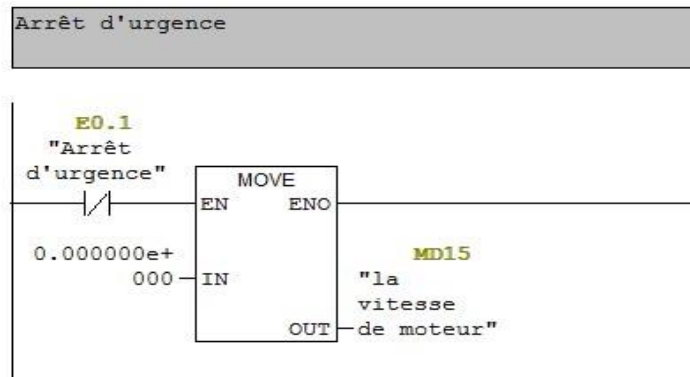


Figure 32 : Arrêt d'urgence.

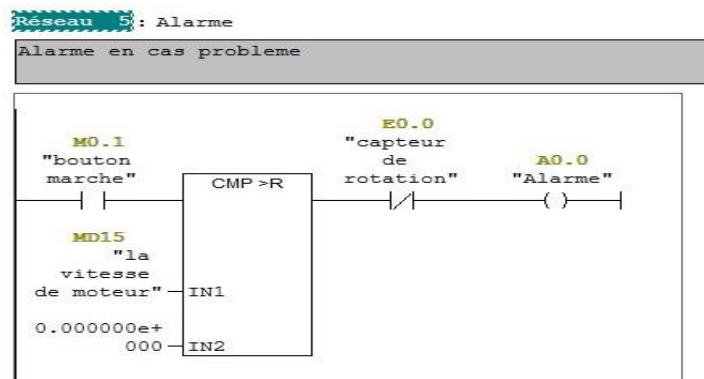
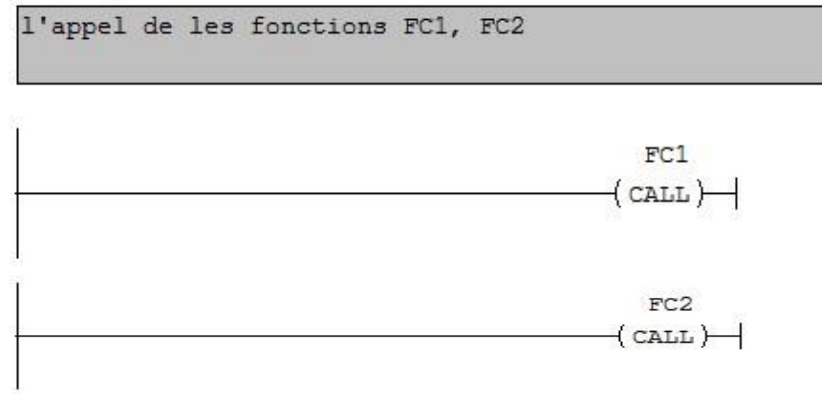


Figure 33 : Alarme en cas un problème.

✓ OB1



**Figure 34 : l'appel des fonctions FC1, FC2.**

## 7. Création destination HMI

Pour cette application, on a introduit un nouvel objet, qui est la station HMI, en choisissant le type de pupitre sur lequel, les informations seront transmises, on a utilisé un SIMATIC MP 270 10'' Touch.

Pour la gestion de ce projet on a suivi les étapes suivantes :

- ✓ Création d'une vue d'interface principale de notre application.

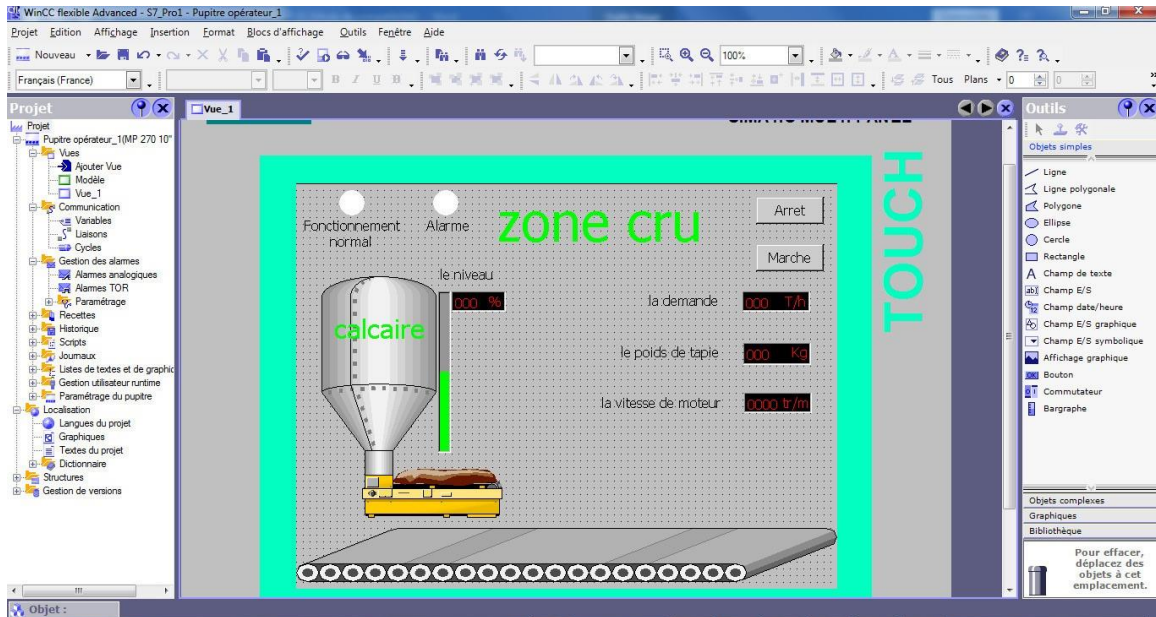


Figure 35: Vue globale dans la station SIMATIC HMI

- ✓ Lancement de la simulation avec Runtime.





Figure 36: Exécution du programme de simulation Le système 'Runtime'.

### Conclusion :

Ce chapitre regroupe les fonctionnalités essentielles du logiciel de programmation des pupitres de supervision et de contrôle de SIEMENS « le WINCC flexible », utilisé à programmer des interfaces IHM facilitant la communication entre l'homme et sa machine, comme on a détaillé un exemple d'application, nous avons réalisé une Interface Homme Machine (IHM) qui a comme objectif l'automatisation d'un système de Dosage. On a présenté l'environnement de programmation avec la configuration de logicielle de programmation Step7 et de supervision WinCC flexible, ainsi que l'ensemble des technologies utilisées dans la phase de mise en œuvre.

# **CHAPITRE III : Variateur de vitesse.**

## **Introduction**

Un variateur de vitesse est un dispositif destiné à régler la vitesse d'un moteur électrique a courant alternatif en faisant varier la fréquence et la tension, respectivement le courant délivré à celui-ci.

Les progrès de l'électronique de puissance ont permis de réduire le coût et la taille des variateurs de vitesse ces quatre dernières décennies. Ils ont concerné à la fois les interrupteurs semi-conducteurs utilisés, la topologie, les méthodes utilisées en contrôle commande et en simulation, ainsi que le matériel et les logiciels employés pour la commande.

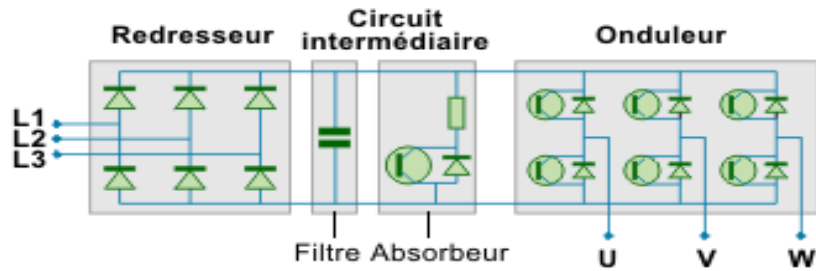
La solution passe par l'utilisation de variateur de fréquence, plus communément appelé variateur de vitesse. L'utilisation d'un tel convertisseur permet d'obtenir une plage de variation de vitesse allant de 0 ( $f = 0$  Hz) à la vitesse nominale du moteur ( $f = 50$  Hz). On peut même faire fonctionner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

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

### **III.1 Principe de fonctionnement d'un variateur de vitesse**

Le variateur de vitesse est composé essentiellement :

- 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



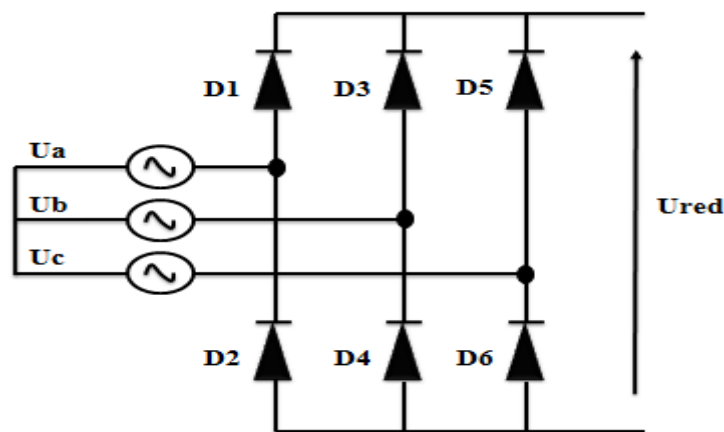
**Figure 37 : variateur de vitesse**

### III.1.1 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.

Nous utilisons le pont triphasé à diodes alimenté par un système de tensions sinusoïdales triphasées.

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



**Figure 38 : Redresseur triphasé à diodes**

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. La tension redressée est à tout instant la différence entre ces deux tensions, soit :

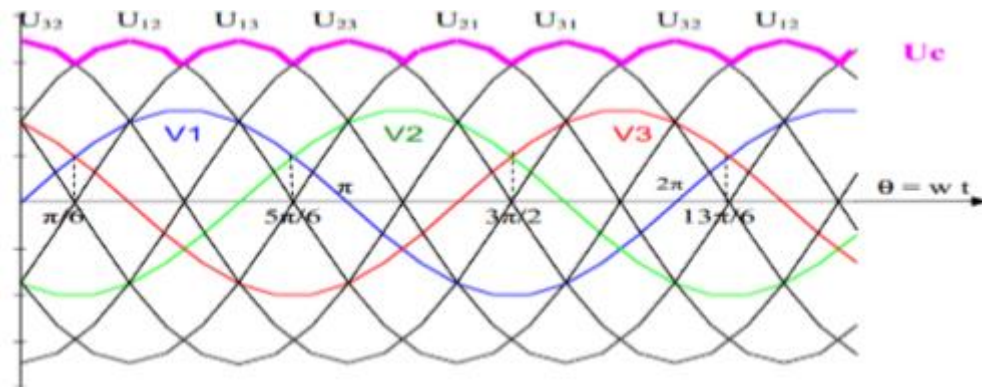


Figure 39 : tension redressée

La valeur moyenne de cette tension redressée :

$$U_{c\ moy} = \frac{1}{T} \int_0^T U_c dt = \frac{3}{\pi} \int_{\pi/6}^{\pi/2} V_m [\sin(\omega t) - \sin(\omega t - \frac{2\pi}{3})] d(\omega t) = \frac{3\sqrt{3}V_m}{\pi}$$

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

### III.1.2 Circuit intermédiaire :

Pour filtrer la tension continue délivrée par le pont redresseur, on utilise le circuit RC :

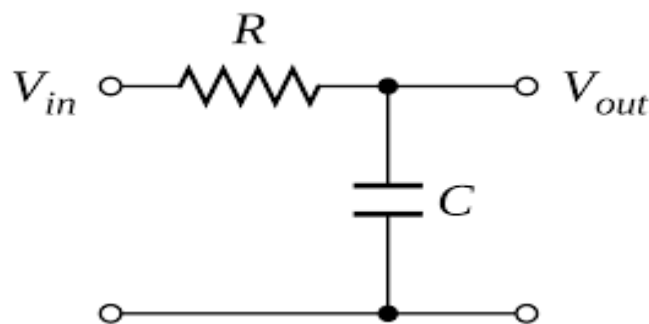


Figure 40 : schéma du principe de filtrage.

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.

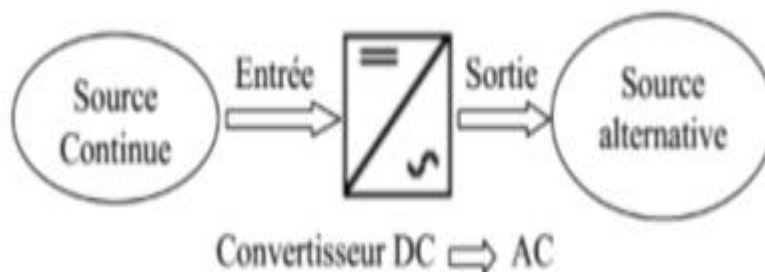
- Il évite les fluctuations importantes de la tension continue d'alimentation, que ces fluctuations viennent du redresseur (filtrage des harmoniques de la tension redressée) ou du débit instantané variable de l'onduleur, le maintien de cette tension continue apparaît comme un facteur favorisant la stabilité 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.

### III.1.3 L'onduleur :

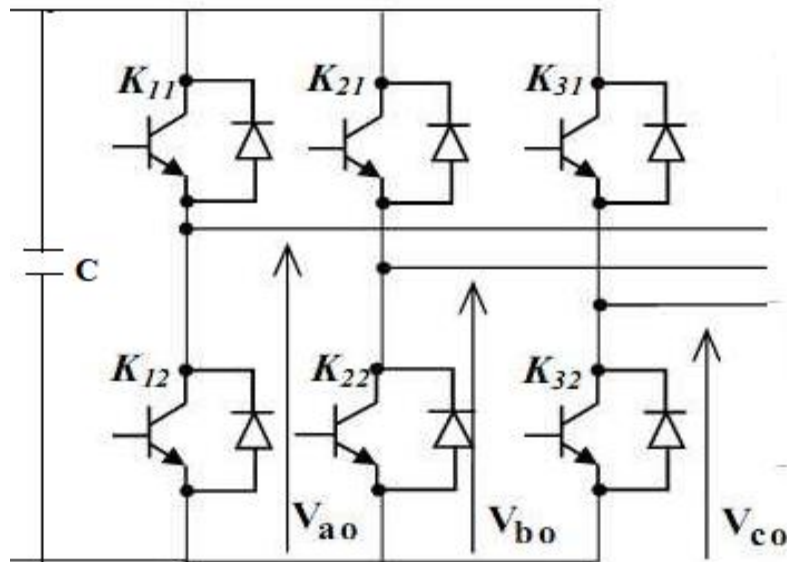
Un onduleur est un convertisseur statique assurant la conversion d'énergie électrique de la forme continue (DC) à la forme alternative (AC). En fait, cette conversion d'énergie est satisfaite au moyen d'un dispositif de commande (semi-conducteurs). Il permet d'obtenir aux bornes du récepteur une tension alternative réglable en fréquence et en valeur efficace, en utilisant ainsi une séquence adéquate de commande



**Figure 41 : schéma de principe d'un onduleur.**

On peut réaliser un onduleur triphasé en regroupant, en parallèle, trois onduleurs monophasés (en pont ou en demi-point) et commander les interrupteurs de chacun pour obtenir à la sortie trois phases décalées de  $120^\circ$ . En fait, en regroupant trois demi ponts monophasés, on obtient un onduleur en pont triphasé à six interrupteurs représenté par la figure 48 dont les interrupteurs du même bras de l'onduleur doivent être complémentaires pour que la tension continue  $U$  ne soit jamais en court-circuit et pour que les circuits des courants  $i_a$ ,  $i_b$  et  $i_c$  ne soient jamais ouverts. Afin que les interrupteurs puissent imposer les tensions de sortie, quelques soient les courants de charge, il faut que ces derniers soient bidirectionnels en

courant. Chacun d'eux est formé d'un semi-conducteur à ouverture et à fermeture commandés et d'une diode montée en antiparallèle. Ces interrupteurs peuvent être réalisées, suivant la puissance à contrôler, avec des transistors MOS ou bipolaires, des IGBTs ou des GTOS associés à une diode en antiparallèle pour obtenir la réversibilité en courant.



**Figure 42: circuit de puissance d'un onduleur de tension**

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

Pour faire face à ce problème, il faut faire fonctionner l'onduleur en M.L.I.

### III.2 Alimentation d'un moteur asynchrone triphasé a fréquence variable :

#### Etude du couple dans le cas où $U_{moteur}$ est constante :

La fem aux bornes d'un enroulement est de la forme :

$$U = E = 4.44 * B_m * N * f * S$$

Avec :

f : fréquence d'alimentation du moteur.

S : section du fer (m<sup>2</sup>).

$B_m$  : Induction dans le moteur (T).

N : nombre de conducteurs.

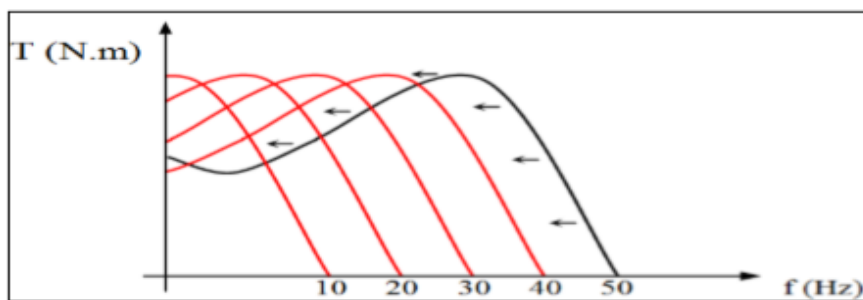
U : tension aux bornes du moteur (V).

- Si  $f$  varie,  $U$  étant constant,  $B_m$  va varier pour garder l'égalité dans la relation.
- Si  $f$  diminue,  $B_m$  va augmenter et va saturer le circuit magnétique, provoquant un échauffement du moteur et surtout une baisse du couple moteur.

Donc si l'on souhaite garder  $B_m$  constant, tout ne faisant varier la fréquence, **il faut varier  $U$  pour maintenir constant le rapport  $U/f$ .**

**Etude du couple dans le cas ou  $U/f = \text{constante}$  :**

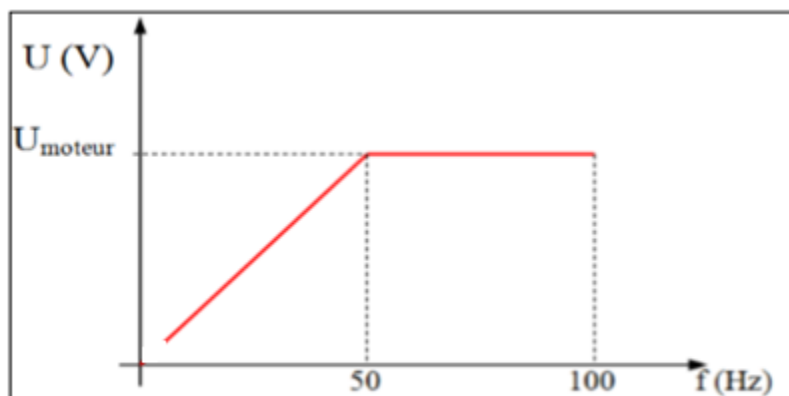
Dans ces conditions, les caractéristiques du couple moteur pour différentes fréquences d'alimentation opèrent une translation sur la gauche comme il est montré dans ci-dessous



**Figure 43: Couple du moteur pour une fréquence variable**

La figure délivre au moteur une tension et une fréquence proportionnelle jusqu'à la valeur de 50 Hz.

Pour des fréquences supérieures à 50hz, la tension du moteur ne pouvant plus augmenter, (l'enroulement est alimenté sous sa tension nominale) le rapport  $U/f$  diminue, le flux décroît, entraînant une diminution du couple maximum.



**Figure 44 : Tension du moteur en fonction de la fréquence**



### III.3 Variateur de vitesse Altivar 312

#### III.3.1 Description

Le variateur Altivar 312 de Schneider Electric est un convertisseur de fréquence pour des moteurs asynchrones triphasés de 200 à 600v et de 0.18 à 15kW.

Le variateur Altivar 312 est robuste, compacte et facile à installer.

Ses fonctions intégrées sont particulièrement adaptées pour répondre aux applications de machines industrielles simples.

La prise en compte des contraintes de mise en œuvre et d'utilisation du produit dès la conception permet de proposer une solution économique et fiable aux constructeurs de machines simples et aux installateurs. Avec ses différentes cartes de communication disponibles en option, le variateur Altivar 312 s'intègre parfaitement dans les principales architectures d'automatisme. [8]

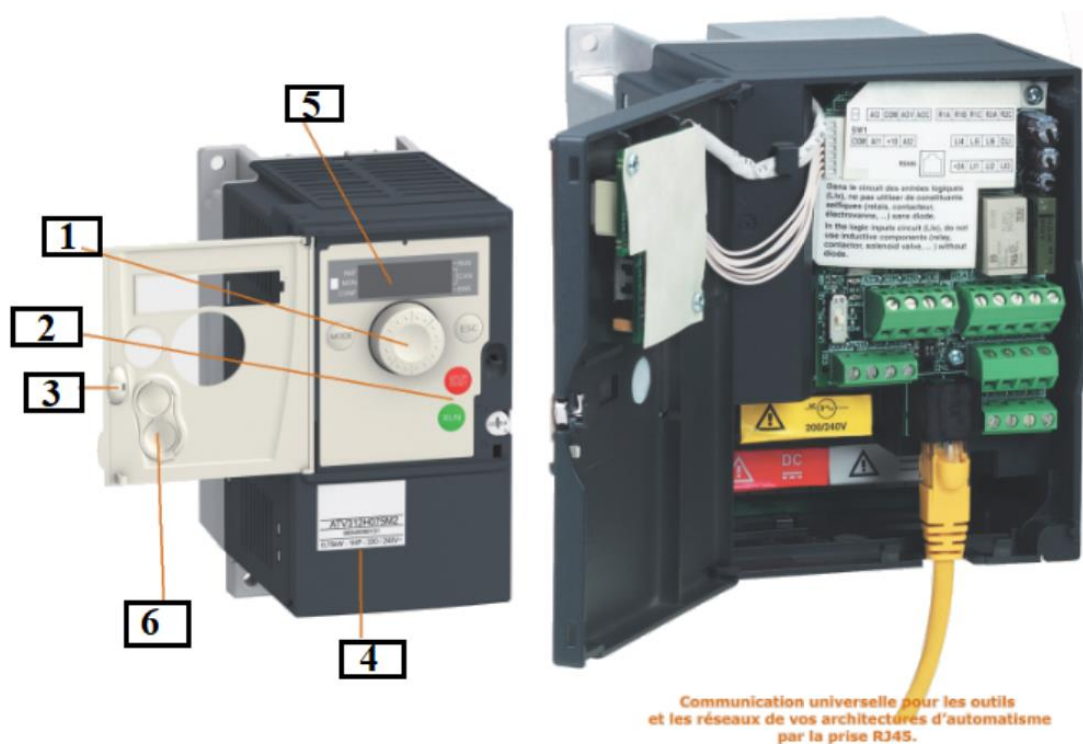


Figure 45: variateur de vitesse Altivar 312 [13]

(1) Bouton de navigation simplifiée : Il permet de naviguer dans les menus, de modifier les valeurs et de modifier la vitesse du moteur en mode local.

(2) Commandes Marche-Arrêt : Les touches “RUN” et “STOP/RESET” permettent de commander la marche et l’arrêt du moteur en mode local. Ces deux touches peuvent être rendues accessibles en face avant en ôtant l’obturateur (6) de la porte.

(3) Verrouillage possible par plombage.

(4) Zone de marquage.

(5) Afficheur à 4 digits permet d’afficher les états, les défauts et les valeurs des paramètres du variateur. [9]

Le variateur Altivar 312 est équipé d’une carte entrées/sorties de contrôle qui intègre :

– Un bornier entrées/sorties comprenant :

- Six entrées logiques : LI1 à LI6.
- Trois entrées analogiques : AI1 à AI3.
- Deux sorties analogiques : AOV et AOC (l’une ou l’autre utilisable).
- Deux sorties à relais : R1 et R2.
- Un port de communication Modbus/CANopen, accessible sur un connecteur de type RJ45

Toutes les fonctions du variateur Altivar 312 sont accessibles par les bus et les réseaux de communication :

- La commande.
- La surveillance.
- Le réglage.
- La configuration.

La commande et la consigne de vitesse peuvent provenir de différentes sources de contrôle

- Borniers d’entrées logiques ou d’entrées/sorties analogiques.
- Bus ou réseau de communication.
- Terminal déporté.

Les fonctions avancées du variateur Altivar 312 permettent de gérer la commutation de ces sources de contrôle suivant les besoins de l’application.

Il est possible de choisir l’affectation des données d’entrées/sorties périodiques de communication par le logiciel de configuration du réseau. Le variateur Altivar 312 est piloté suivant le profil natif CiA 402.

La surveillance de la communication est effectuée selon des critères spécifiques à chaque protocole. Cependant, quel que soit le protocole, il est possible de configurer la réaction du variateur, suite à un défaut de communication :

- Arrêt roue libre, arrêt sur rampe, arrêt rapide ou arrêt freiné.
- Maintien du dernier ordre reçu.
- Position de repli à une vitesse prédéfinie.
- Ignorer le défaut. [9]

### III.3.2 Schéma de raccordement :

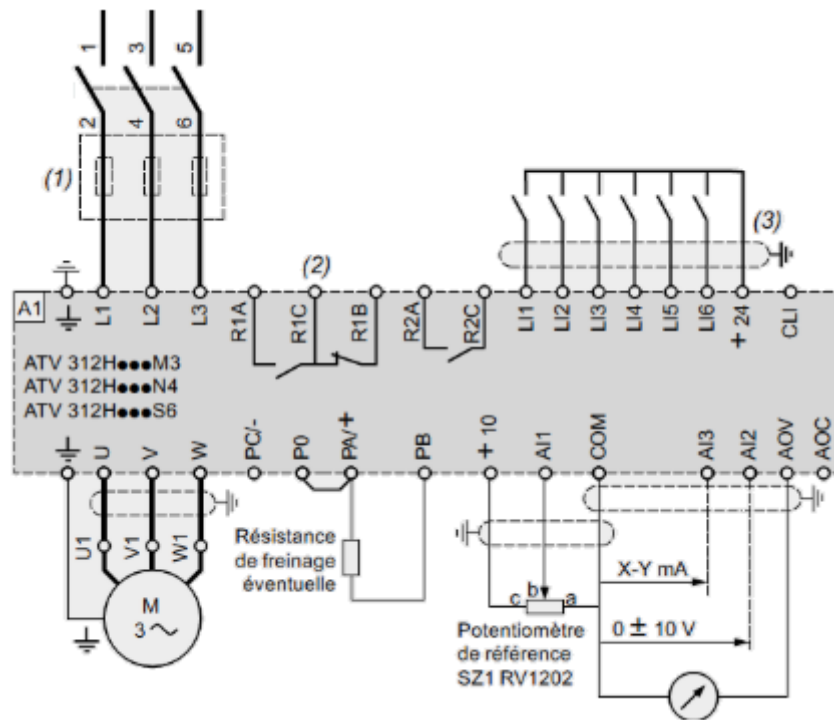


Figure 46: schéma de raccordement d'un Altivar 312 [10]

### III.3.3 Consigne de vitesse :

**La consigne de vitesse peut avoir différentes sources en fonction de la configuration du variateur :**

- Les consignes issues des 3 entrées analogiques.
- La consigne du bouton de navigation.
- La fonction plus vite/moins vite par entrée logique, avec les touches du clavier ou du terminal déporté.
- La consigne du terminal déporté.
- Les consignes de vitesse issues des réseaux ou bus de communication.

Ces différentes sources sont gérées par programmation des fonctions et des canaux de consignes.

### **Entrées analogiques :**

Il existe 3 entrées analogiques.

2 entrées en tension :

- 0-10 V (AI1).
- $\pm 10$  V (AI2).

1 entrée en courant :

- X-Y mA (AI3) avec X configurable entre 0 et 20 mA, et Y configurable entre 4 et 20 mA.

### **Entrées sommatrices :**

Permet d'additionner 2 à 3 consignes de vitesse de sources différentes.

Les consignes à additionner sont à choisir parmi tous les types de consigne de vitesse possibles. Exemple :

- Consigne 1 issue de AI1.
- Consigne 2 issues de AI2.
- Consigne 3 issue de AIP. Consigne de vitesse du variateur = consigne 1 + consigne 2 + consigne 3.

### **III.3.4 Paramétrages du Variateur de vitesse ATV 312 :**

Les paramètres permettent d'adapter le variateur à une application bien déterminée.

Chaque paramètre est identifié par un numéro de paramètre, un texte de paramètre et par des attributs spécifiques. Le numéro de paramètre est unique dans un système de variateur.

L'accès aux paramètres de l'ATV 312 est possible moyennant les panneaux de commande et logiciels suivants :

- **Configuration Locale à travers le panneau de commande de l'ATV 312 :**

Le panneau de commande sert au paramétrage, à la commande et à la surveillance de variateur et moteurs directement au pied de l'appareil. Il fait partie de l'appareil en version de base. Il comporte une visualisation à quatre positions et cinq touches.

Il est utilisé pour le paramétrage d'applications simples ne comptant qu'un petit nombre de paramètres à régler ainsi que pour le paramétrage rapide en cours de fonctionnement.

## Fonctions de l'afficheur IHM et des touches

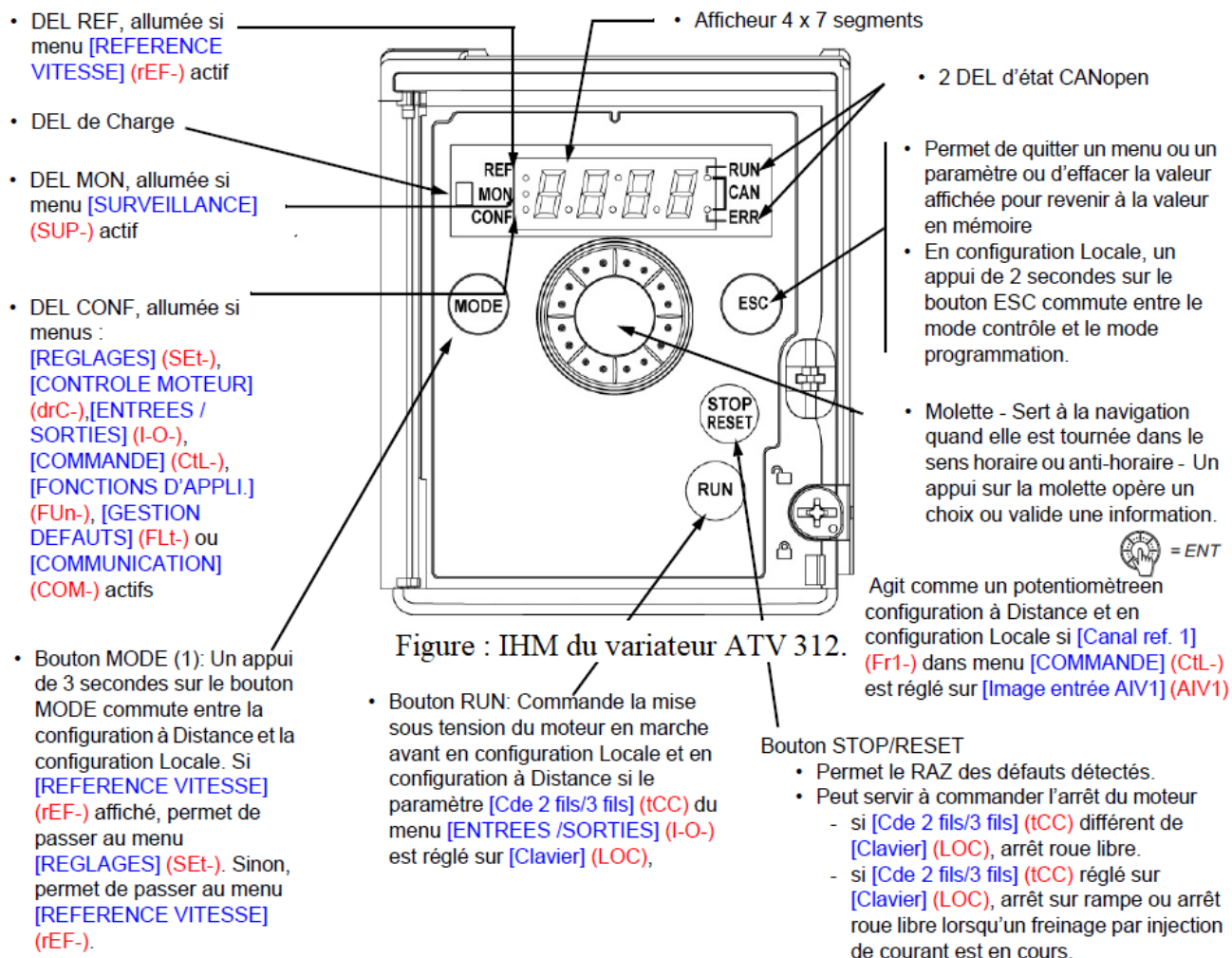


Figure 47 : Réglage du variateur de vitesse

- **InIt**: Séquence d'initialisation.
- **rdY**: Variateur prêt.
- **dCb**: Freinage par injection de courant continu en cours.
- **nSt**: Arrêt en roue libre.
- **FSt**: Arrêt rapide.
- **tUn**: Auto-réglage en cours.

### III.3.5 Les menus du variateur ATV312 : [13]

L'ensemble des paramètres est accessible dans la navigation à travers 8 menus aux fonctions distinctes. A partir de l'écran de départ (qui affiche l'état du variateur, « rdy » pour indiquer qu'il est prêt à être utilisé), on peut appuyer sur la touche « ENT » pour accéder aux menus que l'on fait défiler avec les flèches.

On présente dans le tableau les principaux menus du variateur :

Menus	Description
SET -	Réglage de principaux paramètres de vitesse (LSP, HSP) et d'accélération (ACC, DEC)
drC -	Réglage des paramètres liés au moteur
CtL -	Choix des canaux de commande (principalement accès aux fonctions avancées, affectation des consignes analogiques ...)

**Tableau 2: Principaux menus du variateur ATV312 [11]**

### Menu SET-

Le menu Set- permet d'effectuer quelques réglages de base parmi un grand nombre de paramètres (également accessibles dans d'autres menus). Les quatre principaux paramètres sont :

Paramètre	Description
ACC	Temps de la rampe d'accélération (0.1 à 999.9 s). Défini pour accélérer entre 0 et la fréquence nominale FrS
dEC	Temps de la rampe de décélération (0.1 à 999.9 s). Défini pour accélérer entre la fréquence nominale FrS et 0
LSP	Petite vitesse, peut aller de 0 à la fréquence moteur à consigne minimum
HSP	Grande vitesse, peut aller de LSP à la fréquence moteur à consigne maximum. (Il faut s'assurer que ce réglage convient au moteur et à l'application)

**Tableau 3: principaux paramètres du Set- [11]**

- **Menu contrôle moteur drC- :**

Ce menu permet de fixer les paramètres du moteur commandé par le variateur.

Paramètre	Description
bFr	Fréquence standard du moteur. Ce paramètre modifie les préreglages des paramètres : HSP, Ftd, FrS et tFR
UnS	Tension nominale lue sur le moteur lue sur sa plaque signalétique
FrS	Fréquence nominale du moteur lue sur sa plaque signalétique
nCr	Courant nominal du moteur lu sur sa plaque signalétique
nSP	Vitesse nominale moteur lue sur sa plaque
COS	Cosinus Phi du moteur lu sur sa plaque signalétique (de 0.5 à 1)

**Tableau 4: principaux paramètres du drC-[11]**

- **Menu commande CtL- :**

## CHAPITRE III : Variateur de vitesse.

Ce menu permet de définir les canaux de commande et de consigne.

Les commandes (marche, arrêt ...) peuvent être données par :

-les entrées logiques du bornier (pupitre opérateur reliés aux bornes LI1 à LI6).

- le clavier (RUN/STOP) en façade du variateur.

-un terminal déporté, les signaux Modbus et CANopen.

Paramètre	Description
LAC	Niveau d'accès aux fonctions : -L1 : fonctions standards. -L2 fonctions avancées du menu FUN (plus vite/ moins vite/ fin de course ...) -L3 gestion des modes de commandes mixtes
FrI	Configuration de la consigne 1 : AIP : potentiomètre de façade. AI1 : Entrée analogique AI1 AI2 : Entrée analogique AI2 AI3 : Entrée analogique AI3 (Même options de réglages pour Fr2, configuration de la consigne 2)
rFC	Commutation de consigne : le paramètre rec. permet de choisir le canal Fr1 ou Fr2 ou de configurer une entrée logique ou un bit de commande pour commuter Fr1 ou Fr2

**Tableau 5: principaux paramètres du CtL- [11]**

### • **Menu fonction d'applications Fun- :**

On trouve dans la menu fonction d'applications Fun les types de freinage et ce variateur dispose de quatre types de freinage :

rMp : Sur rampe

FSt : Arrêt rapide

nSt : Arrêt roue libre

dCI : Arrêt par injection de courant continu. (Que l'on a choisi pour freiner le moteur).

### **III.3.6 Paramétrages du Variateur de vitesse ATV 312 :**

Les paramètres permettent d'adapter le variateur à une application bien déterminée.

Dans cette partie nous allons aborder le paramétrage qu'on a effectué pour adapter le variateur à notre application.

Pour faire tourner le moteur :

- On a ponté la borne **AI1**(entrée analogique du variateur) sur la borne sortie analogique de l'automate S7 300.

- On a ponté la borne **LI1** sur la sortie digitale de l'automate pour activer le sens avant.

➤ **Sens avant**

Dès que LI1 sera activée, le variateur va démarrer immédiatement le moteur dans le sens Avant et lui faire atteindre la vitesse programmée dans le cahier des charges.

➤ **Sens arrière**


Pour faire tourner le moteur en sens arrière, on a ponté la borne LI2 sur une sortie digitale de l'automate et l'on a activée.

➤ **Rampes accélération et décélération**

Quel que soit le sens, le variateur respectera les rampes d'accélération et de décélération des paramètres **ACC** et **DEC** que l'on a réglées à 3 secondes.

**Pour diminuer ou augmenter la vitesse, les rampes d'accélération et de décélération**

On est allé dans le menu SET et modifier les paramètres LSP, HSP, ACC et DEC. Exemple pour le paramètre HSP :

Menu SET  = 40 (si la vitesse souhaitée est de 40Hz)

➤ **Utilisation des vitesses présélectionnées.**

Les variateurs de vitesse Schneider sont préconfigurés en vitesses présélectionnées. Nous ne détaillerons pas cette partie, seulement un exemple simple à utiliser :

**Description**

- **Vitesse lente de 5 Hz** si l'entrée **LI3** est **activée**
- **Vitesse moyenne de 10 Hz** si l'entrée **LI4** est **activée**
- **Vitesse rapide de 15 Hz** si **LI3 et LI4** sont **activées**
- **Vitesse variable entre 0 et 15 Hz** si les entrées **LI3 et LI4** sont **désactivées**, vitesse donnée par le potentiomètre.

Quel que soit la vitesse choisie, il est **nécessaire** que l'ordre de Marche Sens Avant **LI1** ou l'ordre de Marche sens arrière **LI2** soient **activés**.



➤ **Paramétrage**

On a vérifié si les réglages usine suivants sont toujours respectés :

Menu SET  **LSP = 0**      **HSP = 50**

Menu FUN  **PSS**      **PS2 = LI3**

Menu FUN  **PSS**      **PS4 = LI4**

Et on a forcé les paramètres suivants :


Menu FUN  **PSS**      **SP2 = 5 Hz**

Menu FUN  **PSS**      **SP3 = 10 Hz**

Menu FUN  **PSS**      **SP4 = 15 Hz.**

➤ **La protection moteur.**


Pour protéger le moteur, on a réglé la protection thermique du variateur en mémorisant l'intensité nominale qu'on a trouvé sur la plaque signalétique du moteur sur le paramètre Ith du variateur

Menu SET  **Ith = I nominale Moteur (10 A).**

➤ **Optimisation de l'ensemble variateur / moteur.**

Cette opération est fortement conseillée, elle permet d'optimiser les courants (les réduire) ce qui évite les échauffements et permet des économies d'énergie.

Pratiquement, il s'agit d'aller lire les informations de la plaque signalétique moteur et de les copier dans le variateur, puis de lancer le Tun ou autoréglage.

Menu drC  **UnS = tension nominale **plaquée** moteur (380 V dans notre cas).**

**FrS**= Fréquence nominale **plaquée**moteur (50 Hz).

**nCr** = Intensité nominale **plaquée** moteur (10 A).

**COS** = cosinus phi du moteur **plaqué** moteur (0.9).


tUn = YES : valider avec la touche ENT pendant au moins 2 sec. Si l'opération d'autoréglage s'est bien passée, tUn évolue à Done (fait).

### **Précision importante**

Le constructeur Schneider livre les variateurs neufs pré-paramétrés en « **Réglage usine** ».

À tout moment, il est possible de retourner au réglage usine :

Menu CtLCFG = Std

 **FCS = InI** (le paramètre repasse à no à la fin)

Cette fiche considère que :

- Les paramètres sont en **réglage usine (cas d'un produit neuf sortie carton)**.
- Le moteur à entrainer est un moteur à **cage d'écureuil classique 50 Hz**.

### **Conclusion :**

Le variateur de vitesse étant intimement lié à la nature de la charge entraînée et aux performances visées, toute définition et recherche d'un variateur de vitesse doivent passer par une analyse des exigences fonctionnelles de l'équipement puis des performances requises pour le moteur lui-même.

En consultant la documentation du moteur fournies par le constructeur et selon sa plaque signalétique, nous avons introduit les différentes caractéristiques : la puissance du moteur, le courant et la tension nominale, le facteur de puissance... ce qui permet des économies d'énergie.

Toutes les informations réunies dans ce chapitre nous ont permis d'aboutir à un bon choix des paramètres de configuration du variateur de telle sorte qu'il soit le plus adaptés à notre manipulation.

# **CHAPITRE IV : La réalisation de l'armoire électrique**

### **Introduction :**

Le projet « entrainement du chariot » est conçu dans le but de réaliser plusieurs applications. Tout au long de ce chapitre, nous allons décrire notre prototype tout d'abord d'un point de vue instrumentation, à savoir les actionneurs utilisés. Ensuite, nous décrirons les cahiers des charges des manipulations effectuées. Nous finirons par décrire les programmes sur STEP7 ainsi que la supervision de ces derniers sur WinCC flexible.

### **IV.1 Structure de la Tâche d'automatisation**

Dans ce qui suit, un aperçu des différentes étapes de la réalisation du projet et des objets que nous avons créé dans le cadre du projet "Déplacement chariot " au sein du logiciel STEP 7.

Le projet en gros comprend les étapes de configuration suivantes :

- Détermination des éléments de l'application pour finaliser le cahier des charges ;
- Détermination des éléments auxiliaires de l'installation tels que les contacteurs, les interrupteurs, les voyants, l'arrêt d'urgence et les boutons poussoir comme entrée ;
- Création du projet "Déplacement chariot" ;
- Insertion et configuration des matériels ;
- Programmer API ;
- Paramétrage de la communication.
- Visualisation du processus ;
- Tester en ligne les manipulations effectuées sur le déplacement du chariot.

### **IV.2 Description de l'installation de déplacement du chariot :**

Notre prototype a été élaboré suivant les phases d'un projet, ces phases sont le design, le développement et la construction.

Les éléments et composants utilisés pour la construction de ce dispositif, ont été choisis en fonction des paramètres : qualité, disponibilité et coût.

La figure suivante donne une vue générale de l'installation, qui permet de réaliser de différentes manipulations citées auparavant.

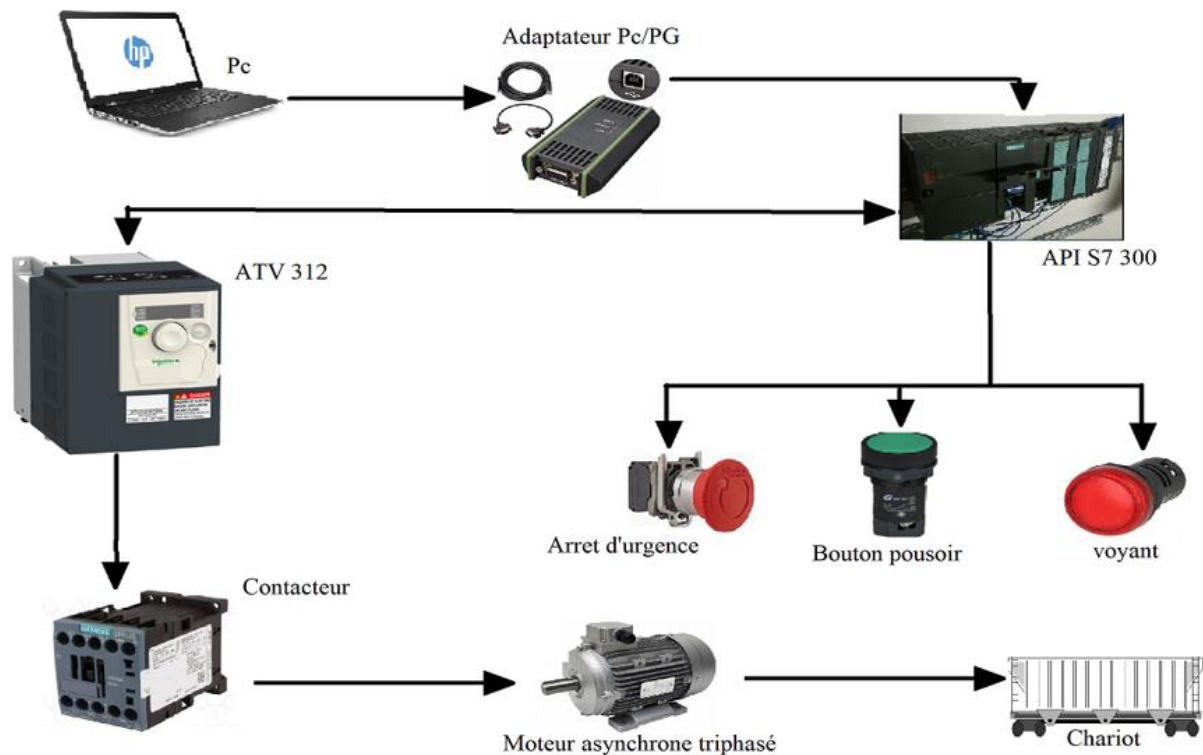
Le principe de fonctionnement sera décrit en détail dans le cahier des charges établi.

L'ensemble des éléments et les caractéristiques du prototype réalisé sont présentés comme suit:

- Le prototype repose sur une table de 1.50 m sur 1m et d'une hauteur de 1.50m qui comporte tous les organes de l'installation.
- Une armoire de 0.15m sur 0.15m sur une hauteur de 0.30m/0.25m qui contient les différents composants électriques de commande et de puissance.

## CHAPITRE IV : La réalisation de l'armoire électrique

- Un moteur asynchrone triphasé, couplé en étoile alimenté à 380 V.
- Un variateur de vitesse ATV 31 qui améliore les performances de démarrage et permet une variation de vitesse contrôlé du moteur.
- Une alimentation stabilisée (PHOENIX CONTACT UNO POWER 220V AC/24V DC), alimenté en monophasé 220 V et qui délivre du 24V.
- Une station PC pour la supervision.
- Un contacteur lié au bouton d'arrêt d'urgence pour assurer la coupure totale de l'alimentation de l'armoire en cas de problèmes.
- Un contacteur servant comme interface de sortie entre l'automate et le moteur.
- Un disjoncteur triphasé pour la sécurité de toute l'installation.
- Un disjoncteur monophasé pour le circuit de commande.



**Figure 48 : Synoptique du prototype.**

Remarque : Chaque composant, cité précédemment, est détaillé dans les annexes liées au rapport.

### IV.3 Les éléments de commande et de supervision utilisés

Pour la réalisation du projet « déplacement chariot », on a utilisé les éléments suivants :

- Un automate Siemens de la gamme SIMATIC S7-300 doté des caractéristiques suivantes :
  - CPU 314 IFM, doté d'une mémoire de travail 48Ko.
  - Un module d'entrées logiques DI16/DO16.
  - Un module d'entrées/sortiesAI4/AO1.
  - Un port MPI.
  - Une alimentation S307 2A alimenté en monophasé 220 V et qui délivre du 24V.
- Un variateur de vitesseSchneider ATV312.
- PC, système d'exploitation Windows
- Logiciel STEP7 V 5.5 + WinCC flexible.
- Interface MPI pour PC (adaptateur PC)
- Moteur asynchrone triphasé, avec une charge constante pour la simulation.

### **IV.4 Mise en œuvre de l'application : « Entraînement d'un chariot »**

On va mettre en œuvre une application d'entraînement d'un chariot sur un plan horizontale par un moteur asynchrone triphasé.

Le moteur asynchrone à courant alternatif va être commandé par le variateur de vitesse ATV 312.

Une station SIMATIC S7-300 (CPU314 IFM) va être utilisée pour la gestion de l'installation.

#### **IV.4.1Description des cahiers de charge.**

Le fonctionnement de l'installation est géré par l'automate, cependant l'opérateur peut intervenir pour changer sa configuration et pour se faire, l'opérateur dispose soit des boutons placés sur le pupitre (système PC) soit sur le logiciel car il y aura une interface pour éventuellement changer et observer l'état de l'installation.

L'Automate Siemens S7-300 aura pour rôle essentiel de gérer les actionneurs tels que le moteur, le variateur de vitesse et les éléments de signalisation (les voyants). Il assurera la sécurité de l'installation afin d'éviter toutes fausse manipulation de la part de l'opérateur, mais aussi il permettra la communication entre les équipements et le superviseur.

IV.4.2 Grafcet et définition des cahiers de charge des 3 manipulations :

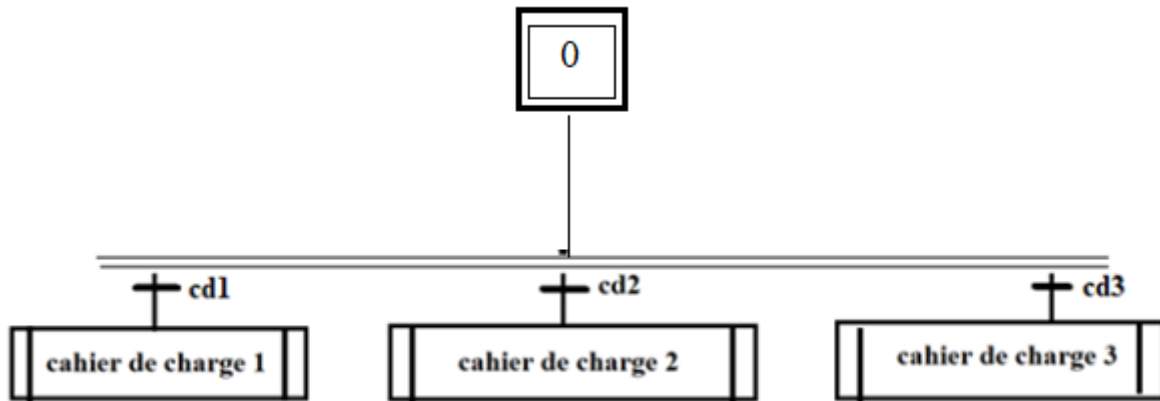


Figure 49: Diagramme des deux manipulations.

**Manipulation 1 :** « cahier de charge n°1 ».

La figure suivante montre le grafcet de la manipulation 1 :

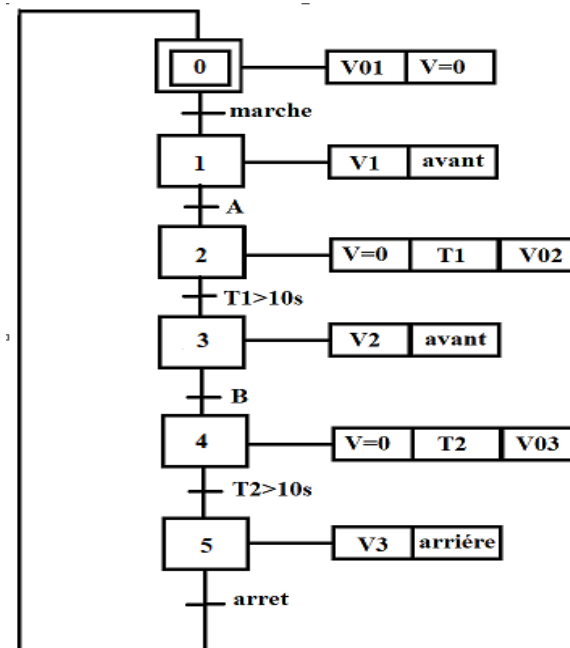


Figure 50: Grafcet de la manipulation 1 « Cahier de charge n°1 ».

**Description :**

## CHAPITRE IV : La réalisation de l'armoire électrique

En appuyant de manière brève sur le bouton "MARCHE " le chariot COMMENCE à se déplacer en avant avec une vitesse V1 et dès que le bouton "A " sera appuyé le chariot s'arrête et un TIMER " T1" sera lancer.

Après l'écoulement des 10s programmer du TIMER "T1", le chariot démarre avec une vitesse "V2" toujours en avant jusqu'à ce que le bouton "B " soit appuyer de manière brève et un autre TIMER "T2" sera activer.

Une fois les 10s du TIMER écoulé, le chariot se déplace en arrière avec une vitesse " V3 ". En appuyant sur la touche Arrêt, le déplacement du chariot est stoppé.

Dans la phase de marche avant le chariot accélère jusqu'à la vitesse requise. Quand le bouton " A" est appuyé, il entre dans la phase de freinage. Le chariot sera donc freiné jusqu'à ce qu'il soit arrêté, et de même pour les deux autres phases du déplacement du chariot.

Voici les normes à respecter pour cela :

- La durée pour atteindre le régime haut de fréquence (50Hz) doit valoir 10 secondes ;
- Le moteur doit s'arrêter en 3 secondes (pour une fréquence de 5Hz) ;
- Le chariot se déplace conformément à la fréquence F1 = 30Hz pour la première phase ;
- Le chariot se déplace conformément à la fréquence F2 = 50Hz pour la deuxième phase;
- Lors de la phase de marche arrière (phase3) , la nouvelle consigne devient F3 = 5Hz.

En détaillant chaque phase comme suit :

### **Marche avant du chariot :**

- Actionnez le bouton « Marche » ; Le moteur démarre et atteint la fréquence requise F1=30Hz en 6 secondes ;
- Actionnez le bouton "A" ; Le moteur freine et s'arrête en 7,5 secondes ;
- Le TIMER T1=10s activer ;
- Temps de 10s écouler, Le moteur démarre et atteint la fréquence requise F1=30Hz en 6 secondes ;
- Actionnez « B » ; le moteur s'arrête en 3 secondes environ (durée de lissage).
- Le TIMER T1=10s activer ;
- 

### **Marche arrière du chariot :**

- Temps de 10s écouler, Le moteur démarre mais en sens contraire et atteint la fréquence requise F1=30Hz en 6 secondes ;
- Actionnez le bouton "Arrêt";Le moteur s'arrête au bout d'environ 3 secondes (Temps de lissage).



- Le chronogramme suivant illustre le fonctionnement exigé dans le cahier de charge :

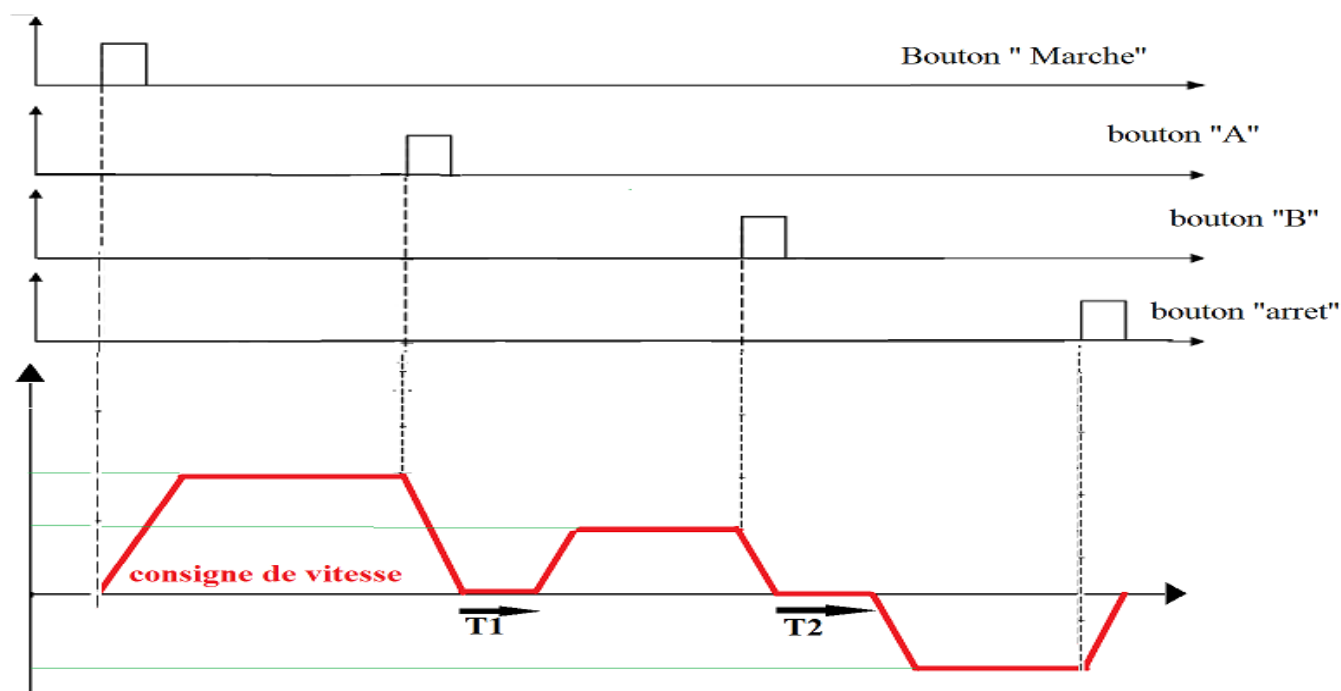


Figure 51: Chronogramme de fonctionnement

**Manipulation 2 : « cahier de charge n°2 ».**

La figure suivante montre le grafcet de la manipulation 2 :

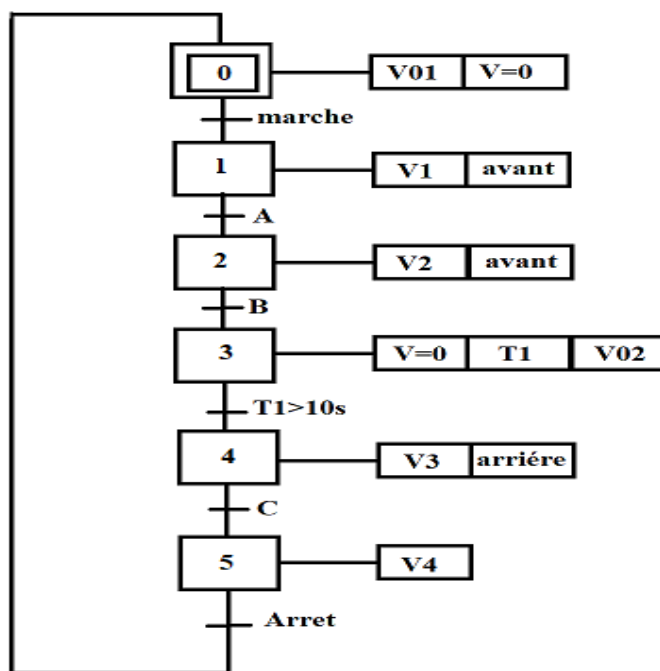


Figure 52 : Grafcet de la manipulation 2 « Cahier de charge n°2 ».

**Description :**

En appuyant de manière brève sur le bouton "MARCHE " le chariot commence à se déplacer en avant avec une vitesse  $V1$  et en appuyant sur le bouton "A " le moteur commence à décélérer jusqu'à ce qu'il atteigne la vitesse  $V2$ , si on actionne le bouton "B le chariot s'arrête et un TIMER " T1" sera lancer.

Après l'écoulement des 10s programmé du TIMER "T1", le chariot se déplace en arrière avec une vitesse "  $V3$  "et dès que le bouton "C " sera appuyé le moteur décélère. En appuyant sur la touche Arrêt, le déplacement du chariot est stoppé.

En détaillant chaque phase comme suit :

**Marche avant du chariot :**

- Actionnez le bouton « Marche » ; Le moteur démarre et atteint la fréquence requise  $F1=30\text{Hz}$  en 6 secondes ;
- Actionnez le bouton "A" ; Le moteur décélère et atteint la fréquence  $F2=20\text{Hz}$  ;
- Actionnez le bouton "B" ; Le moteur freine et s'arrête en 6 second ;
- Le TIMER  $T1=10\text{s}$  activer ;

**Marche arrière du chariot :**

- Temps de 10s écouler, Le moteur démarre mais en sens contraire et atteint la fréquence requise  $F3=30\text{Hz}$  en 6 secondes ;
- Actionnez le bouton "C", le moteur décélère et atteint la fréquence  $F4=20\text{Hz}$
- Actionnez le bouton "Arrêt" ; Le moteur s'arrête au bout d'environ 3 secondes (Temps de lissage).

Le chronogramme suivant illustre le fonctionnement exigé dans le cahier de charge n°2 :

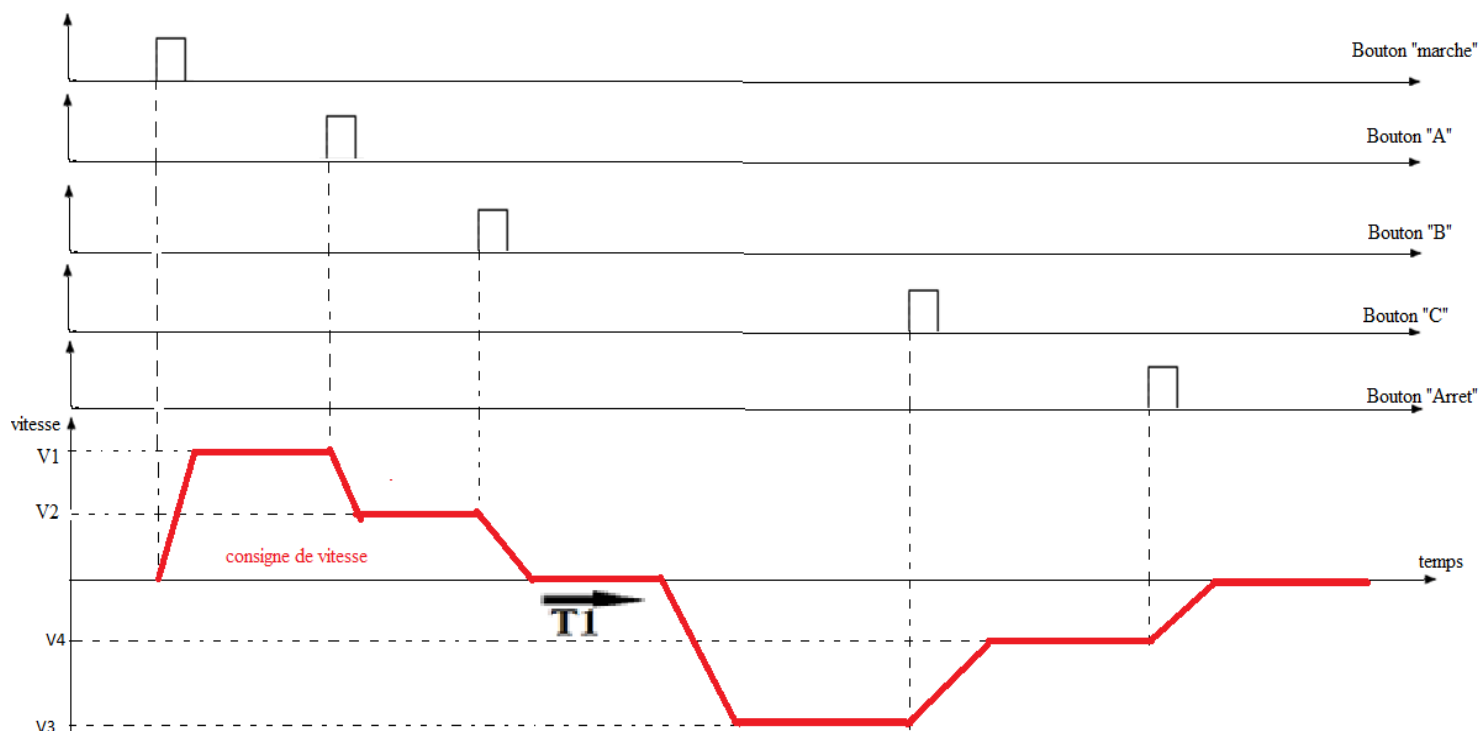


Figure 53: Chronogramme de fonctionnement.

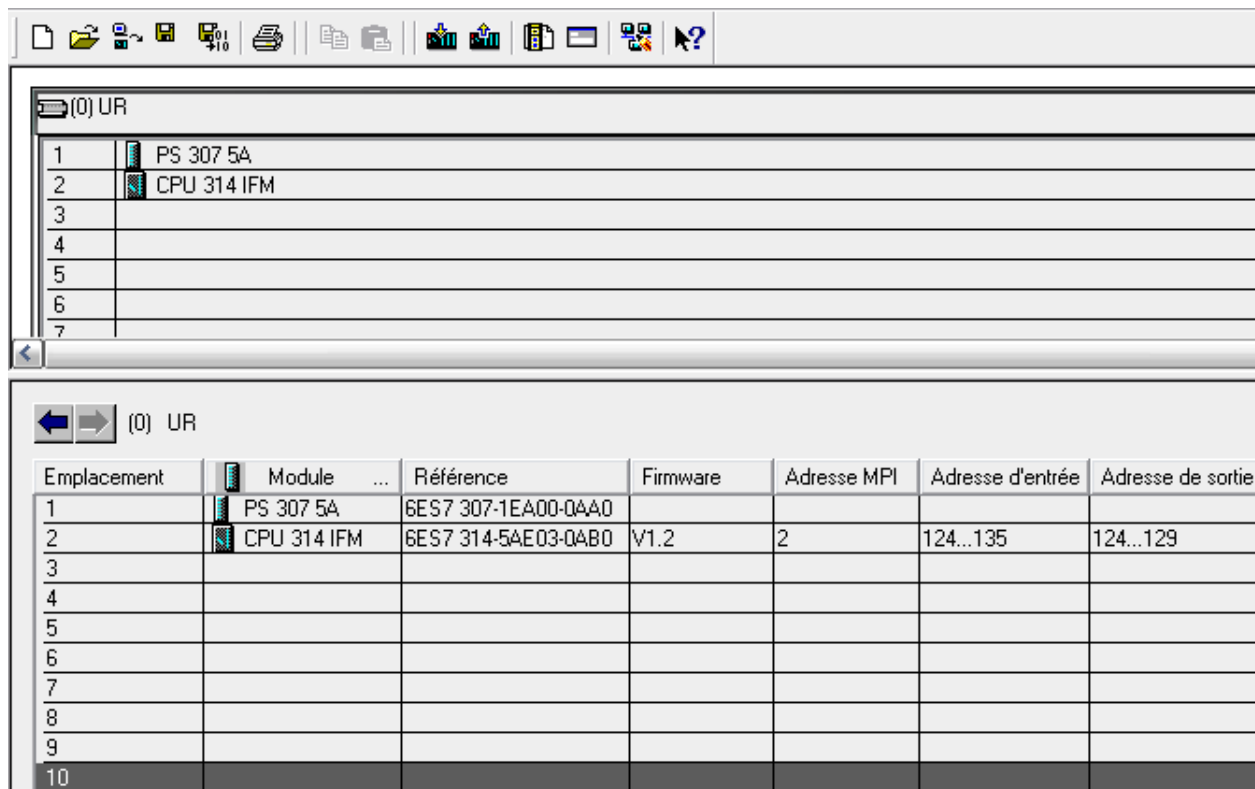
#### IV.4.3 Développement du projet S7 pour la maquette :

Création du projet dans STEP7

Dans cette partie, pour la création du projet dans STEP 7, on va suivre les mêmes principes de programmation que lors de la précédente application. On va suivre la méthodologie présentée auparavant dans l'organigramme du projet (Figure : l'organigramme de l'application).

### a. Configuration du matériel

Pour la configuration matérielle, on va suivre les mêmes instructions, et cela en utilisant le même automate. On peut déjà considérer que la configuration de la CPU et des modules est déjà faite.



**Figure 54: Configuration matériel pour la CPU314 IFM.**

### b. Conception et écriture du Programme

Dans cette partie, nous allons créer le programme à transférer dans la CPU. En premier lieu, on attribue des adresses absolues dans la CPU pour les entrées et sorties du processus, comme indiqué dans le tableau suivant :

Entrées /Sorties (Boutons/ voyants)	Adresse absolue
Marche	E 124.0
A	E 124.1
B	E 124.2
arrêt	E 124.3
reset	E 124.4
C	E 124.5

V01	A124.0
V02	A124.1
V03	A124.2
Vitesse de référence	PAW 128

**Tableau 6: Adressage des entrées /sorties du processus**

Avec pour chaque bloc une tâche spécifiée :

- Bloc d'organisation cyclique OB1 : c'est le bloc principal qui fait appel au :
  - Fonction de gestion des tâches FC1 : cette fonction a pour tâche de donner les ordres de Marche/Arrêt ainsi que les consignes vitesse pour le variateur et les 3 autres voyants conformément au cahier de charge n°1.
  - Fonction de diagnostic FC2 : cette fonction a pour tâche de donner les ordres de Marche/Arrêt ainsi que les consignes vitesse pour le variateur et les 3 autres voyants conformément au cahier de charge n°2.
  - Fonction de diagnostic FC3 : cette fonction a pour tâche de donner les ordres de Marche/Arrêt ainsi que les consignes vitesse pour le variateur et les 3 autres voyants conformément au cahier de charge n°3.
  - Fonction UNSCALE FC 106 : la fonction UNSCALE (Annuler la mise à l'échelle) prend une valeur d'entrée réelle (IN) exprimée en unités physiques comprises entre une limite inférieure (LO\_LIM) et une limite supérieure (HI\_LIM) et la convertit en une valeur entière.



**Figure 55: blocs programme dans SIMATIC Manager**

Pour cette application, on utilisera l'adressage relatif, qui consiste, à créer une table de mnémorique :

## CHAPITRE IV : La réalisation de l'armoire électrique

Editeur de mnémoniques - [Programme S7(1) (Mnémoniques)]

Table Edition Insertion Affichage Outils Fenêtre ?

Tous les mnémoniques

	Etat	Mnémonique	Opérande	Type de d	Commentaire
1		a	M 0.1	BOOL	
2		ae#	E 124.1	BOOL	
3		arret	M 0.3	BOOL	
4		arrett#	E 124.3	BOOL	
5		arriere#	A 124.3	BOOL	
6		avant#	A 124.4	BOOL	
7		b	M 0.2	BOOL	
8		b#	E 124.2	BOOL	
9		c	M 0.5	BOOL	
10		c#	E 124.5	BOOL	
11		cahier de charge 1	FC 1	FC 1	
12		cahier de charge 2	FC 2	FC 2	
13		cahier de charge 3	FC 3	FC 3	
14		cd1	M 0.6	BOOL	
15		cd1#	E 124.6	BOOL	
16		cd2	M 0.7	BOOL	
17		cd2#	E 124.7	BOOL	
18		cd3	M 2.0	BOOL	
19		cd3#	E 125.0	BOOL	
20		deplacement	MW 20	WORD	
21		deplacement 3	MW 24	WORD	
22		deplacement2	MW 22	WORD	
23		deplacement4	MW 26	WORD	
24		marche	M 0.0	BOOL	
25		marche#	E 124.0	BOOL	
26		reset	M 0.4	BOOL	
27		reset#	E 124.4	BOOL	
28		UNSCALE	FC 106	FC 106	Unscaling Values
29		v01#	A 124.0	BOOL	
30		v02#	A 124.1	BOOL	
31		v03#	A 124.2	BOOL	
32		visibi fleche1	A 1.3	BOOL	
33		visibil fleche2	A 1.4	BOOL	
34		visibilite	A 0.7	BOOL	
35		visibilite 2	A 1.1	BOOL	
36		visibilite railles	A 1.0	BOOL	
37		visibilite3	A 1.2	BOOL	
38		vite	A 0.5	BOOL	
39		vite2	A 0.6	BOOL	
40		vitesse	MD 4	REAL	
41		vitesse reel	PAW 128	INT	

40		vitesse	MD	4	REAL	
41		vitesse reel	PAW	128	INT	
42		x0	M	1.0	BOOL	
43		x1	M	1.1	BOOL	
44		x2	M	1.2	BOOL	
45		x3	M	1.3	BOOL	
46		x4	M	1.4	BOOL	
47		x5	M	1.5	BOOL	
48						

**Figure 56: Tableau de mnémonique**

Description bref des mnémoniques :

- Bloc d'organisation OB1, les blocs de fonction FC1, FC2, FC3 et FC106 ont déjà étaient décrit.
- X0, X1, X2, X3, X4 et X5 sont des mementos qui représentent les différentes étapes du grafcet.
- V01, V02, V03 sont des voyants.
- « Vitesse réel » PAW128 : c'est la consigne de vitesse transmis de l'automate s7 300 (via une sortie analogique) vers le variateur ATV312.
- Cd1, cd2, cd3 sont des mementos qui active et désactive les différents cahiers de charge.

Remarque : Tous les blocs de programme, seront écrits en langage « CONT ». Le code du programme dans ces blocs est donné en annexe.

Cette étape de « Création de projet dans STEP7 » va être conclue par la compilation de tous les blocs dans « SIMATIC Manager » et le chargement vers la CPU.

#### **IV.5 Présentation de l'installation :**

Le dispositif est constitué d'une table recevant l'ensemble des éléments composant l'installation.

##### **IV.5.1 Organisation électrique de l'armoire de commande.**

L'automatisme de commande est piloté par l'automate S7-300

On distingue essentiellement:

- L'alimentation triphasée 380V.
- L'API S7 300.
- Le variateur de fréquence ATV 312.

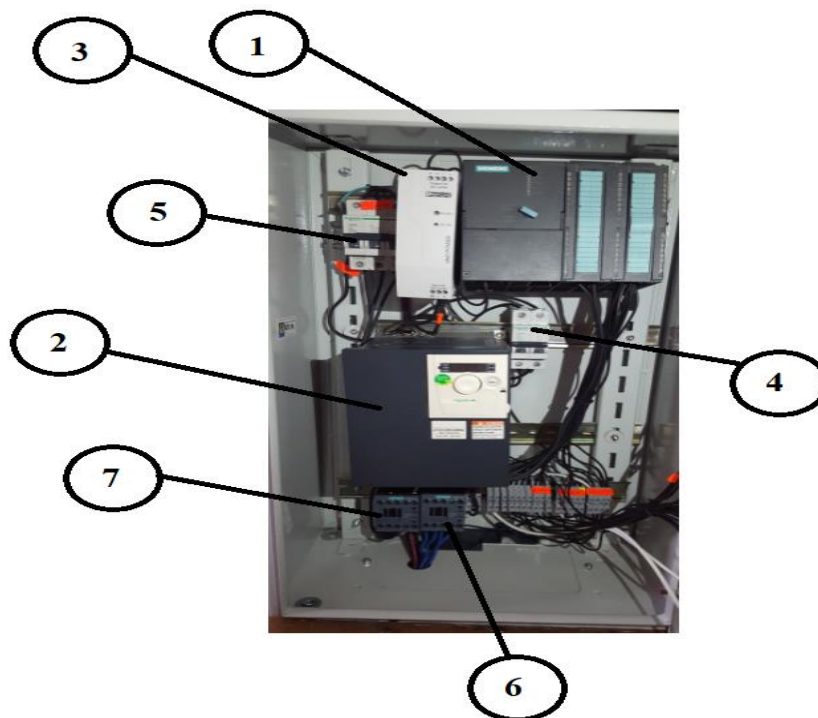


Figure 57:l'organisation électrique de l'armoire.

**Nomenclature de l'armoire de commande.**

Repère Schéma	Désignation	Fonction	Caractéristique
1	Automate programmable industriel	Commande et supervision du système	CPU 314 IFM
2	Variateur de fréquence	Pilotage moteur asynchrone triphasé	ATV312 (voir chapitre III)
3	Alimentation stabilisé	Alimentations circuit de commande	S=63VA Primaire 230V Secondaire 24V
4	Disjoncteur monophasé	Protection contre le court-circuit	
5	Disjoncteur triphasé	Protection contre le court-circuit	
6	Contacteur	Commande du bouton d'arrêt d'urgence	Bobine en 24VAC
7	Contacteur	Commande du moteur	Bobine en 24VAC

**Tableau 7 : la fonction principale des composants de l'armoire**



IV.5.2 Schéma de connexion des entrées et des sorties

La figure 58 représente le schéma de câblage

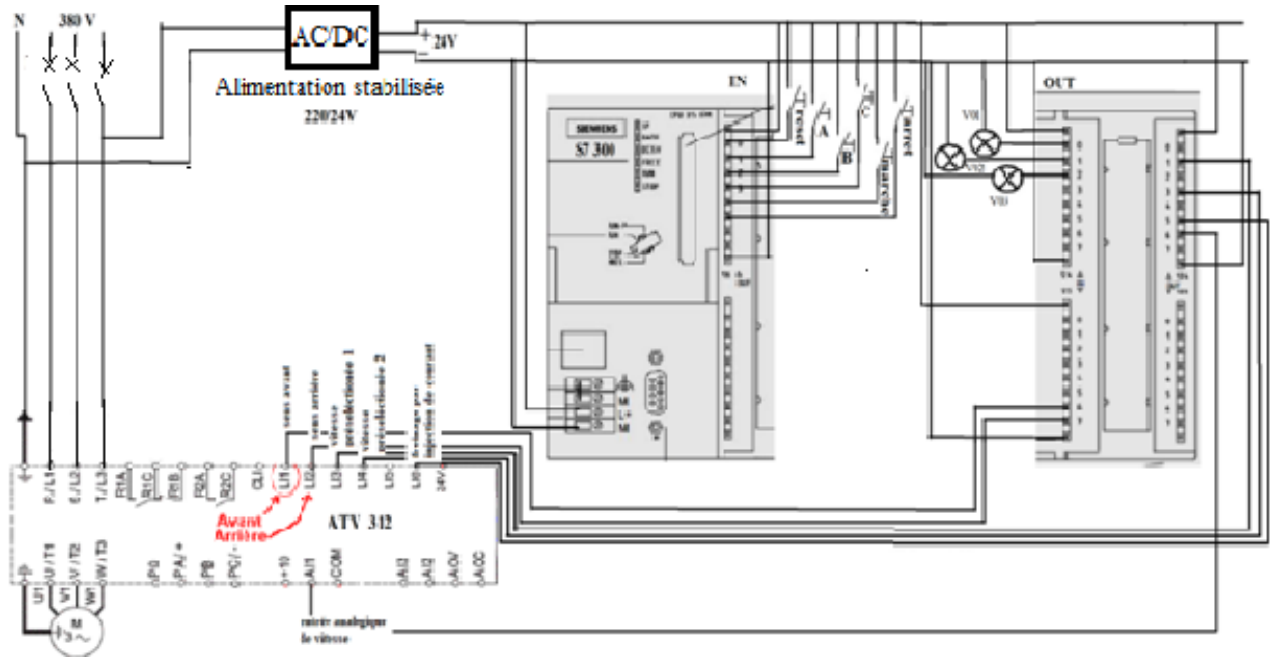


Figure 58 : Schéma de connexion des entrées/sorties.

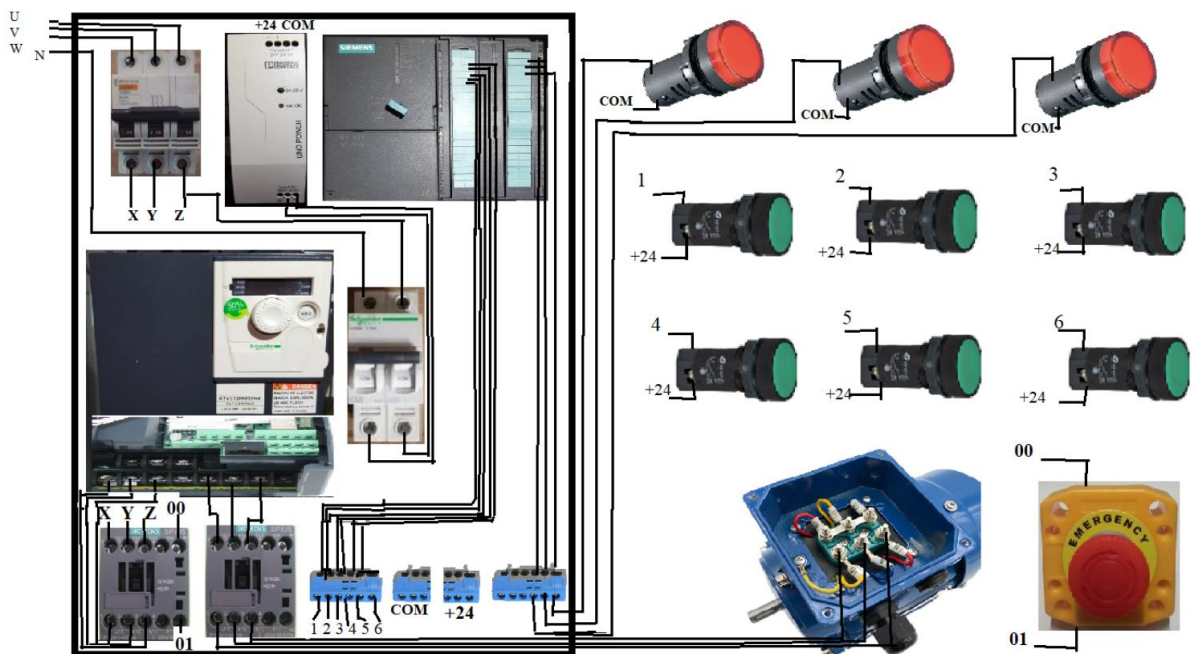


Figure 59 : Schéma de câblage et l'emplacement des composants dans l'armoire.

### IV.5.3 L'armoire de commande.



Elle regroupe les éléments de puissance, de sécurité et de gestion de l'automatisme « Entraînement chariot »

On trouve dans l'armoire :

- La paire Automate/Variateur qui constitue les deux éléments principaux pour la gestion et l'entraînement de cette installation.

- Une alimentation S307 5A alimente en monophasé 220 V et qui délivre du 24V.

Système PC comme Panneau de supervision est mis à disposition configure selon le besoin et programmée avec le logiciel HMI

On trouve dans l'armoire :

### IV.5.4 Les éléments de manœuvre et de signalisation.

La manœuvre du déplacement de ce chariot est commandée manuellement depuis les boutons placés sur la porte de l'armoire électrique.

Ou bien depuis l'interface homme machine HMI qui est dans notre cas l'écran du PC.



Figure 60 : les éléments de manœuvre et de signalisation.

#### IV.6 L'interface homme machine HMI configurée par le logiciel « Win CC flexible »

L'interface homme machine, WINCC Flexible Runtime ou il est possible de visualiser l'état du processus et du moteur ainsi que le chariot et le sens de rotation du moteur ...etc.

##### IV.6.1 Création de la station HMI :

Dans notre projet "entraînement d'un chariot" nous avons introduit la station HMI en choisissant le pupitre WINCC FLEXIBLE RUNTIME sur lequel les informations seront transmises, pour notre application nous avons utilisé l'écran de l'ordinateur en liaison avec l'automate programmable à l'aide d'un câble MPI.

Pour la gestion de notre station on crée les vues suivantes :

- Une vue principale nommée "page d'accueil".



Figure 61 : vue principale créée pour la HMI

- Si on clique sur la flèche, la vue qui contient les trois cahiers de charge apparaît

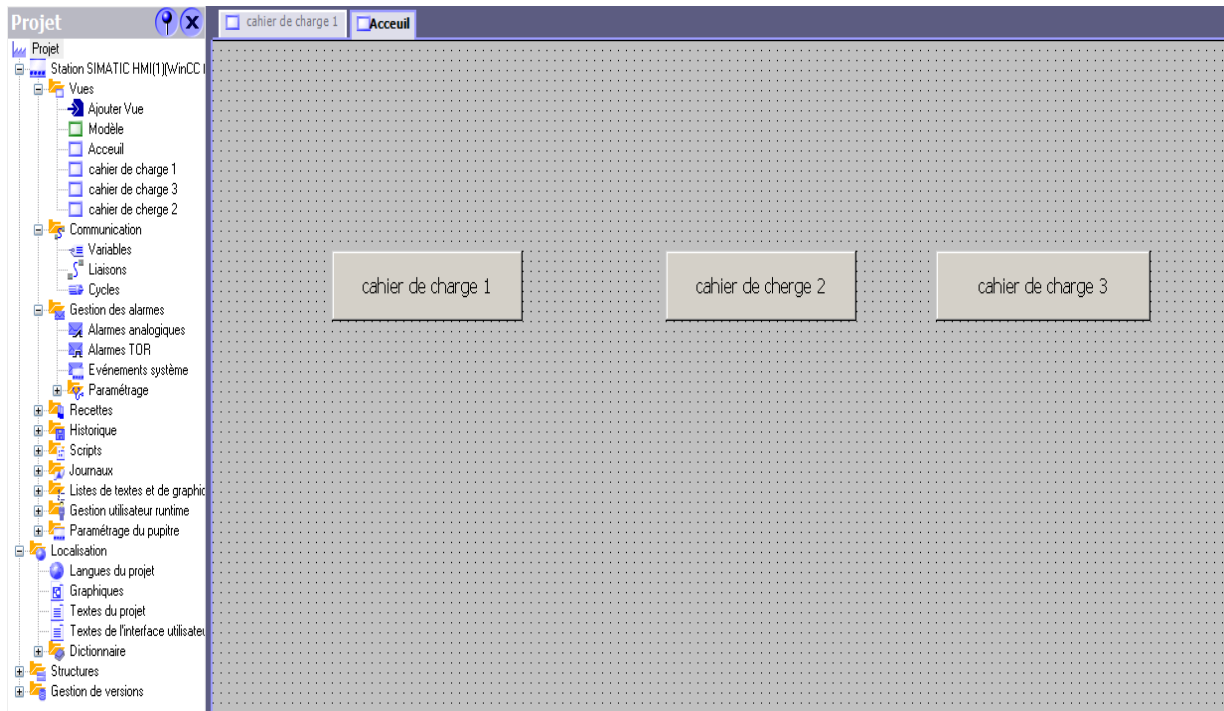


Figure 62 : La vue des trois cahiers de charge.

- Les autres vues nommées “chier de charge 1“, “chier de charge 2“ et “chier de charge 3“ sert à visualiser les différents états du chariot ainsi que le moteur et son sens de rotation et les états des voyants.

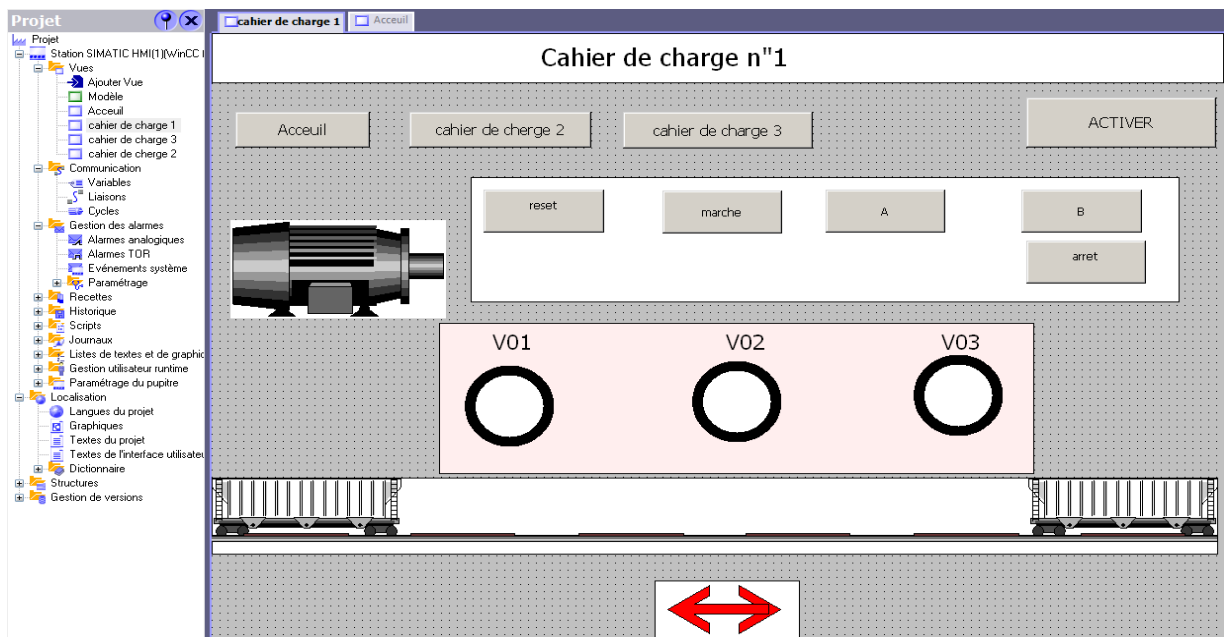


Figure 63: Vue cahier de charge n°1

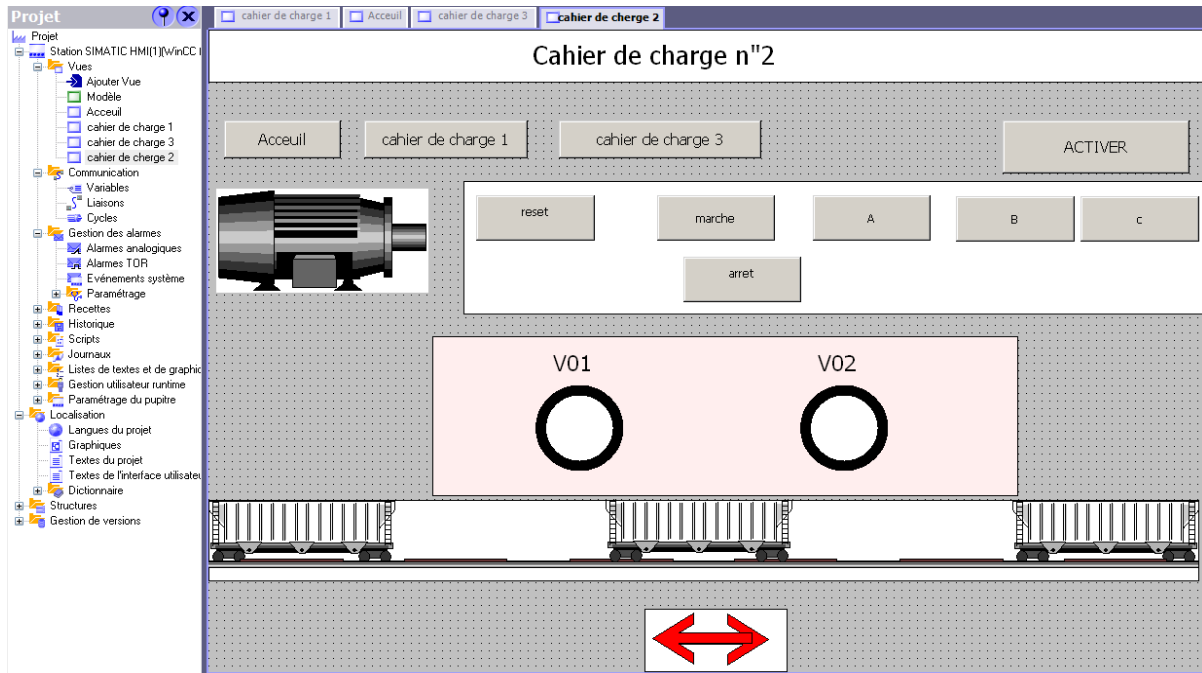


Figure 64 : Vue cahier de charge n°2

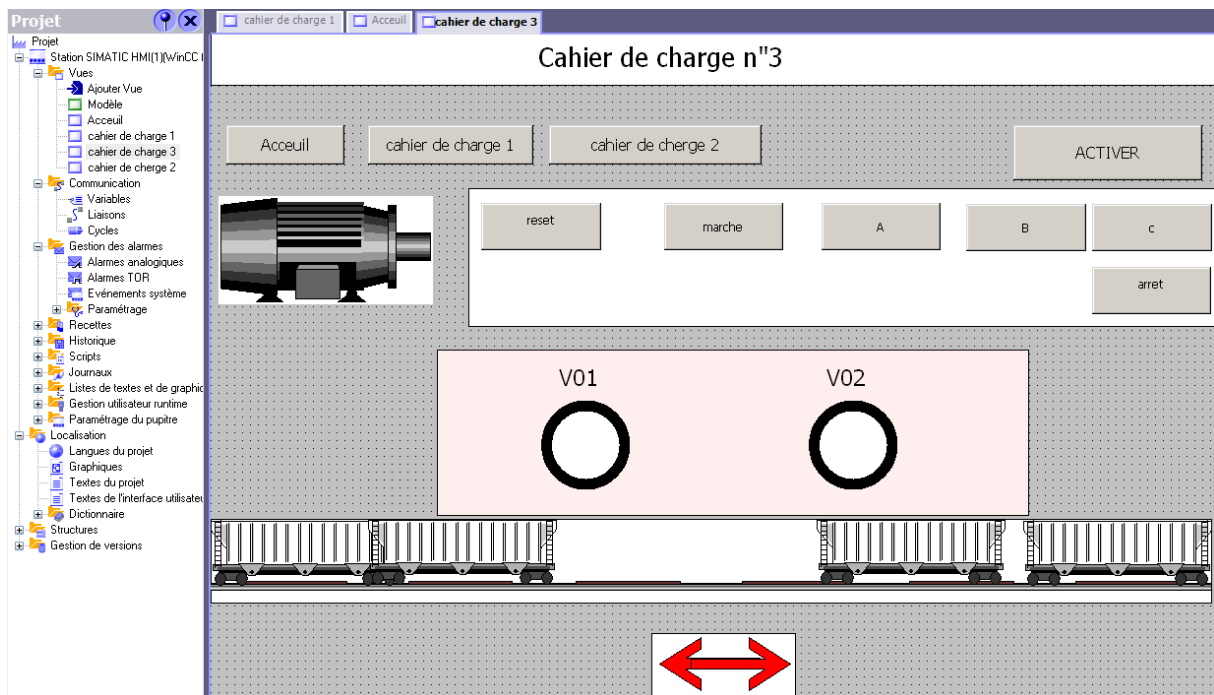


Figure 65: Vue cahier de charge n°3

#### IV.6.2 Paramétrage du pupitre :

Dans cette étape on doit :

- Nommer notre station HMI ;
- Choisir le type de pupitre ;
- Choisir la vue initiale.

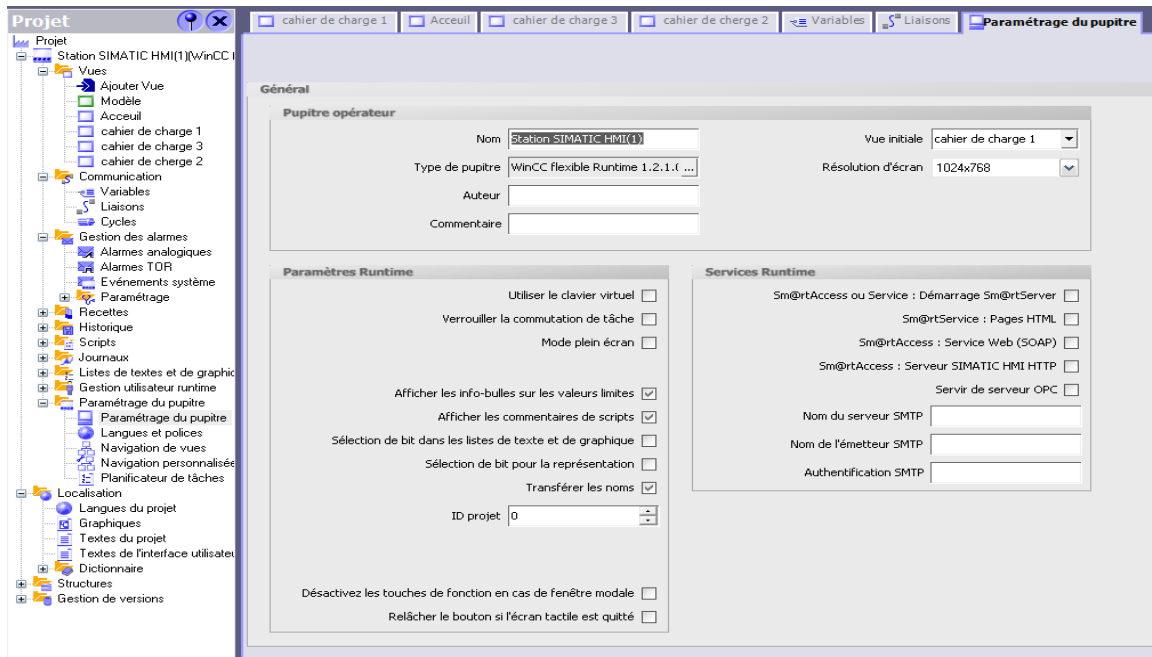


Figure 66 : paramétrage du pupitre.

## IV.6.3 L'édition des variables :

Pour établir une liaison entre le logiciel step7 et le WinCC, il faut éditer les variable dans le WinCC et donner a chaque variable l'adresse éditer déjà dans le step7

Nom	Liaison	Type de données	Mnémonique	Adresse	Éléments du ta...	Cycle d'acq...	Archive de don...	Mode d'acquisition archivage	Cycle d'archivage
a	CPU 314IFM	Bool	<indéfini>	M 0.1	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
arrêt	CPU 314IFM	Bool	<indéfini>	M 0.3	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
b	CPU 314IFM	Bool	<indéfini>	M 0.2	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
c	CPU 314IFM	Bool	<indéfini>	M 0.5	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
cd1e	CPU 314IFM	Bool	<indéfini>	M 0.6	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
cd2e	CPU 314IFM	Bool	<indéfini>	M 0.7	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
cd3e	CPU 314IFM	Bool	<indéfini>	M 2.0	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
d	CPU 314IFM	Bool	<indéfini>	M 1.7	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
deplacement	CPU 314IFM	Word	<indéfini>	MW 20	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
deplacement2	CPU 314IFM	Word	<indéfini>	MW 22	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
deplacement3	CPU 314IFM	Word	<indéfini>	MW 24	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
deplacement4	CPU 314IFM	Word	<indéfini>	MW 26	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
deplacement5	CPU 314IFM	Word	<indéfini>	MW 28	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
droite	CPU 314IFM	Bool	<indéfini>	Q 0.4	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
gauche	CPU 314IFM	Bool	<indéfini>	Q 0.3	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
marche	CPU 314IFM	Bool	<indéfini>	M 0.0	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
reset	CPU 314IFM	Bool	<indéfini>	M 0.4	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
v01	CPU 314IFM	Bool	<indéfini>	Q 0.0	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
v02	CPU 314IFM	Bool	<indéfini>	Q 0.1	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
v03	CPU 314IFM	Bool	<indéfini>	Q 0.2	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
v3	CPU 314IFM	DWord	<indéfini>	QD 8	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
visibi fleche1	CPU 314IFM	Bool	<indéfini>	Q 1.3	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
visibilite	CPU 314IFM	Bool	<indéfini>	Q 0.7	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
visibilite 2	CPU 314IFM	Bool	<indéfini>	Q 1.1	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
visibilite ralles	CPU 314IFM	Bool	<indéfini>	Q 1.0	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
visibilite3	CPU 314IFM	Bool	<indéfini>	Q 1.2	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
visibili fleche2	CPU 314IFM	Bool	<indéfini>	Q 1.4	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
vite	CPU 314IFM	Bool	<indéfini>	Q 0.5	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
vite2	CPU 314IFM	Bool	<indéfini>	Q 0.6	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
vitesse	CPU 314IFM	DWord	<indéfini>	QD 4	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
x0	CPU 314IFM	Bool	<indéfini>	M 1.0	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>
v1	CPU 314IFM	Bool	<indéfini>	M 1.1	1	100 ms	<indéfini>	Cyclique en continu	<indéfini>

Variable	Type	Adresse	Commentaire	Unité	Initialisation	Mode	Temps	Comportement	Commentaire
vite	Bool	Q 0.5	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
vite.2	Bool	Q 0.6	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
vitesse	DWord	QD 4	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x0	Bool	M 1.0	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x1	Bool	M 1.1	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x2	Bool	M 1.2	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x3	Bool	M 1.3	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x4	Bool	M 1.4	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x5	Bool	M 1.5	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>
x6	Bool	M 1.6	<indéfini>	1	100 ms	<indéfini>		Cycle en continu	<indéfini>

Figure 67 : l'édition des variables

### IV.6.4 Configuration de la connexion entre l'API et le WinCC Advanced RT :

Pour établir la connexion entre l'API et WinCC, on utilise une connexion de type MPI a l'aide d'une adaptateur MPI/USB.

Il faut ouvrir l'éditeur de connexion en appuyant sur "liaison".

Dans l'éditeur, on ajoute une connexion de type MPI en prenant la configuration suivante.

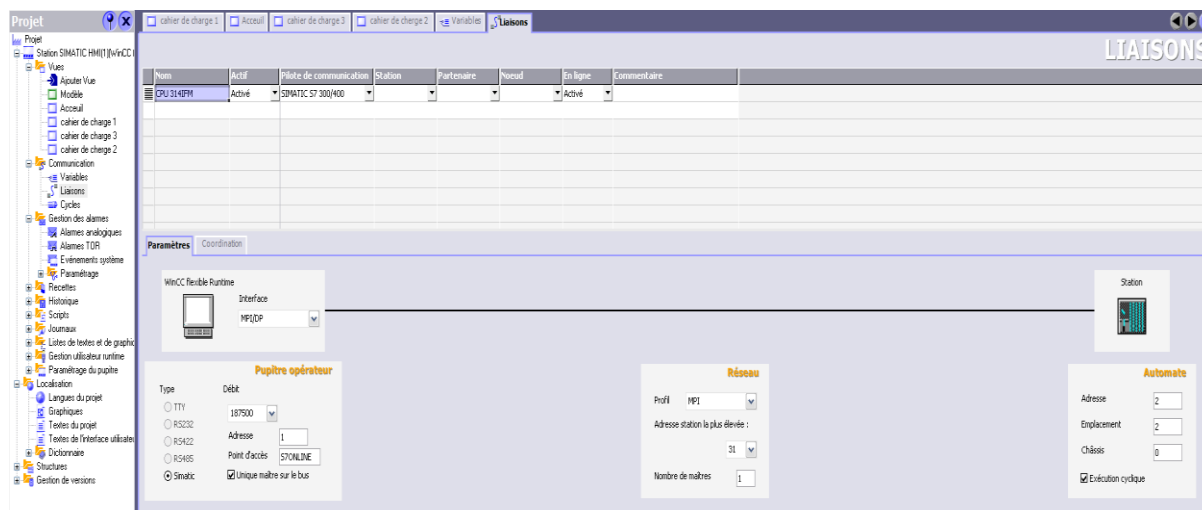


Figure 68 : Configuration de la liaison MPI.

### IV.6.5 Simulation du programme STEP 7 dans PLCSIM :

Pour tester le programme, on doit démarrer la simulation dans PLCSIM. Cette simulation nous permet de tester le bon fonctionnement du programme avant de démarrer la production. Nous chargeons d'abord la configuration et le programme utilisateur dans le module simulé et nous insérons des sous-fenêtres pour visualiser et forcer les sorties et mémentos. Insérez une sous-fenêtre "Memento" pour forcer et afficher les variables API avec bits de memento dans la plage d'adresse voulu.

### IV.6.6 Chargement dans le système cible à partir de la PG/PC

Le chargement du programme dans le système cible se fait sous certaines conditions :

- Une liaison est établie entre la PG et la CPU du système cible (via l'interface MPI ou PROFIBUS).
- L'accès au système cible est possible.

- La compilation du programme à charger doit se faire sans erreur.
- La CPU doit se trouver dans un état de fonctionnement autorisant le chargement (STOP ou RUN-P).

Si la syntaxe est correcte, le bloc est ensuite compilé en code machine, enregistré et chargé par la Commande Système cible > Charge.

#### IV.7 Les tests en ligne :

- Après le chargement du programme dans le système cible il suffit juste de mettre la CPU en mode RUN.
- On lance le système.

On voit que la vue principale apparaît, après l'apparition de cette vue il faut choisir l'un des cahiers de charge.

- On choisit le cahier de charge 1.
- En appuyant sur le bouton reset le voyant vo1 s'allume (état 0).

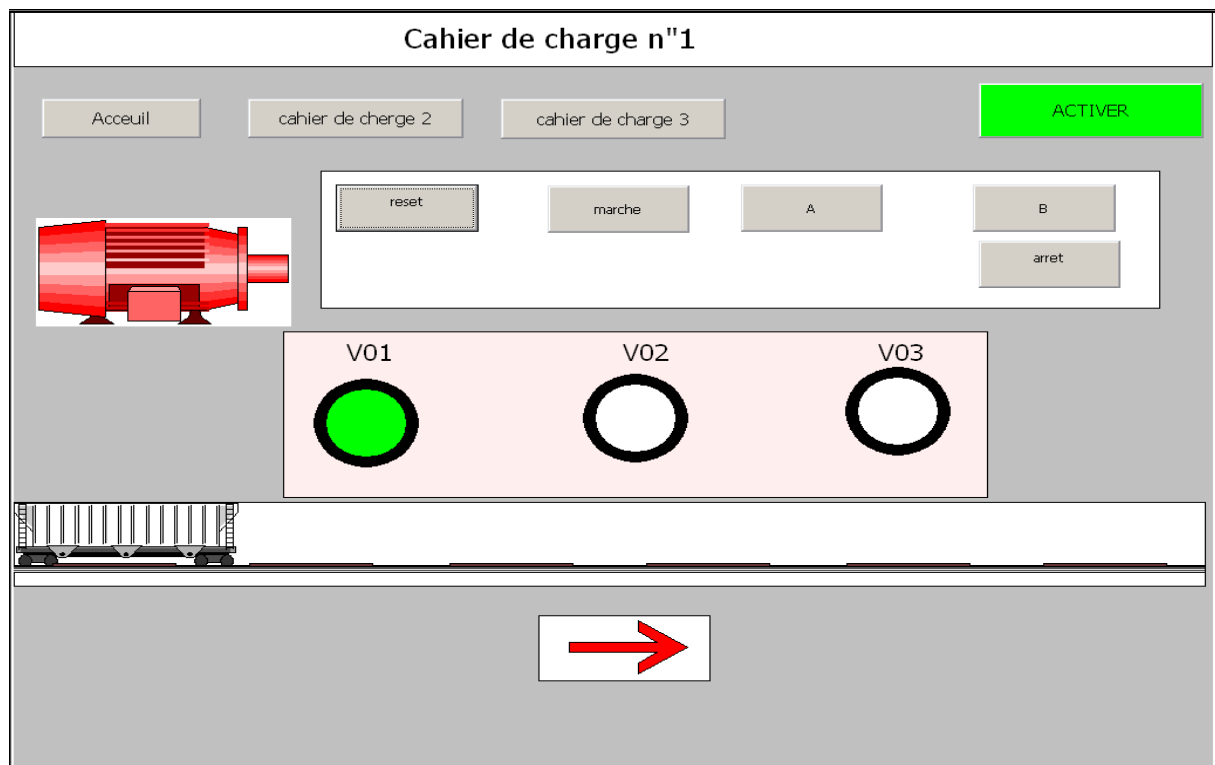
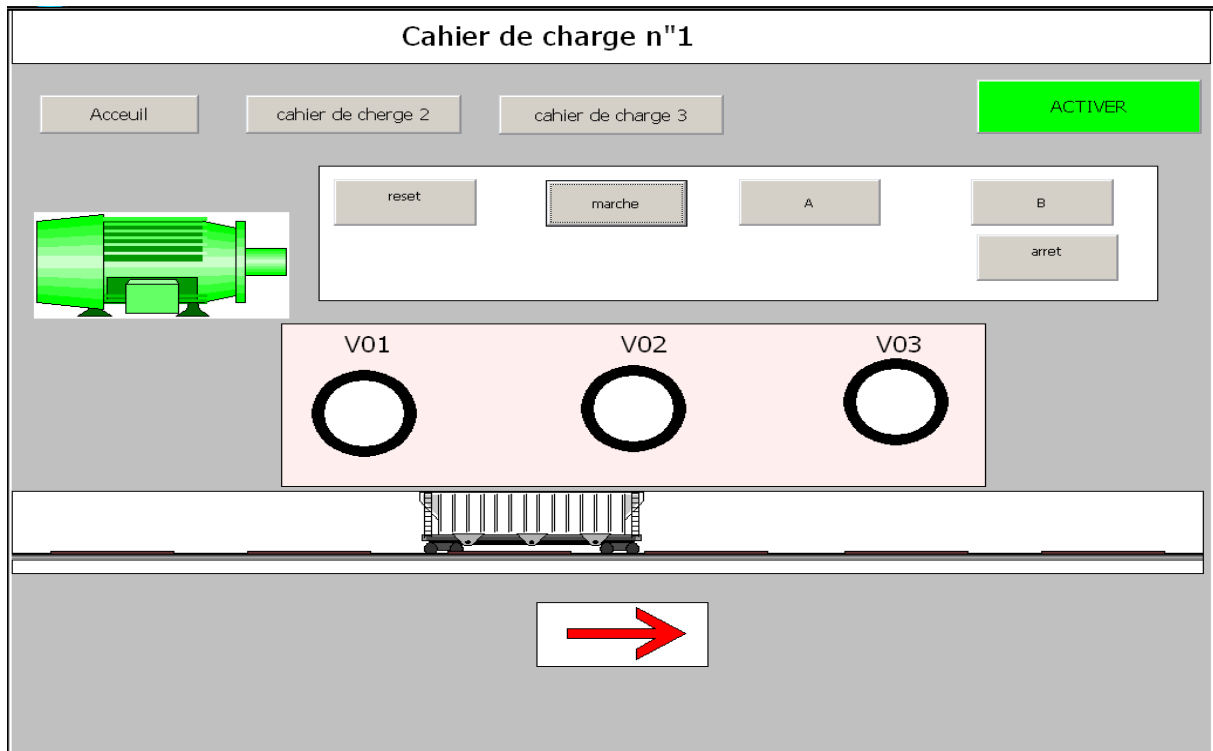


Figure 69 : visualisation de l'état zéro du premier cahier de charge

- En appuyant sur le bouton marche le moteur tourne et le chariot avance.





**Figure 70 : visualisation de l'état 1 du premier cahier de charge.**

En appuyant sur le bouton A, le moteur s'arrête pendant 5s et le voyant vo2 s'allume.

**Conclusion :**

Nous avons pu réaliser une armoire électrique à base de variateur de vitesse ATV 312 commandé par un automate programmable S7-300. On a programmé le fonctionnement par le logiciel step7 et la supervision à travers l'interface graphique WinCC flexible.

# Conclusion générale

Un élève ingénieur ne doit pas être satisfait de l'acquisition de connaissances théoriques, il est plutôt essentiel qu'il complète sa formation par une bonne expérience dans le domaine de la pratique, c'est à cet effet que nous avons choisi un projet de fin d'étude qui se concentre sur la réalisation d'une armoire électrique à base de variateur de vitesse commandé par un automate programmable.

Pour la commande du système, nous avons utilisé un automate programmable SIEMENS de série S7-300 de CPU 314-IFM et pour la supervision nous avons utilisé l'écran du pc comme une interface homme machine HMI connectée à l'automate programmable via un câble MPI.

La programmation de l'automate programmable a été faite par le logiciel STEP7 et pour l'écran de supervision nous avons utilisé le WINCC flexible.

Pendant, la réalisation nous avons acquis de nouvelles compétences et connaissances à propos des deux logiciels utilisés et l'utilisation des automates programmables ainsi que les variateurs de vitesse.

Nous avons eu des difficultés au début puisque ce fut la première fois que nous manipulons les automates programmables de Siemens mais avec le temps s'est adapté.

On a appris de nouvelles compétences telles que : le travail en groupe, la planification, la gestion du budget, ainsi que des principes de bases dans le câblage électrique et l'instrumentation.

Enfin, nous espérons que notre travail sera utile aux étudiants, pour appliquer les connaissances requises en cours d'automate programmable.

## **Perspective :**

La maquette qu'on a réalisée se base totalement sur les entrées sorties TOR et une sortie analogique pour commander la vitesse du moteur. Il serait très intéressant de faire une régulation de vitesse en ajoutant un :

- Capteur de vitesse.

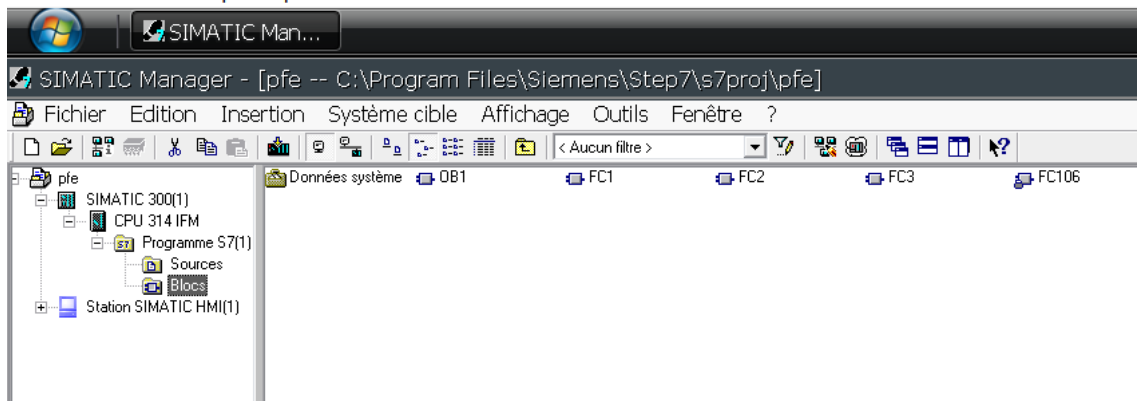
# Bibliographie :

- [1] M. BERTRAND, «Automates programmables industriels,» Technique de l'ingénieur, vol. S8015.
- [2] G. MICHEL, « Les A.P.I Architecture et application des automates programmables industriels», Edition DUNOD, 1987
- [3] William Bolton «automates programmables industriels 2e édition », Edition DUNOD, 2015
- [4] Les gammes d'automate Siemens, [en ligne]. [Consulté le 28/03/2019].  
Disponible sur :  
<<https://www.automation-sense.com/blog/automatisme/les-gammes-d-automates-siemens.html>>
- [5] SIEMENS-SIMATIC-Programmer avec STEP 7-Manuel-référencée : 6ES7810-4CA08-8CW0
- [6] SIEMENS-SIMATICSTEP 7 -Global library "Sample Library for Instructions" – Manual
- [7] SIEMENS-SIMATIC-Configuration matérielle et communication dans STEP 7- Manuelréférencée: 6ES7810-4CA08-8CW0.
- [8] SIEMENS-SIMATIC HMI-WinCC flexible 2008-Compact / Standard / AdvancedManuel référencé 6AV6691-1AB01-3AC0.
- [9] Forma TIS (Centre de formation Technique. Industrielle. Sécurité), Altivar 312 [en ligne]. [Consulté le 23/05/2019]. Disponible sur : <<http://blog.formatis.pro/altivar-312>>
- [10] Altivar 312 <<Programming Manual Directives de programmation >>.

- [11] Schanen, Thierry. Guide des automatismes Version 7 [en ligne]. [Consulté le 23/05/2019] disponible sur :  
<<https://www.cours-electromecanique/2014/12/telecharger-gratuitement-guide-des.html>>
- [12] Altivar 312.<<Variateurs de vitesse pour moteurs asynchrones /Guide de programmation. BBV46384>>
- [13] Forma TIS (Centre de formation Technique. Industrielle. Sécurité), Altivar 312 [en ligne]. [Consulté le 23/05/2019]. Disponible sur :  
<<http://blog.formatis.pro/altivar-312>>

# ANNEXE A : Le programme de l'application « mise en œuvre d'un chariot »

»



CONT/LIST/LOG - [OB1 -- pfe\SIMATIC 300(1)\CPU 314 IFM]

Fichier Edition Insertion Système cible Test Affichage Outils Fenêtre ?

Contenu de : 'Environnement\Interface'

Nom
TEMP

Nouveau réseau

- Opérations sur bits
- Comparaison
- Conversion
- Comptage
- Appels de DB
- Sauts
- Nombres entiers
- Nombres réels
- Transfert
- Gestion de programme
- Décalage/rotation
- Bits du mot d'état
- Temporisations
- Opérations sur mots
- Blocs FB
- Blocs FC
- Blocs SFB
- Blocs SFC
- Multi-instances
- Bibliothèques

Interface

- TEMP

Réseau 1 :

M3.6 "cd1" — EN — FC1 "cahier de charge 1" — ENQ

M124.6 "cd1#" — EN — FC1

Réseau 2 : Titre :

Commentaire :

Réseau 2 :

M124.7 "cd2#" — EN — FC2 "cahier de charge 2" — ENQ

M3.7 "cd2" — EN — FC2

Réseau 3 : Titre :

Commentaire :

Réseau 3 :

M125.0 "cd3#" — EN — FC3 "cahier de charge 3" — ENQ

M3.0 "cd3" — EN — FC3

CONT/LIST/LOG - [FC1 -- "cahier de charge 1" -- pfe\SIMATIC 300(1)\CPU 314 IFM\...\FC1]

Fichier Edition Insertion Système cible Test Affichage Outils Fenêtre ?

Contenu de : 'Environnement\Interface'

Nom	Nom
IN	IN
OUT	OUT
IN OUT	IN OUT

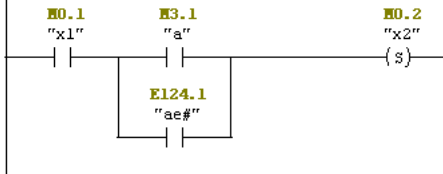
FC1 : Titre :  
Commentaire :

Réseau 1: Titre :  
Commentaire :

Réseau 2: Titre :  
Commentaire :

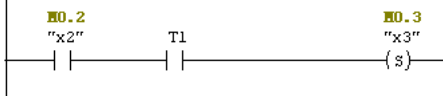
Eléments de p... Structure d'a...

- ☞ Conversion
- ☞ Comptage
- ☞ Appels de DB
- ☞ Sauts
- ☞ Nombres entiers
- ☞ Nombres réels
- ☞ Transfert
- ☞ Gestion de programme
- ☞ Décalage/rotation
- ☞ Bits du mot d'état
- ☞ Temporisations
- ☞ Opérations sur mots
- ☞ Blocs FB
- ☞ Blocs FC
- ☞ Blocs SFB
- ☞ Blocs SFC
- ☞ Multi-instances
- ☞ Bibliothèques



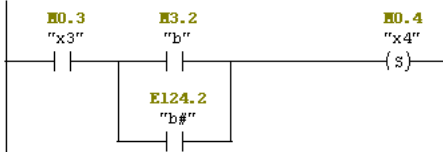
Réseau 4 : Titre :

Commentaire :



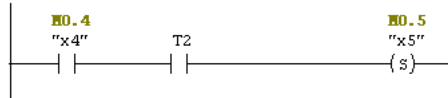
Réseau 5 : Titre :

Commentaire :



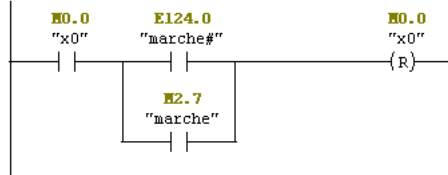
Réseau 6 : Titre :

Commentaire :



Réseau 7 : Titre :

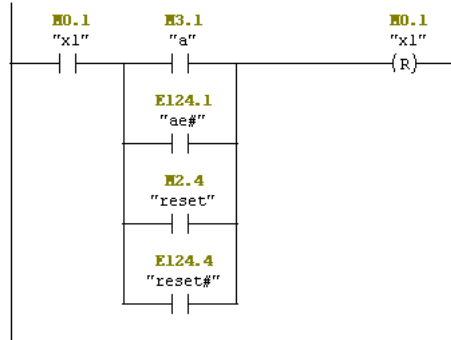
Commentaire :





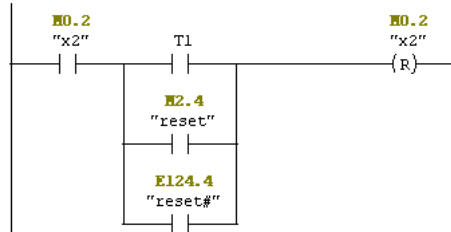
Réseau 8 : Titre :

Commentaire :



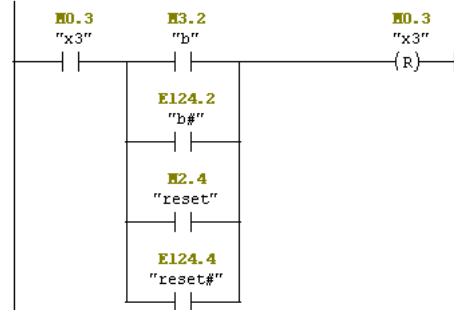
Réseau 9 : Titre :

Commentaire :



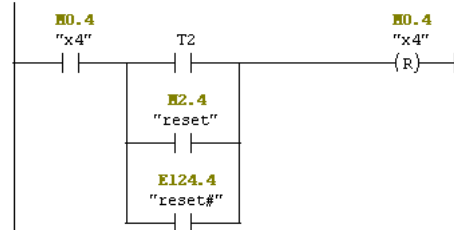
Réseau 10 : Titre :

Commentaire :



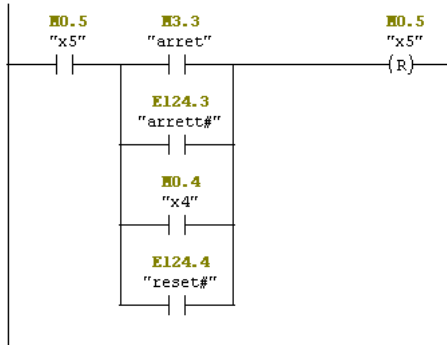
Réseau 11 : Titre :

Commentaire :



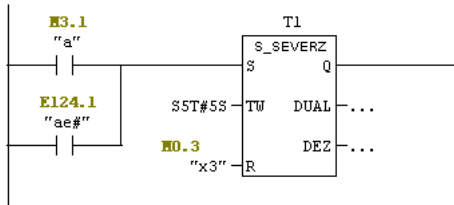
Réseau 12 : Titre :

Commentaire :

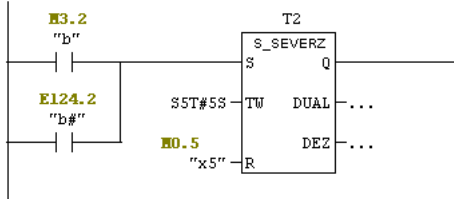


Réseau 13 : Titre :

Commentaire :

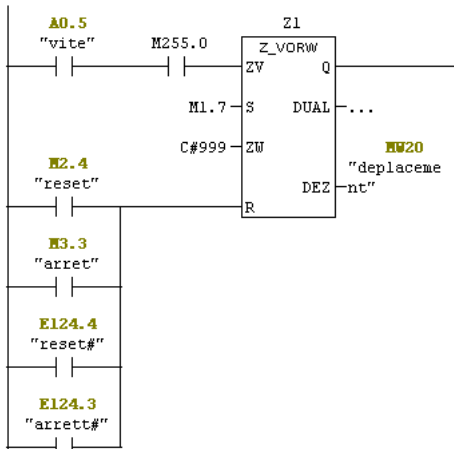


Commentaire :



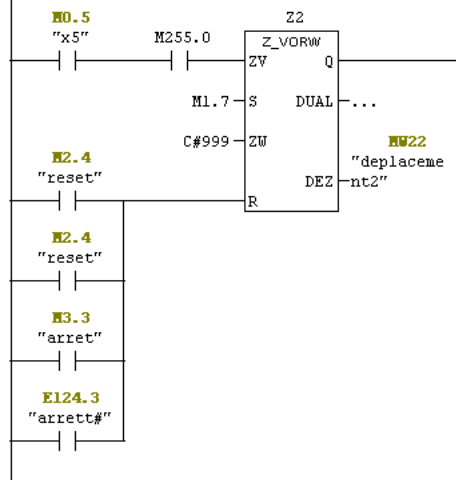
Réseau 15 : Titre :

Commentaire :



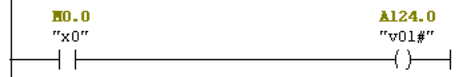
Réseau 16 : Titre :

Commentaire :



Réseau 17 : Titre :

Commentaire :



Réseau 18 : Titre :

Commentaire :



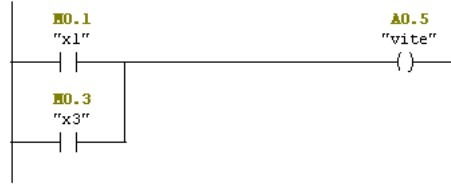
Réseau 19 : Titre :

Commentaire :



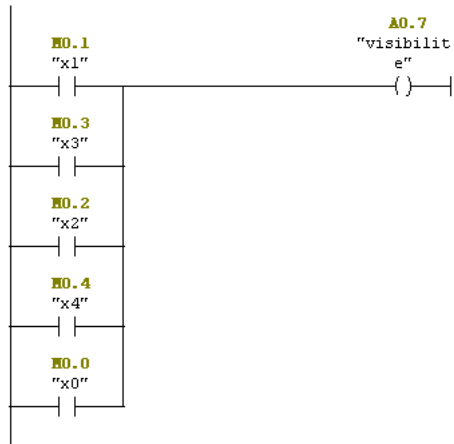
Réseau 20 : Titre :

Commentaire :



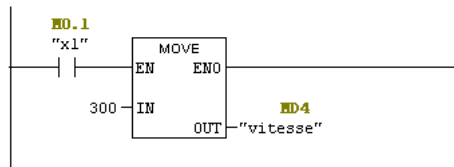
Réseau 21: Titre :

Commentaire :



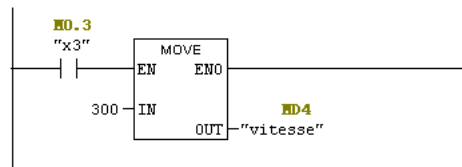
Réseau 22: Titre :

Commentaire :



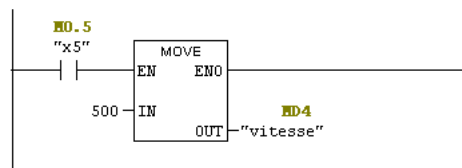
Réseau 23: Titre :

Commentaire :



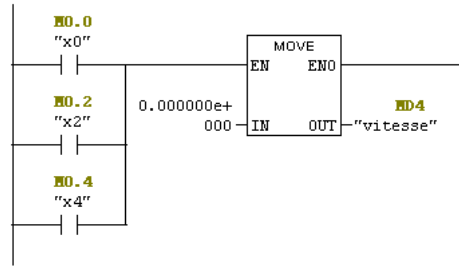
Réseau 24: Titre :

Commentaire :



Réseau 25: Titre :

Commentaire :



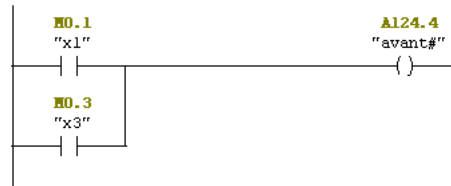
Réseau 26: Titre :

Commentaire :



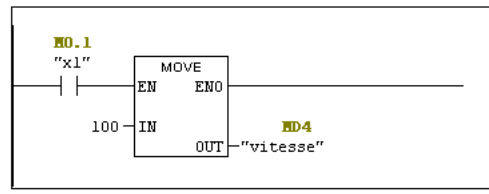
Réseau 27: Titre :

Commentaire :



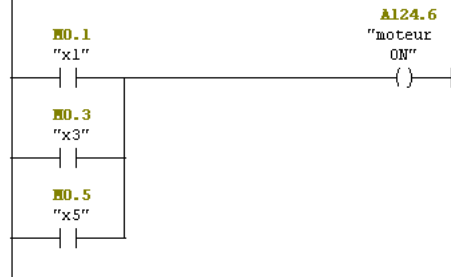
Réseau 28: Titre :

Commentaire :



Réseau 29 : Titre :

Commentaire :



Réseau 30 : Titre :

Commentaire :



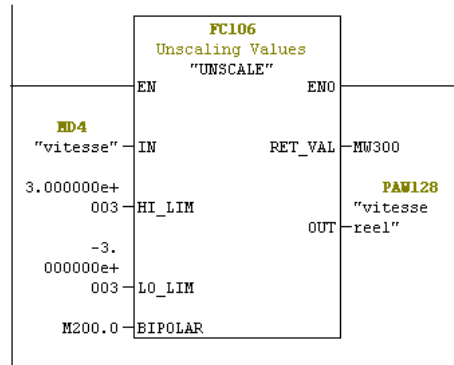
Réseau 31 : Titre :

Commentaire :



Réseau 32 : Titre :

Commentaire :



# Annexe B : Description des éléments utilisés.

## 1. Le moteur asynchrone.

Étant donné que nous utilisons ce type de moteur dans notre système d'irrigation, nous allons parler de façon plus détaillée :

De par sa robustesse, la simplicité de sa structure, son poids et sa taille, la machine asynchrone offre de nouvelles perspectives technologiques dans de nombreux domaines industriels, où l'on apprécie principalement son peu de maintenance et son faible coût.

Sa puissance s'étend d'une centaine de watts à 20 MW. Les vitesses de rotation sur le réseau triphasé 50 Hz sont de 3000, 1500, 1000, 750 tr/mn pour des moteurs dont le nombre de paires de pôles est respectivement 1, 2, 3, 4

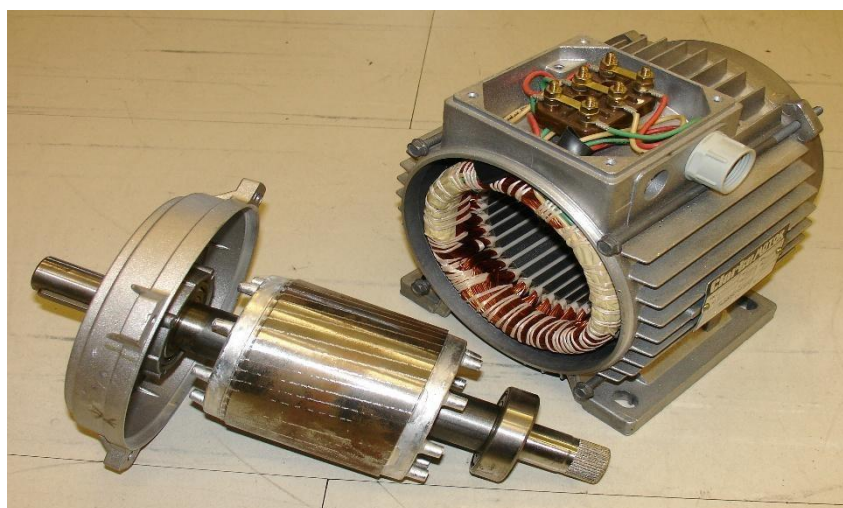


Figure : moteur asynchrone

## 2. Alimentation stabilisée.

### Définition :

Une alimentation stabilisée est un dispositif électronique, utilisant une régulation, permettant le réglage de la tension et du courant nécessaires au fonctionnement d'un appareil électrique ou électronique. Il est généralement conçu pour fonctionner en générateur idéal de tension et de courant.

### Description :

Exemple d'une alimentation stabilisée

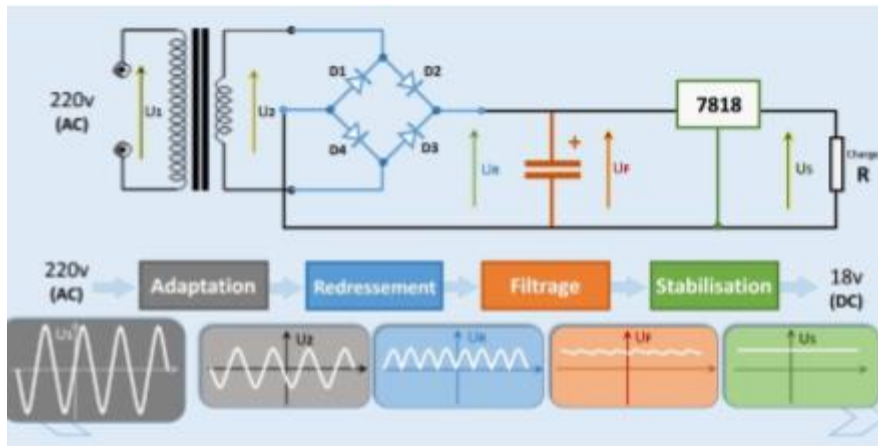


Figure : Schéma électrique d'une alimentation stabilisée.

### Caractéristique :

Tension et courant d'entrée :                      Tension et courant de sortie :  
 200-240VAC 2.0A 50/60Hz                      24V DC 5A

### 3. Contacteur :

#### Définition :

Un contacteur est un appareil électrotechnique destiné à établir ou interrompre le passage du courant, à partir d'une commande électrique ou pneumatique.

#### Le contacteur est constitué de :

- Une Bobine.
- Un ressort de rappel.
- De 2 à 4 contacts de puissance ou pôles (unipolaires, bipolaires, tripolaires Tétra polaires).
- Un circuit magnétique constitué d'un aimant fixe et d'un aimant mobile (armature fixe et mobile).
- Une bague de déphasage qui stabilise les vibrations des bobines alimentées en courant alternatif.
- Des contacts auxiliaires ouverts ou fermés.





FigureIII.10 Contacteur.

**Principe de fonctionnement :**

La bobine du contacteur (bornes A1-A2), peut-être alimentée en courant alternatif ou en courant continu (24V,48V,110V,230V,400 V).

Lorsque la bobine est alimentée, un champ magnétique se forme, la partie mobile de l'armature est attirée contre la partie fixe et les contacts se ferment (ou s'ouvrent suivant le modèle).

Lorsque la bobine n'est pas alimentée, le ressort de rappel sépare les deux parties de l'armature et maintient les contacts de puissance ouverts (ou fermés).

# ANNEXE C : Support TP.

## **Objectif du TP :**

L'objectif du TP est de former les étudiants sur le logiciel step7 et le logiciel de supervision WINCC flexible, ainsi que la maîtrise du variateur de vitesse ATV312.

## **Enoncé du problème :**

On désire implémenter une structure d'automatisation complète pour commander un chariot selon un cahier des charges imposé.

Et pour y faire l'étudiant va transformer ce cahier des charges en grafset qui sera programmé sur la machine virtuelle.

Après la vérification du fonctionnement de son programme sur Step7, il peut voir l'application directe de son cahier de charge en réalité.

- La commande du chariot est conformément aux modes de fonctionnement décrits ci-dessous :

En appuyant de manière brève sur le bouton "MARCHE " le chariot COMMENCE à se déplacer en avant avec une vitesse  $V1$  et en appuyant sur le bouton "A " le chariot va se déplacer avec une vitesse  $v2$ , et si on actionne le bouton "B " le moteur commence à décélérer jusqu'à ce qu'il atteigne la vitesse  $V3$ , et dès que le bouton "C " sera appuyé le chariot s'arrête et un TIMER " T1" sera lancer.

Après l'écoulement des 10s programmé du TIMER "T1", le chariot se déplace en arrière avec une vitesse "  $V4$  ". En appuyant sur la touche Arrêt, le déplacement du chariot est stoppé.

En détaillant chaque phase comme suit :

Marche avant du chariot :

- Actionnez le bouton « Marche » ; Le moteur démarre et atteint la fréquence requise  $F1=30\text{Hz}$  en 6 secondes ;
- Actionnez le bouton "A" ; Le moteur accélère et atteint la fréquence  $F2=50\text{Hz}$  ;

- Actionnez le bouton "B" ; Le moteur freine et atteint la fréquence  $F3=20\text{Hz}$  au bout de 6 seconds ;
- Actionnez le bouton "C" ; Le moteur freine et s'arrête en 6 second ;
- Le TIMER  $T1=10\text{s}$  activer ;

Marche arrière du chariot :

- Temps de 10s écouler, Le moteur démarre mais en sens contraire et atteint la fréquence requise  $F4=50\text{Hz}$  en 6 secondes ;
- Actionnez le bouton "Arrêt" ; Le moteur s'arrête au bout d'environ 3 secondes (Temps de lissage).

### Travail demandé :

Pour chacun des modes de fonctionnement décrits ci-dessous on demande de :

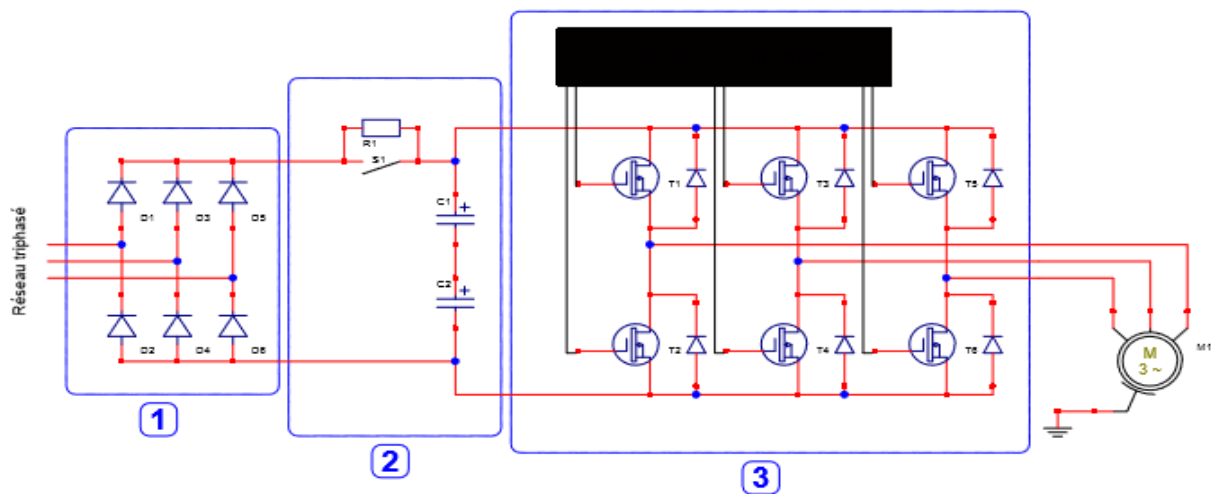
- 1- Etablir le GRAFCET de point de vue système.
- 2- programmer le graphcet sur step7. (Le programme est donné en détails dans l'annexe A)
- 3-utiliser la sortie analogique pour donner la vitesse de référence.
- 4-Paramétrer le ATV312 afin de l'adapter à la manipulation.
- 5- Simuler le fonctionnement du système

- **Paramétrage du variateur de vitesse ATV 312.**

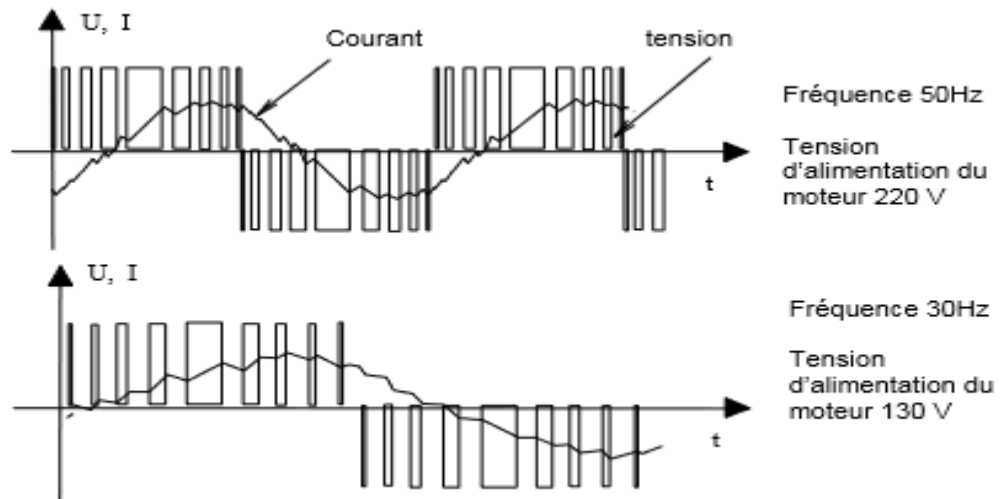
SYSTEME UTILISE :	Variateur de vitesse ATV 312
OBJECTIF PRINCIPAL :	Mettre en service et paramétrer un variateur de vitesse associé à un moteur asynchrone à cage.
OBJECTIFS INTERMEDIAIRES :	-Décoder les informations à partir de documents techniques - Réaliser les câblages permettant d'effectuer un mesurage
COMPETENCES ATTENDUES A L'ISSUE DE LA SEQUENCE	- justifier le choix du convertisseur d'énergie - configurer le modulateur d'énergie. - effectuer les réglages externes du modulateur d'énergie

PEDAGOGIQUE :	- vérifier, à l'aide de mesurages pertinents, les performances attendues du système de conversion
MATERIELS MIS A VOTRE DISPOSITION :	- Appareils de mesure (multimètres ...). - Système « ensemble Banc moteur + Altivar ». - Dossier technique ATV.

### 1. Schéma de principe d'un variateur :



1. Rechercher le nom des différents symboles électroniques utilisés dans le schéma ci-dessus. Présenter votre réponse sous forme d'un tableau : Symbole | Nom
2. Rechercher la fonction des trois sous-ensembles 1, 2 et 3 et trouver leur représentation sous forme de schéma bloc puis dessiner sur votre copie le schéma fonctionnel du variateur.
3. Tension et courant de sortie du variateur pour deux fréquences différentes :



- Donner la loi reliant la fréquence d'alimentation du moteur  $f$  et la vitesse de synchronisme  $n_s$ .
- Donner le type de commande de l'onduleur.

## 2. Caractéristiques du variateur utilisé

1. Relever la tension du réseau disponible
2. Relever la référence du variateur de vitesse utilisé :
3. Quelle est la tension d'alimentation nécessaire à ce variateur ?

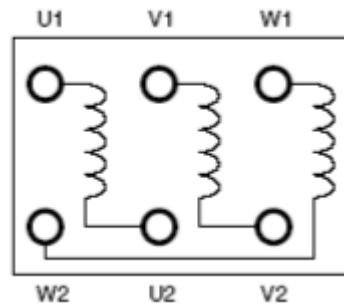
Tension	Nombre de phases

## 3. Caractéristiques du moteur utilisé

1. Compléter le tableau pour le moteur utilisé :

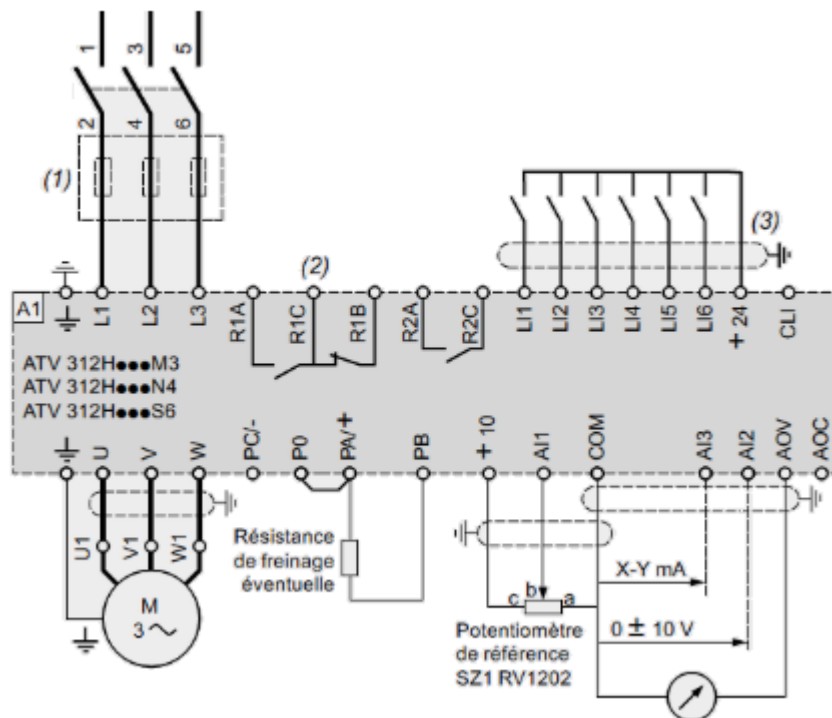
Tension (Y/ $\Delta$ )	Fréquence $f$	Vitesse $N$	Puissance $P$ (kW)	Rendement $\eta$	Courant (Y/ $\Delta$ )	Cos $\phi$

2. Quel est le couplage moteur à effectuer ?
3. Dessiner ci-dessous le raccordement de la plaque à bornes du moteur : (couplage  $\Delta$  ou  $\gamma$ )



#### 4. Installation et câblage du variateur

Câbler la partie puissance du variateur (voir figure suivante) :



Câbler les éléments de commande du variateur permettant : validation ; Marche/Arrêt ; Avant/Arrière. (Utiliser les entrées logiques : LI1 à LI6).

#### 5. Paramétrage du variateur de vitesse

Paramétrer les informations tirées de la plaque signalétique du moteur dans le variateur

Programmer les entrées logiques pour varier la vitesse du moteur et le sens de rotation.

LI1 marche avant

LI2 marche arrière

LI3 vitesse lente 5Hz

LI4 vitesse moyenne 10 Hz.

LI5 vitesse rapide 15 Hz.

Utiliser l'entrée logique LI6 pour faire un freinage par injection de courant continu.

Pour configurer ces paramètres sur le ATV 312 consulter le Chapitre III pages : 60,61,62.


### **La manipulation :**

Une fois le programme implémenté dans l'API, on peut passer à la manipe.

Les instructions à suivre :


- Assurez que la maquette est sous tension, et que les deux disjoncteurs triphasé et monophasé sont fermés à l'intérieur de l'armoire.
- Attendez que l'HMI affiche la figure ci-dessous :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
**Ecole Nationale Polytechnique**



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique


**Département d'ELECTROTECHNIQUE**  
Laboratoire de Commande des processus



مختبر التحكم في العمليات التكنولوجية  
**LCP**  
Laboratoire de Commande des Processus

**Réalisation d'une armoire à variateur de vitesse commandé par automate programmable.**

**Amar TIOURGUIOUNE et Mohamed Hocine SOUMATIA**  
Sous la direction de Pr. El Madjid BERKOUK et Pr MO.MAHMOUDI

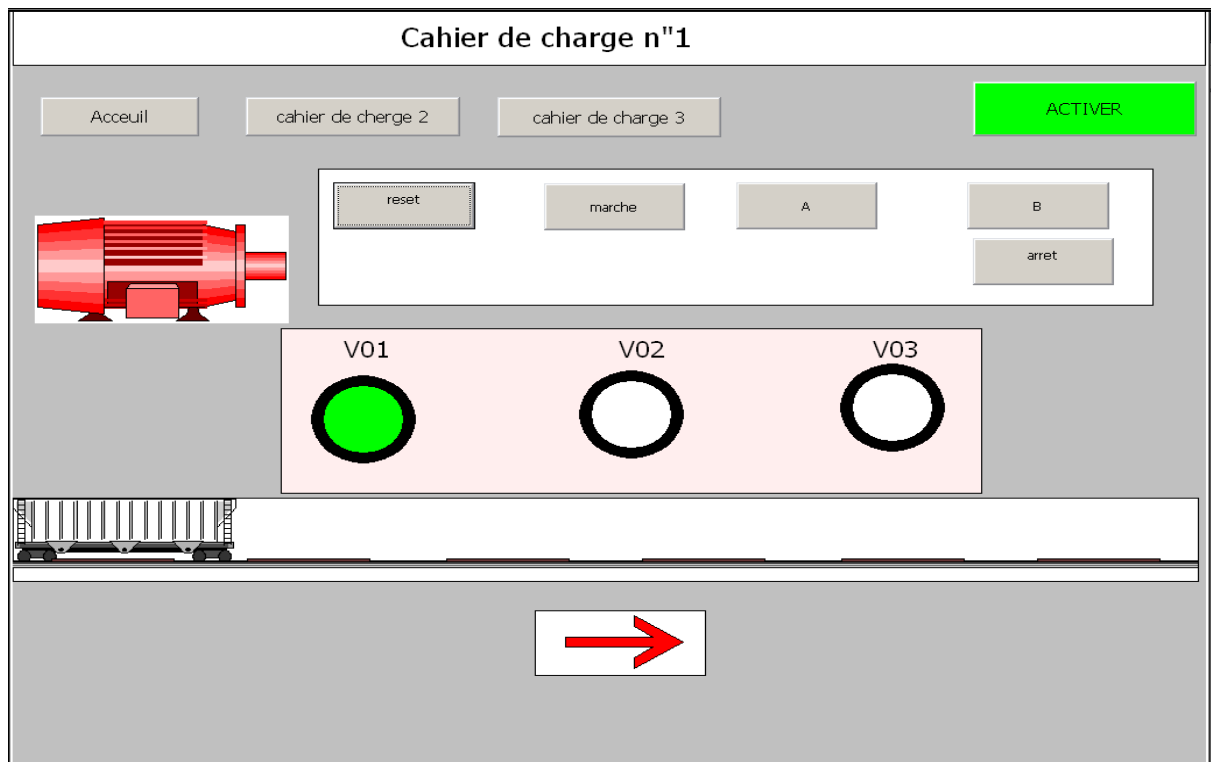


- Appuyez sur suivant puis choisissez le 1<sup>er</sup> cahier de charge.



- Dès que vous appuyez sur « cahier de charge 1 », vous serez orienté vers la vue suivante :





Appuyez sur « ACTIVER » pour activer le cahier de charge 1 et suivre les différentes instructions donner sur le cahier des charges.

Assurez-vous que la partie opérative « chariot réel » est en synchronisme avec celui simuler sur l'interface de supervision IHM.