

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE

DEPARTEMENT DE GENIE ELECTRIQUE

N° : Auto-05



DOMAINE : SCIENCES ET TECHNOLOGIE

FILIERE : AUTOMATIQUE

OPTION : AUTOMATIQUE ET SYSTEMES

**Mémoire présenté pour l'obtention  
Du diplôme de Master Académique**

**Par:**

SMAINI Marwa

MAMI Soumia

**Intitulé**

**Pilotage et supervision d'un moteur à courant  
continu par l'automate programmable  
industriel (API)**

**Soutenu devant le jury composé de :**

Dr. GHEDBAN Ismail	Université Mohamed Boudiaf - M'sila	Président
Dr. GHELLAB Mohamed Z.	Université Mohamed Boudiaf - M'sila	Encadreur
Dr. BOUZIDI Riad	Université Mohamed Boudiaf - M'sila	Examineur

**Année universitaire : 2022/2023**

# *Dédicace*

*Je dédie ce modeste travaillé :*

*A mes très chers parents*

*Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos  
innombrables sacrifices*

*Je souhaite que ce travail soit pour vous une petit compensation et reconnaissance  
envers ce qui vous avez fait d'incroyable pour moi.*

*Amés très chers frères, mes très chères amies et à ma binôme SMAINI.MRWA  
avec qui j'ai passé de meilleurs moments qui restent de bon souvenir pour toujours  
et tout la vie*

*Je vous dire tout simplement un grand merci, Je vous aime.*

*Soumia .Mami*

# *Dédicace*

*Avec tous mes sentiments de respect, avec l'expression de ma reconnaissance, je  
dédie ma remise de diplôme et ma joie*

*Mon paradis, à la prunelle de mes yeux, à la source de ma joie et mon bonheur, ma  
lune et le fil d'espoir qui allumer mon chemin, ma moitié maman.*

*À celui qui m'a fait une femme, ma source de vie, d'amour et d'affection, à mon  
support qui était toujours à mes côtés pour me soutenir et m'encourager, à mon  
prince papa.*

*À mes frères pour l'amour qu'il me réserve.*

*À mes sœurs qui n'ont pas cessée de me conseiller, encourager et soutenir tout au  
long de mes études.*

*À tous les membres de ma grande famille, mes tantes et mes cousines.*

*À mon encadreur Mr Ghellab Mohammed Zinelaabidine.*

*Sans oublier ma binôme Soumia pour son soutien moral sa patience et sa  
compréhension tout au long de ce projet.*

*À tout ce qui ont participé à ma réussite et à tous qui m'aiment.*

*SMAINI MARWA*

# *Remerciement*

*Tout d'abord, Nous remercions dieu le tout puissant qui nous a donné le courage et la volonté afin de mener à bien ce travail. Nous voudrions exprimer notre gratitude et remerciements à Mr Ghellab Mohamed qui a accepté d'encadrer nous et de nous diriger le long de ce modeste travail. Notre remerciement va aussi aux étudiants de doctorat (Laboratoire génie électrique et Automatique) de L'université, Mr khoudja et Mr Zougar pour l'aide qui n'a cessé de nous prodiguer.*

*D'une manière générale, tous les enseignants du département De génie électrique sont remerciés, chacun par son nom, pour durant notre Formation.*

# *Sommaire*

## *Table des matières*

<b>INTRODUCTION GENERALE.....</b>	<b>1</b>
<b>I. Généralité sur les automates programmables industriels(API) .....</b>	<b>1</b>
<b>I.1. Introduction .....</b>	<b>2</b>
<b>I.2 Historique.....</b>	<b>2</b>
<b>I.3 Définition : .....</b>	<b>3</b>
<b>I.4 Avantages d'un API : .....</b>	<b>3</b>
<b>I.5 Inconvénients d'un API : .....</b>	<b>3</b>
<b>I.6 Structure d'un API : .....</b>	<b>4</b>
<b>I.6.1 Structure interne.....</b>	<b>4</b>
<b>I.6.2 Structure externe : .....</b>	<b>5</b>
<b>I.7 Langages de programmation : .....</b>	<b>5</b>
<b>I.8 Nature des informations traitées par l'automate : .....</b>	<b>7</b>
<b>I.9 Le principe de fonctionnement : .....</b>	<b>12</b>
<b>I.10. Les Caractéristique d'un API : .....</b>	<b>13</b>
<b>I.11. Critère de choix d'un API : .....</b>	<b>13</b>
<b>I.12. Description de L'automate S7-314 : .....</b>	<b>14</b>
<b>I.12.1Caractéristiques de L'automate S7-314 : .....</b>	<b>15</b>
<b>I.14 Conclusion : .....</b>	<b>16</b>
<b>II. Généralité sur le logiciel STEP7 :.....</b>	<b>17</b>
<b>II.1 Introduction : .....</b>	<b>17</b>
<b>II.2 Historique : .....</b>	<b>17</b>
<b>II.4 Définition de logiciel step7 : .....</b>	<b>19</b>
<b>II.5 Les Langages de programmation : .....</b>	<b>19</b>
<b>II.6 Elaboration du programme S7 (Partie Software) : .....</b>	<b>20</b>
<b>II.7 Créations d'un projet Step7 : .....</b>	<b>21</b>
<b>II.8 le logiciel S7-PLCSIM : .....</b>	<b>27</b>
<b>II.8.1 Présentation de S7-PLCSIM : .....</b>	<b>27</b>
<b>II.8.2 Mise en route du logiciel S7-PLCSIM : .....</b>	<b>27</b>
<b>II.9 Exemple d'application de Création et d'édition d'un projet S7 : .....</b>	<b>28</b>
<b>II.9.1 L'objectif de cette application : .....</b>	<b>28</b>

II.9.2 Cahier de charge :	29
II.9.3 Fonctionnement :	29
II.9. 4. L'adressage de notre GRAFECET :	30
II.9.5. Simulation de modules :	33
II.9.5. Chargement dans le système cible à partir de la PG/PC :	33
II.9.6. Surveillance du fonctionnement et diagnostic du matériel :	34
II.10 Conclusion :	35
III.1 Introduction :	37
III.2 Progiciel de conception et configuration d'interface Win CC flexible.....	37
III.2.1 Présentation Générale .....	37
III.3. La supervision :	38
III.3.1 Architecteur d'un réseau de supervision :	38
III.3.2. Constitution d'un système de supervision :	39
III.4 Utilisation de SIMATIC Win CC Flexible :	39
III.4.1. Win CC flexible Runtime: .....	39
III.4.2 La liaison Win CC avec PLC Sim :	40
III.5 Principalement des vues : [19] .....	41
III.6 Elément du progiciel Win CC flexible.....	42
III.6.1 Intégration de Win cc avec step7 :	45
III.6.2 Création des Vues :	45
III.6.3 Utilisation de variable :	47
III.6.4 Etablissement de la liaison automate HMI :	47
III.6.5 Compilation et Simulation :	48
III.7 Conclusion :	49

*Simulation et commande d'un moteur à courant continu(MCC)41* IV.1. Introduction ...49

IV.1 Définition :	49
IV.3 Les avantages de Moteur à courant continu :	50
IV.4 Modélisation du moteur à courant continu :	50
IV.5 Commande d'un moteur à courant continue.....	51
IV.5.1Commande en boucle ouverte avec automate programmable.....	51
IV.5.2 Rappels sur moteur à courant continu .....	52
IV.5.3 Description du matériel.....	53
IV.6 Description du programme :	55

<b>IV.6.1</b>	<b>Chargement de la consigne :</b>	<b>55</b>
<b>IV.6.2</b>	<b>Lecture de la vitesse</b>	<b>56</b>
<b>IV.6.3</b>	<b>Calcul de la position</b>	<b>Error! Bookmark not defined.56</b>
<b>IV.7</b>	<b>Création du projet S7 pour la commande d'un MCC :</b>	<b>58</b>
<b>IV.7.1</b>	<b>Insertion des stations dans SIMATIC MANAGER</b>	<b>58</b>
<b>IV.7.2.</b>	<b>Configuration du matériel dans STEP7</b>	<b>58</b>
<b>IV.8</b>	<b>Application et visualisation de la commande :</b>	<b>59</b>
<b>IV.9</b>	<b>CONCLUSION</b>	<b>62</b>
	<b>CONCLUSION GENERALE</b>	<b>63</b>

## *Liste des figures*

Figure I.1 : Le sigle de la société Siemens AG -----	2
Figure I.2 : Automate compact (EATON- ESSY control).-----	5
Figure I.3 : Automate modulaire (Siemens) -----	5
Figure I.4 : Les éléments de FRAFCET -----	6
Figure I.5 : Programme LD -----	7
Figure I.6 : Fonction de comptage et de temporisation par LD -----	7
Figure I.7 : Automate SIMATIC S7-314 -----	14
Figure II.1 : L'API S7-400-----	18
Figure II.2 : L'API S7-300-----	18
Figure II.3 : Station PC et PG\PC-----	Error! Bookmark not defined.19
Figure II.4 : Mode de présentation des langages basiques de programmation step7 [16] --	20
Figure II.5 : Programmation structure. -----	20
Figure II.6 : Programmation avec logiciel STEP7.-----	21
Figure II.7 : Lancera l'assistant STEP7. -----	22
Figure II.8 : Assistante de STEP7.-----	22
Figure II.9 : Choix de CPU. -----	22
Figure II.10 : Choix de bloc et langage utilisé. -----	23
Figure II.11 : Fenêtre de nommage de projet.-----	23
Figure II.12 : Choix de matériel. -----	24
Figure II.13 : Fenêtre de configuration du matériel.-----	25
Figure II.14 : Tableau de mnémonique. -----	25
Figure II.15 : Fenêtre des blocs software-----	25
Figure II.16 : Fenêtre de bloc OB1 -----	26
Figure II.17 : Création de programme S7. -----	26
Figure II.18 : Mode de simulation.-----	27
Figure II.19 : Station de Transfer des pièces.-----	28
Figure II.23 :Le programme étant écrit, on doit le charger dans le CPU.-----	33
Figure II.24 : Diagnostique du matériel [13].-----	34
Figure III.3 : Schéma de la liaison et pelotage (MPI)-----	41
Figure III.4 : Vue d'ensemble du progiciel Win CC flexible-----	42
Figure III.5 : Etapes pour la conception d'une interface via Win CC flexible -----	43
Figure III.6 : l'assistante de Win CC flexible-----	43
Figure III .7 : Création d'un nouveau projet-----	44
Figure III.8 : Sélection du pupitre -----	44
Figure III.9 :l'intégration dans le step7-----	45
Figure III.10 : fenêtre de création des vues -----	46
Figure III.11 : Sélection les objets-----	46
Figure III.12 : Edition de variable dans le Win CC flexible-----	47
Figure III.13 : éditeur de la liaison-----	48
Figure III.14 : Démarrage de système Runtime-----	48
Figure III.15 : L'HMI de notre station -----	49

<b>Figure IV.1 : Représentation d'une MCC</b>	<b>50</b>
<b>Figure IV.2 : Schéma électrique d'un moteur à courant continu</b>	<b>50</b>
<b>Figure IV.3 : Commande en BO avec boucle de mesure.</b>	<b>51</b>
<b>Figure IV.4: Le moteur et le circuit d'amplification</b>	<b>52</b>
<b>Figure IV.5 : Moteur CC à aimant permanents</b>	<b>53</b>
<b>Figure IV.6 : Moteur à courant continu</b>	<b>53</b>
<b>Figure IV.7 : Schéma électrique du circuit d'amplification.</b>	<b>54</b>
<b>Figure IV.20 : Schéma de MOSFET '11N80C3'</b>	<b>Error! Bookmark not defined.</b>
<b>Figure IV.8 : Organigramme de la commande en BO.</b>	<b>55</b>
<b>Figure IV.9 : Organigramme de chargement de la consigne.</b>	<b>56</b>
<b>Figure IV.10 : Organigramme de lecteur de la vitesse.</b>	<b>56</b>
<b>Figure IV.12 : Organigramme de l'application finale.</b>	<b>57</b>
<b>Figure IV.13 : prototype de la commande par API.</b>	<b>57</b>
<b>Figure IV.14 : Création du projet</b>	<b>58</b>
<b>Figure IV.15 : Configuration matériel.</b>	<b>58</b>
<b>Figure IV.16 : Vue générale crée pour la HMI</b>	<b>59</b>
<b>Figure IV.17 : La tension et la vitesse du système Entrée/Sortie).</b>	<b>59</b>
<b>Figure IV.18 : La réponse de la vitesse et tension correspondante</b>	<b>60</b>
<b>Figure IV.19 : Bloc UNSCALE</b>	<b>61</b>
<b>Figure IV.20 : Bloc SCALE</b>	<b>61</b>

## *Liste des tableaux*

<b>Tableau I.1 : Description du module d'entrée TOR SM321 DI 16X 24 VCC</b>
<b>Tableau I.2 : Description du module de sortie TOR SM322 DO 16X 24VCC</b>
<b>Tableau I.3 : Description du module d'E/S analogique SM 334 AI 4/AO 2 x 8/8 bits</b>
<b>Tableau I.4 : Correspondance valeurs analogiques tension</b>
<b>Tableau I.5 : Correspondance valeurs analogiques courant</b>
<b>Tableau II.1 : L'adressage absolu des entrées/sorties de la première application</b>
<b>Tableau II.2 : Les actions dans le S7-GRAPH.</b>

## *Liste des abréviations*

**API** : Automate Programmable Industriel

**MCC** : Moteur à Courant Contenu

**PLC** : Programmable Logic control

**SIMATIC MANAGER** : Siemens automatic

**PLCSIM** : Programmable logic controller simulator

**E/S** : Entrée et sortie

**IHM** : Interface Homme-Machine

**HMI** : Humain machine interface

**CONT** : langage Contact

**LIST** : langage liste

**LOG** : langage logigramme

**GRAF** : langage grafcet.

**TOR** : tout ou rien

**CPU** : central processing unit

**PG** : Programming consol

**MPI** : Multi point interface

**BO** : Boucle ouverte

## Résumé :

Ce travail permet de montrer les différentes possibilités qu'offre le STEP7, pour la commande séquentielle et la commande continuée en temps réel, afin d'implémenter des solutions et des tâches d'automatisation et de contrôle sur l'automate de la série 7 de la firme allemande SIEMENS S7-314.

Le STEP7 est une interface graphique facile à utiliser, les programmes sont organisés dans des blocs, ce qui facilite la subdivisant des problèmes et la mise à jour. Les différentes applications ont été réalisées dans un laboratoire au niveau du département d'automatisation de l'université de M'sila. Elles montrent que l'automate S7-314 offre une large possibilité de commande et d'automatisation.

**Mot clef :** STEP7, automate programmable S7-314, WinCC, commande d'un moteur cc.

### المخلص :

هذا العمل يعتمد اساسا على تقديم مختلف الاحتمالات باستخدام المبرمج ستابس. من اجل التحكم التعاقبي والتحكم المستمر في الوقت الحالي بعد ايجاد الحلول والاضافات في النظام. و S7-314 الاليات والمراقبة في سلسلة الآلية 7 نوع سيمنس.

الستاب سات هو الواجهة البيانية سهلة الاستعمال. وأيضا الأنظمة المنظمة داخل مجموعة عناصر من أجل التسهيل في حل المشاكل و التحديث في مختلف التطبيقات التي أنجزت في المخبر على مستوى الكلية في جامعة المسيلة الآلية المبرمجة من نوع 7-314 تسمح للحصول على نطاق واسع من الاحتمالات للتحكم في الآليات.

### الكلمات المفتاحية :

ستاب سات، الآلية المبرمجة S7-314 , المبرمج Win CC، التحكم في المحرك ذو التيار المستمر.



# ***INTRODUCTION GENERALE***

### Introduction générale

Aujourd'hui, l'automatisation tient une place très importante dans le domaine industriel, elle est devenue en peu de temps un outil indispensable et une solution pour la majorité des entreprises industrielle qui doivent répondre aux exigences d'avantages du client, Le but d'un système automatique est réalisé des tâches trop complexes ou dangereuses pour l'homme, faire des tâche répétitives et pénibles et d'accroître la précision [1].

La supervision et pilotage d'un moteur à courant continu par l'automate programmable (API) sont des techniques importantes pour contrôler et automatiser les processus industriels. Ces techniques permettent de contrôler la vitesse, la direction et la position du moteur de manière précise et fiable, ce qui peut améliorer l'efficacité et la sécurité des processus industriels [2].

Le but de travail est de faire de mise en œuvre automates programmable industriel de la famille SIMATIC S7-300 pour la commande de MCC avec boucle de mesure et de traitement. Ce document englobe du logiciel de programme STEP7 et la supervision par le Win CC, qui permettent d'avoir une meilleure qualité des produits en plus de sécurité et de flexibilité des processus.

Notre travail est réparti en quatre chapitres, Le premier chapitre traite les connaissances de base après une brève description de toute la gamme SIMATIC S7 des automates SIEMENS.

Dans le deuxième chapitre, on aborde l'environnement de développement STEP7. Pour mettre en relief cet environnement, on traite une application en décrivant les différentes étapes pour la création de ce projet. La fin de ce chapitre est consacrée à la commande d'une station pneumatique et sa mise en œuvre dans STEP 7.

L'exploitation visuelle dans les milieux industriels n'a pas été oubliée dans le troisième chapitre. Une présentation du logiciel Win CC flexible de SIEMENS sera abordée, en vue de configurer une interface Homme/Machine nécessaire à la commande, le diagnostic et la visualisation à distance.

Ces trois derniers chapitres constituent une préparation au projet qui nous a été soumis. Le dernier chapitre présent une application sur le moteur à courant continue avec boucle de mesure et de traitement (commande en boucle ouverte).

On termine par une conclusion générale qui montre les difficultés qu'un étudiant à ses débuts peut rencontrer, des enseignements qu'il en tire, sans oublier les perspectives.

***Chapitre I :***  
***Généralité sur les automates***  
***programmables industriels(API)***

## I.1. Introduction

L'automate programmable est un composant électronique adapté aux conditions industrielles. Il intègre des fonctions et des modules qui facilitent son interaction avec l'environnement extérieur. Vu le nombre important de marques disponibles, Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe.

Ce chapitre consiste à décrire les généralités sur les automates programmables, l'automate S7-314, le principe de fonctionnement, l'architecture, la caractéristique, Critères de choix et les langages de programmation de l'automate programmable.

## I.2 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.



**Figure I.1** : Le sigle de la société Siemens AG

Dans le cadre de son expansion, Siemens crée le 28 janvier 1972, le consortium Uni data. Ce projet européen permet, dans les années 1970, l'émergence d'une grande industrie informatique européenne. Les compétences de trois participants ont été mises en commun. La maîtrise d'œuvre, l'architecture des machines et le logiciel ont été attribués à la Compagnie Internationale

pour l'Informatique (CII), la technologie électronique revenait à Philips tandis que Siemens se chargeait des périphériques mécaniques.

En 1975 : la France abandonne unilatéralement l'accord Uni data, CII fusionne avec Honeywell-Bull, Philips délaisse l'informatique et Siemens rejoint Fujitsu pour devenir, aujourd'hui, un des plus grands constructeurs mondiaux [11].

### **I.3 Définition :**

Les Automates Programmables Industriels (API), ou en anglais Programmable Logic control (PLC), sont apparus aux Etats-Unis vers 1968 à la demande de l'industrie automobile Américaine (Général Motors) .l'API est le constituant le plus répandu dans les automatismes

L'automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique [3].

### **I.4 Les Avantages d'un API :**

- Simplification du câblage.
- Modification du programme facile à effectuer par rapport à une logique câblée.
- Enormes possibilités d'exploitation.
- Fiabilité professionnelle.
- Les éléments qui les composent et son particulièrement robustes.
- Ils possèdent des circuits électroniques optimisés pour s'interface avec les entrées et les sorties physiques du système

### **I.5 Les inconvénients d'un API :**

Utilisation d'un personnel formé à cette technologie (technicien de maintenance, de control...).

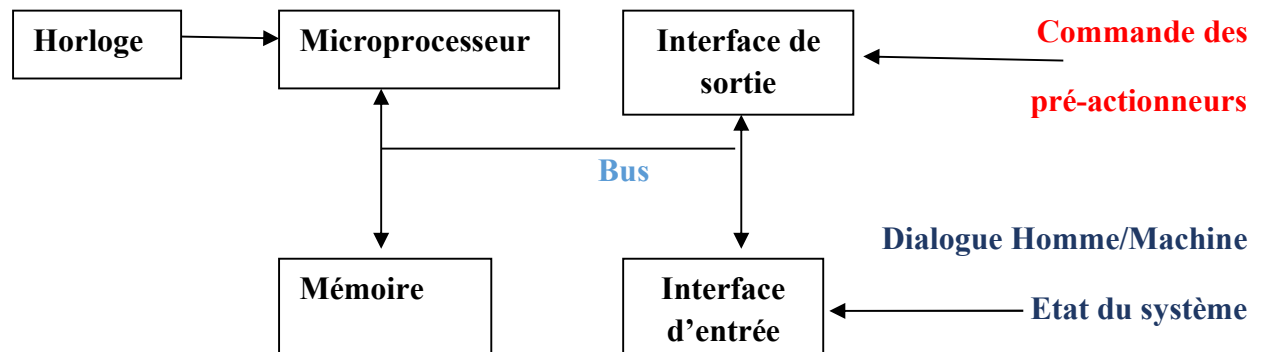
Le coût de la réalisation reste élevé même si le fonctionnement est simple.

La consommation d'énergie.

## I.6 Structure d'un API :

### I.6.1 Structure interne

La structure interne peut se présenter comme suit :



**Figure I.2 :** Structure interne d'un API

- **le microprocesseur :** Le microprocesseur réalise toutes les fonctions logique (ET, OU) il est connecté à l'autre élément (mémoire et interface E/S) par des liaisons parallèles appelées 'Bus' qui véhiculent les informations sous forme binaire.
- **Mémoire :** Recevoir les informations issues des capteurs d'entrées, les informations générées par le processeur et recevoir et conserver le programme du processus.
- **Interfaces d'entrées :** Les interfaces d'entrées sont destinées à recevoir l'information en provenance des capteurs et Traiter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.
- **Interfaces de sorties :** Les sorties transmettent des informations aux pré-actionneurs (relais, électrovannes, etc.) et aux éléments de système, sont destinées à commander les pré-actionneurs et éléments de signalisations du système.

On distingue aussi des modules spéciaux utilisés suivant les besoins de l'application, comme :

- Des modules de communication Profibus, Ethernet pour dialoguer avec d'autres automates, des E/S des supervisions ou autres interfaces homme-machine (IHM).
- Des modules dédiés métiers tel que de comptage rapide.
- Des modules d'interface pour la commande mouvement, dits modules motion tel que démarreurs progressifs, variateurs de vitesse, commande d'axes.
- Des modules de dialogue (homme-machine) [4].

### I.6.2 Structure externe :

L'automate est conçu pour être de type modulaire pour d'éventuelles extensions d'E/S (dans la limite de la capacité de l'automate) mais il existe un autre d'aspect compact.

➤ **Aspect compact :**

IL intégré le processeur L'alimentation, les entrées et les sorties, il pourra réaliser certaines fonction supplémentaires (comptage rapide, E/S analogiques, etc.) et recevoir des extensions en nombre limité.



**Figure I.2 :** Automate compact (EATON- ESSY control).

➤ **Aspect modulaire :**

Le processeur, L'alimentation et les interfaces E/S résident dans des modules et sont fixées sur plusieurs racks.



**Figure I.3 :** Automate modulaire (Siemens)

### I.7 Langages de programmation :

La norme IEC 1131-3 (la Commission Électrotechnique Internationale) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces langages peuvent être divisés en deux grandes catégories :

•Langages graphiques :

- SFC « Séquentiel Funiculite Chart » ou GRAFCET
- LD « Ladder Diagramme » ou schéma à relais
- FBD « Function Block Diagramme » ou schéma par bloc

• Langages textuels :

- ST « structured text » ou texte structuré
- IL « Instruction List » ou Liste d'instructions

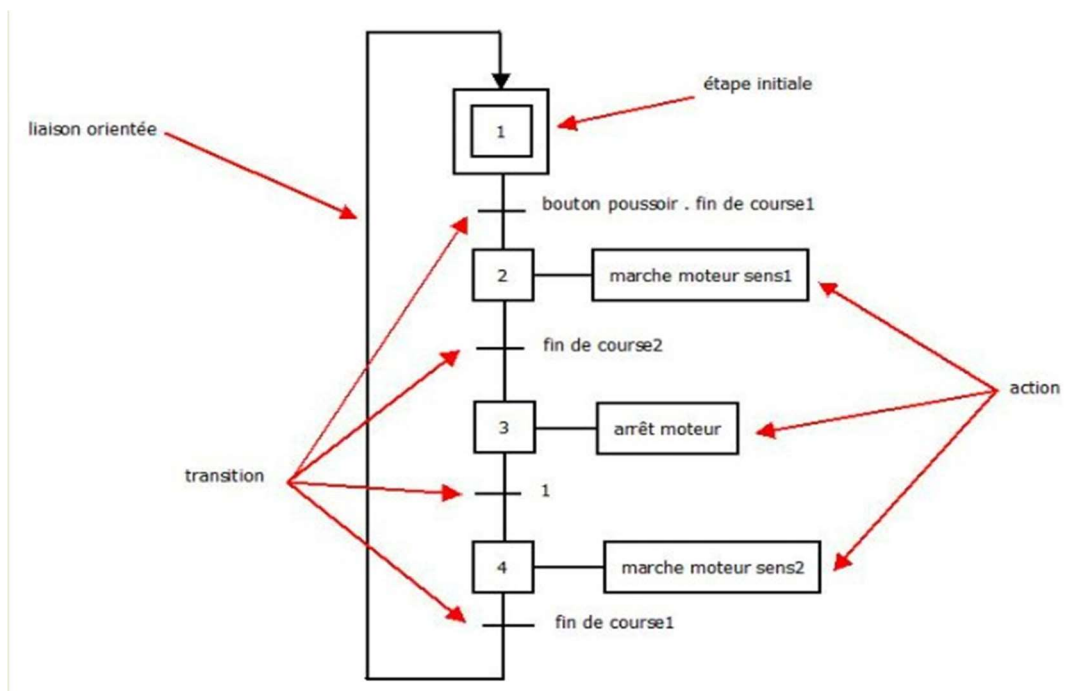


Figure I.4 : Les éléments de GRAFCET

Par exemple pour réaliser  $x = (a + b)(\bar{c} + \bar{d}e)$  la fonction logique, voici le programme LD :

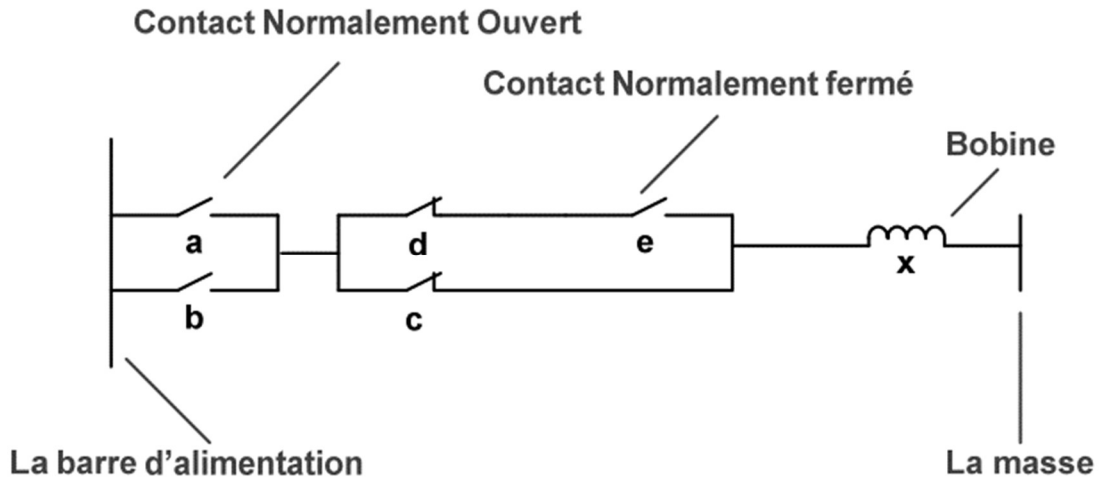


Figure I.5 : Programme LD

Le langage LD propose d'autre type de fonction tel que les fonctions de comptages et de temporisations, les fonctions arithmétiques et logiques, les fonctions de comparaison et de transfert. Par exemple pour réaliser la fonction, on utilise directement la fonction déjà disponible.

$$z = (x \geq y)$$

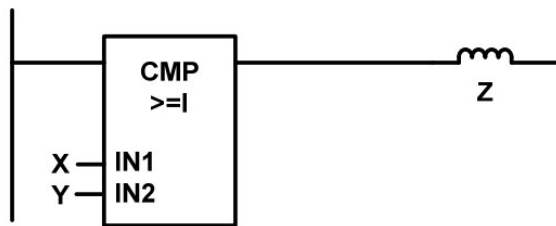


Figure I.6 : Fonction de comptage et de temporisation par LD

**I.8 Nature des informations traitées par l'automate :**

Les informations traitées par un API peuvent être de type :

**Tout ou rien (T. O. R) ou logique :** l'information ne peut prendre que deux états (0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir...

➤ **Modules des sorties TOR**

Il s'agit du SM321 DI 16x 24V cc, il contient 16 entrées Tout Ou Rien de 24 V cc munis par une séparation galvanique par groupes de 8.

Le tableau suivant résume ses caractéristiques techniques :

**Tableau I. 6 :** Description du module d'entrée TOR SM321 DI 16x 24 VCC

<b>Dimensions &amp; Poids</b>	
Poids (200 g)	L x H x P 40 x 125 x 117
<b>Caractéristiques spécifiques</b>	
Nombres d'entrées	16
Signalisation de l'état	Une LED verte par voie
Longueur max du câble	600 m
<b>Caractéristiques pour la sélection d'un capteur</b>	
Tension d'entrée	
- Valeur nominale	24 V cc
- Pour signal '1'	13 à 30 V
- Pour signal '0'	De -30 jusqu'à +5 V
Courant d'entrée	
- Pour signal '1'	7 mA
Temporisation d'entrée	
- De '0' à '1'	1.2 à 4.8 ms
- De '1' à '0'	1.2 à 4.8 ms

➤ **Modules de sorties TOR**

Le SM322 DO 16x24 V cc comprend 16 sorties Tout Ou Rien de 24 V cc munis d'une séparation galvanique par groupes de 8, il délivre en sortie un courant de 0.5 A.

Le tableau suivant résume ses caractéristiques techniques :

**Tableau I.7 :** Description du module de sortie TOR SM322 DO 16x 24VCC

<b>Dimensions &amp; Poids</b>	
L x H x P (mm )	40 x 125 x 117
Poids	190 g
<b>Caractéristiques spécifiques</b>	
Nombres de sorties	16
Signalisation de l'état	Une LED verte par voie
Longueur max du câble	600 m
<b>Caractéristiques pour la sélection d'un actionneur</b>	
Tension de sortie	24 V cc
- Pour signal '1'	
Courant de sortie	0.5 A
- Pour signal '1'	
o Valeur nominale	
o Plage admissible	5 mA à 0.6 A
- Pour signal '0'	Max. 0.5 mA
Temporisation de sortie	Max. 100µs
- De '0' à '1'	
- De '1' à '0'	Max. 500 µs
Plage de résistance de charge	Oui, par hachage électronique.
Protection des sorties contre les courts-circuits	

**Analogique :** l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)

**Numérique :** l'information est contenue dans des mots codés sous forme binaire. C'est le type d'information délivrée par un ordinateur ou un module intelligent. [6]

#### **Module d'entrées sorties analogiques**

Le module SM 334 AI 4/AO 2 x 8/8 bits présente les propriétés suivantes :

- 4 entrées, 2 sorties
- résolution 8 bits

- étendue de mesure au choix 0 à 10 V ou 0 à 20 mA
- étendue de sortie au choix 0 à 10 V ou 0 à 20 mA
- choix libre entre tension et courant s
- ans séparation galvanique par rapport au couplage de bus interne
- séparation galvanique par rapport à la tension d'alimentation

Voici dans le tableau ci-dessous quelques spécifications techniques

**Tableau I.8 :** Description du module d'E/S analogique SM 334 AI 4/AO 2 x 8/8 bits

<b>Dimensions &amp; Poids</b>	
Poids (285 g )	L x H x P 40 x 125 x 117
<b>Caractéristiques spécifiques</b>	
Nombres d'entrées	4
Nombres de sorties	2
Signalisation de l'état	Une LED verte par voie
Longueur max du câble	max. 200 m
<b>Formation des valeurs analogiques</b>	
Résolution	8 bits
Temps de conversion par voie	500 $\mu$ s
<b>Caractéristiques pour la sélection d'un capteur</b>	
Etendue d'entrée (valeurs nominales/résistance d'entrée)	0 à 10 V/100k
- Tension	0 à 20 mA/5
- courant	
Tension d'entrée admissible	Max. 20V permanents
Limite de destruction	75 V durant max. 1s
Courant d'entrée admissible	40 mA
<b>Caractéristiques pour la sélection d'un actionneur</b>	
Etendue d'entrée (valeurs nominales/résistance d'entrée)	
- Tension	0 à 10 V
- courant	0 à 20 mA
Résistance de charge (dans l'étendue de sortie nominale)	
- sortie de tension	Min. 5 K
○ charge capacitive	Max. 1 $\mu$ F
- sorties de courant	Max. 300

○ charge inductive	Max. 1 mH
Tension à vide	Max. 15 V
Protection contre les courts-circuits	Oui
Limite de destruction	
- Tension	Max 15 V permanents
- Courant	max. DC 50 mA

**Remarque :**

- Les entrées et sorties du module sont adressées à partir de l’adresse initiale du module.  
L’adresse d’une voie correspond à l’adresse de début du module plus un déport d’adresse (2octets).
- Les voies d’entrées non utilisées doivent être court-circuitées. Par cette mesure, on obtient une immunité optimale aux perturbations pour le module analogique.
- Les voies de sortie libres doivent être laissées en l’air.
- Le SM 334 AI4/AO2 x 8/8bits n’a pas d’étendues de mesure négatives.

La correspondance valeurs analogiques tension est représentée dans le tableau suivant :

**Tableau I.9 :**Correspondance valeurs analogiques tension

Système			Etendue de tension	
Pourcentage %	Décimale	Hexadécimale	Tension	Domaine
117.56 %	32511	7EFF	11.76 V	Domaine de dépassement
27649	6C01			
100%	27648	6C00	10 V	Etendue nominale
75%	20736	5100	7.5 V	
0.0036%	1	1	361.7 µV	
0%	0	0	0 V	

-1	FFFF	0 V	Impossible la tension min. d'entrées ou de sortie est limitée à 0 V	
-32768	8000	0 V		

La correspondance valeurs analogiques courant est représentée dans le tableau suivant

**Tableau I.10 :** Correspondance valeurs analogiques courant

Système			Etendue de tension	
Pourcentage %	Décimale	Hexadécimale	Courant	Domaine
117.56 %	32511	7EFF	32.52 mA	Domaine de dépassement
	27649	6C01		
100%	27648	6C00	20 Ma	Etendue nominale
75%	20736	5100	15 Ma	
0.0036%	1	1	723.4 nA	
0%	0	0	0 Ma	
	-1	FFFF	0 mA	Impossible le courant min. d'entrées ou de sortie est limitée à 0 mA.
	-32768	8000	0 mA	

**I.9 Le principe de fonctionnement :**

Les API très proches aux calculateurs industriels ordinateurs disposent d'une programmation autorisant des méthodes plus puissantes, plus complexes.

Au début de chaque cycle l'automate scrute systématiquement les entrées binaires pour transférer leur état. IL est segmentée en 3 secteurs qui contiennent respectivement les variables d'entrées, les variables de sorties et les variables internes utilisées aux cours de traitement, à la fin de la phase de lecture, le secteur des variables d'entrée de la mémoire de données contient

l'image de l'état des entrées binaires et l'automate commence une phase de traitement essentiellement réduite à des opérations booléennes sur les variables d'entrées et sur les variables internes utilisées pour les calculs intermédiaires. Ce traitement se traduit par la modification de certaines variables de sortie et leur remise à jour dans la zone correspondante de la mémoire de données.

A la fin du traitement, la zone des variables de sortie contient l'image de l'état de sortie qui doit être imposées aux différentes sorties binaires [5].

Le cycle de fonctionnement d'un automate se compose de trois étapes :

- l'obtention d'une entrée (lecture).
- le traitement des données.
- l'émission de commande (écriture).

Ces trois opérations sont effectuées en continu en quelques millisecondes par un automate (processus cyclique). Le cycle de fonctionnement d'un API comme suivant :

#### **I.10. Les Caractéristiques d'un API :**

- Tension d'alimentation (220V, 110V) continu ou alternatif.
- Taille et type de mémoire (Rom, Ram).
- Nombre d'entrée et sortie
- Langage de programmation

#### **I.11. Critère de choix d'un API :**

Le choix de l'automate va se faire suivant des critères qui peuvent être différents suivant les



personnes, les projets et les pays. Dans ce qui suit quelques critères à prendre en considération.

a) **Le critère de familiarité** : On est plus familier à une certaine marque d'automate, on a donc tendance à choisir un automate car on la maîtrise déjà.

b) **La renommée** : Certaines marques d'automates sont plus connues que d'autres ce qui témoigne de leur qualité de service.

c) **Le prix** : Pour un petit projet d'automatisation, il est choisi des micros ou mini automates, ce qui nous éviterons des dépenses exagérées.

d) **Cahier de charge** : Le choix définitif dépendra de ce cahier de charge et des spécificités techniques du projet. Donc il faut quantifier les besoins en nombres entrées/sorties, type de processeur (la taille de mémoire, la vitesse de traitement ...), fonctions ou modules spéciaux, fonctions de communication avec autre automate, supervision...

### I.12. Description de L'automate S7-314 :

L'automate programmable utilisé est le S7-313, c'est un automate modulaire constitué d'un module d'alimentation, une CPU, un module d'entrées TOR, un module de sorties TOR et un module d'E/S analogiques.



Figure I.7 : Automate SIMATIC S7-314

L'automate S7-314 est un automate modulaire destiné à des tâches d'automatisation moyennes et de hautes gammes. Gammes complètes de produits pour la résolution de tâches technologiques. Telles que le comptage, la mesure, repositionnement et la régulation.

- **Comptage /Mesure :** Comptage d'impulsions mono-coup périodique ou sans fin comptage, mesure de longueurs, de déplacements, de fréquences ou des périodes.
- **Positionnement :** Positionnement en boucle ouverte et asservissement de position, interpolation de plusieurs axes, synchronisation par réducteur électronique ou selon profile de came, moteur asynchrones, pas à pas, servomoteurs.
- **Régulation :** Régulation de température /pression /débit. Régulateurs pas à pas, à sortie cyclique et à action continue, régulation de maintien, de poursuite, en cascade, de rapport et de mélange.
- **Came électronique :** Action en fonction de la position, du sens ou du temps, anticipation dynamique, comportement d'hystérésis, temps de à partir de 1 µs [2].

**I.12.1Caractéristiques de L'automate S7-314 :**

- **La CPU**

Sur la plupart des CPU 300, on peut remarquer les éléments suivants : des LEDs, un commutateur, une pile, une carte mémoire et une interface MPI.

- **Commutateur du mode de fonctionnement**

**Tableau I.11 :** Position du commutateur du mode de fonctionnement

Position	Signification	Explications
RUN-P	RUN PROGRAM	La CPU traite le programme utilisateur. La clé ne peut être retirée dans cette position.
RUN	Mode de marche	La CPU traite le programme utilisateur. Ce dernier ne peut être modifié.
STOP	Mode d'arrêt	La CPU ne traite aucun programme utilisateur
MRES	Effacement général	Position instable du commutateur pour effacement général de la CPU.

- **Pile de sauvegarde :** Elle permet de sauvegarde le programme utilisateur (s'il n'est pas »sauvegardé sur une carte mémoire) et de plus grandes zones de données que celles

qui sont rémanentes dans les blocs de données sans pile, la durée de sauvegarde des données peut attendre une année.

- **Nota :** La CPU peut sauvegarder une partie des données, même sans pile. Le recours à une pile n'est nécessaire que si on désire étendre la rémanence à une plus grande quantité de données.
- **Carte mémoire :** Elle étend la mémoire de chargement de la CPU et permet de sauvegarder le programme utilisateur et les paramètres qui déterminent le comportement de la CPU et des modules. On peut également sauvegarder le système d'exploitation de la CPU sur une carte mémoire. Si le programme utilisateur a été sauvegardé sur une carte mémoire, il est conservé après une mise hors tension de la CPU même si celle-ci ne contient pas de pile de sauvegarde. Les cartes mémoires disponibles pour la 314 sont du type FEPR0M 5 V (16Ko, 32Ko...4Mo).
- **Interface MPI :** elle est utilisée pour connecter l'automate à la console de programmation PG, un pupitre opérateur OP ou pour la communication au sein d'un sous-réseau MPI. La vitesse de transmission typique (par défaut) est de 187,5 kbauds.

#### I.14 Conclusion :

Dans ce chapitre, nous avons présenté les généralités sur un système automatisé, l'historique de l'automate programmable industriel, son architecture (interne/externe), son principe de fonctionnement... Nous avons constaté aussi son importance pour le système automatisé dans le domaine industriel. Pour avoir un produit complet et fini dans une chaîne de production, ça nécessite une grande précision et des fois des tâches dangereuses. Afin de réaliser un produit parfait convainquant et satisfaisant, il faut introduire un ou plusieurs automates programmables dans les systèmes de production. Il faut ensuite quantifier les besoins :

- Nombre et types d'entrées / sorties : tension de sorties des capteurs et tension d'entrées des actionneurs ainsi que le nombre de voies nécessaire.
- Type de CPU : la taille mémoire, la vitesse de traitement et les fonctions spéciales intégrés...).
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus...).

***Chapitre II :***

***Le STEP 7***

## **II. Généralité sur le logiciel STEP7 :**

### **II.1 Introduction :**

Dans ce chapitre nous donnons une présentation générale sur le logiciel de programmation des automates programmables Step7, et les outils que nous allons utiliser pour développer notre application tels que le simulateur PLCSIM, la création et l'utilisation de le logiciel et à la fin nous présente une application de la programmation step7 « transfert des pièces dans un chaîne par PLC ».

### **II.2 Historique :**

Step7 est un logiciel de programmation de contrôleur logique programmable (PLC) développé par Siemens. Le développement de Step7 remonte au début des années 1990, lorsque Siemens a commencé à développer une nouvelle plate-forme de programmation pour ses contrôleurs logiques programmables SIMATIC.

La première version de Step7 est sortie en 1993 sous le nom de SIMATIC S7-300. Cette version initiale a introduit de nouvelles fonctionnalités, telles que le langage de programmation graphique “ logique à relais “, qui permettait aux programmeurs de concevoir des programmes en utilisant des symboles graphiques plutôt que du texte. Siemens a continuellement développé Step7 et ajouté de nouvelles fonctions au fil des ans. En 1996, Siemens a publié une mise à jour majeure de Step7, appelée SIMATIC S7-400, qui a introduit une nouvelle architecture système pour les automates Siemens. En 2001, Siemens a publié la version SIMATIC S7-200, qui est une version simplifiée de Step7 pour les petites et moyennes entreprises. En 2010, Siemens a lancé la version TIA Portal (Totally Integrated Automation Portal), qui est une plate-forme de développement intégrée pour Siemens PLC. Il intègre non seulement Step7, mais intègre également des outils de programmation tels que des onduleurs, des commandes numériques (CNC), des systèmes de surveillance, et autres équipements.

Depuis lors, TIA Portal est devenu la principale plate-forme de développement pour les automates programmables Siemens, mais Step7 reste une partie importante de cette plate-forme pour la programmation des automates des séries S7-300, S7-400 et S7-1200 [11].

### II.3 Programmation STEP 7 : [12]

Les automates programmables industriels effectuent des tâches d'automatisation traduites sous formes de programme d'application, définissant la manière dont l'automate doit superviser le système à travers une série d'instructions, le programme doit être écrit dans un langage spécifique et définir des règles d'exécution de l'automate. A cet effet, les API de la série SIMATIC sont utilisés le logiciel step7 pour programmer via la console de programmation sur le PC dans l'environnement Windows.

Une station SIMATIC représente une configuration matérielle S7 comportant un ou plusieurs modules programmables. Il existe différents types :

- SIMATIC 400 : Automate à performances extrêmes, adapté à l'exécution de programme de lourds calculs.



**Figure II.1 8 : L'API S7-400**

- .SIMATIC 300 : Automate à extensibilité modulaire.



**Figure II.2 9 : L'API S7-300**

- SIMATIC H : Automate insensible aux défaillances, il se compose de 2 CPU du même type, en cas de problème elle commute de l'une vers l'autre sans perte de données.
- SIMATIC PC : ou Station PC, représente un PC ou une station OS contenant des composants SIMATIC : des applications (WinCC, par ex.), un automate logiciel ou une carte CPU enfichée dans le PC.
- Autres stations : se sont soit des appareils d'autres fabricants ou bien des stations de SIMATIC S7 contenus dans un autre projet.
- SIMATIC S5 : liaison vers un projet S5.
- PG/PC : Outils de programmation pour contrôleurs SIMATIC, c'est une console de programmation compatible avec le milieu industriel.

#### II.4 Définition de logiciel step7 :

Step7 est le logiciel de base qui prend en charge la configuration et la programmation des systèmes d'automatisation SIMATIC. Il s'exécute à partir d'une console de programmation ou d'un PC dans un environnement Windows [12].

Il existe plusieurs versions : STEP 7 micro/ Win pour les applications S7-200 et STEP 7 pour les applications S7-300 et S7-400

Le logiciel STEP7 offre les possibilités suivantes :

- Configuration et mise en place du matériel et de communication.
- Création de gestion des projets.
- La création des programmes.
- Gestion des mnémoniques.
- Test d'installation de l'automatisation.
- Le diagnostic lors des perturbations dans l'installation.
- Documentation et archivage.

#### II.5 Les Langages de programmation :

Le STEP7 dispose de quatre langages de programmation (CONT, LIST, LOG, GRAF) pour S7-300/400 :

- **Langage liste (LIST)** : image textuelle proche du comportement interne de l'automate.
- **Langage logigramme (LOG)** : langage graphique, utilisant les symboles de l'électronique numérique (portes logiques).

- **Langage à contacts (CONT)** : suite de réseaux parcourus séquentiellement dont les entrées sont représentées par des interrupteurs et les sorties par des bobines.
- **Graph** : utilise le GRAFCET comme outil [12].

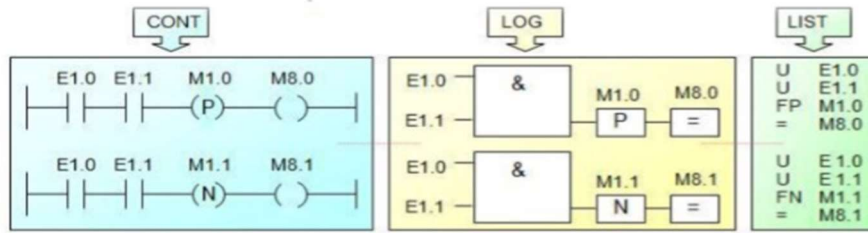


Figure II.4: Mode de présentation des langages basiques de programmation step7 [16].

**II.6 Elaboration du programme S7 (Partie Software) :**

Le dossier bloc, cité auparavant, contient les blocs que l’on doit charger dans la CPU pour réaliser la tâche d’automatisation, il englobe :

- Les blocs de code (OB, FB, SFB, FC, SFC) qui contiennent les programmes,
- Les blocs de données DB d’instance et DB globaux qui contiennent les paramètres du programme.

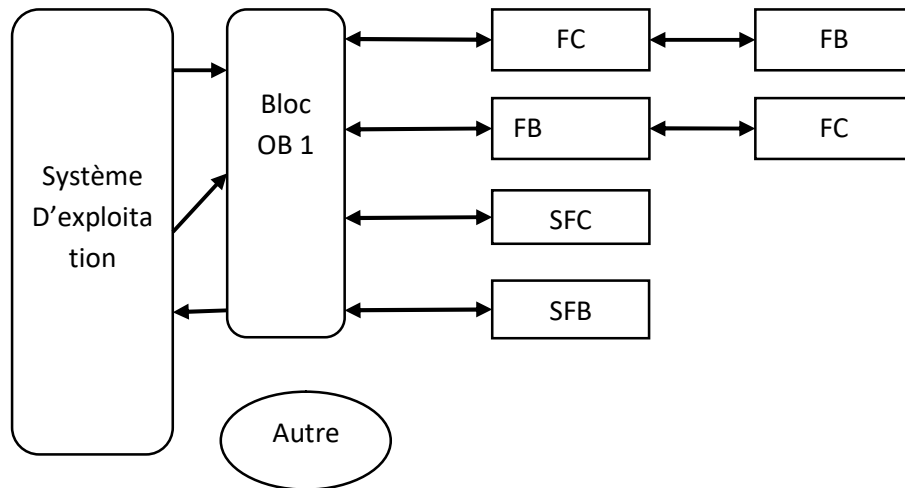


Figure II.5 : Programmation structure.

➤ **Les blocs d’organisation (OB) :**

Les OB sont appelés par le système d’exploitation, on distingue plusieurs types :

- Ceux qui gèrent le traitement de programmes cycliques
- Ceux qui sont déclenchés par un événement,
- Ceux qui gèrent le comportement à la mise en route de l’automate programmable
- Et en fin, ceux qui traitent les erreurs [13].

Le bloc OB1 est généré automatiquement lors de la création d'un projet. C'est le programme cyclique appelé par le système d'exploitation.

- **Les blocs fonctionnels (FB), (SFB) :** Le FB est un sous-programme écrit par l'utilisateur et exécuté par des blocs de code. On lui associe un bloc de données d'instance relatif à sa mémoire et contenant ses paramètres

Les SFB système sont utilisés pour des fonctions spéciales intégrées dans la CPU [13].

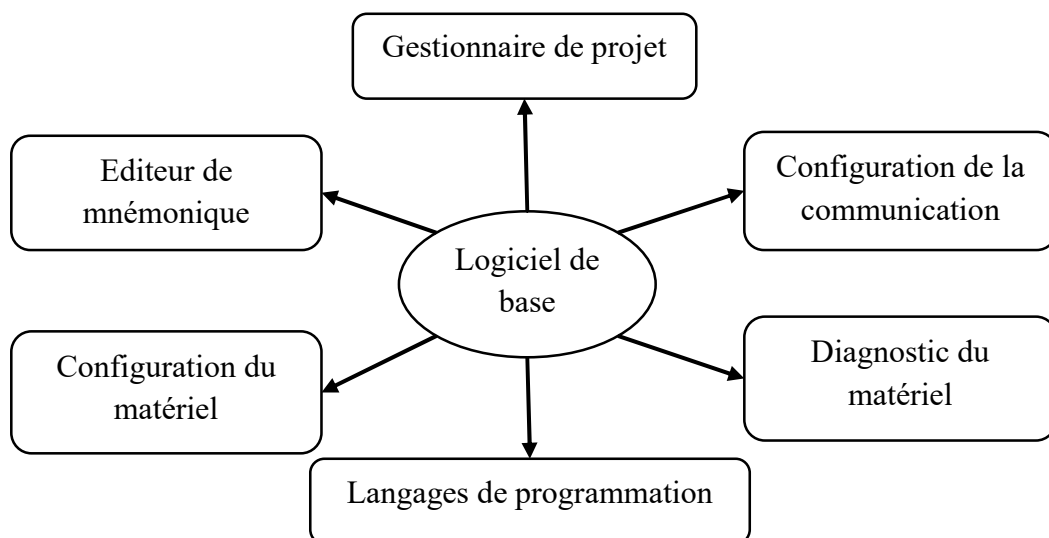
- **Les fonctions (FC), (SFC) :** La FC contient des routines pour les fonctions fréquemment utilisées. Elle est sans mémoire et sauvegarde ses variables temporaires dans la pile de données locales. Cependant elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données [13].

Les SFC sont utilisées pour des fonctions spéciales, intégrées dans la CPU S7, elle est appelée à partir du programme.

- **Bloc de données (DB) :** Les DB sont utilisés pour la mise à disposition d'espace mémoire pour les variables types données. Il existe deux types de blocs de données. Les DB globaux dans lesquels tous les OB, FB et FC peuvent lire les données enregistrées ou écrire des données et les DB d'instance qui sont affectés à un FB donné.

## II.7 Créations d'un projet Step7 :

Il existe 2 approche pour crée un projet dans le STEP7 :



**Figure II.6 :** Programmation avec logiciel STEP7.

Les étapes pour créer un projet sous le logiciel step7 sont les suivantes :

- ❖ Double cliquer sur l'icône du gestionnaire de « SIMATIC Manager », cela lancera l'assistant step7 :



Figure II.7 : Lancera l'assistant STEP7.

- ❖ Cliquez en suit sur l'icône « suivante » pour la création d'un nouveau projet :

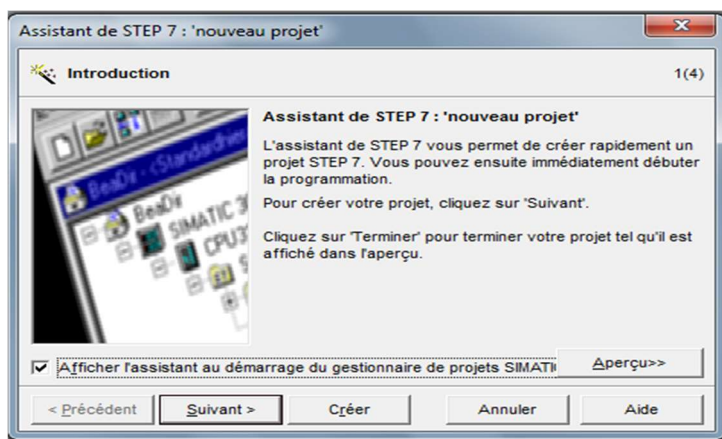


Figure II.8 : Assistante de STEP7.

- ❖ Choisir un CPU qui utilise dans le projet, après la choisir en clique sur « suivante » :

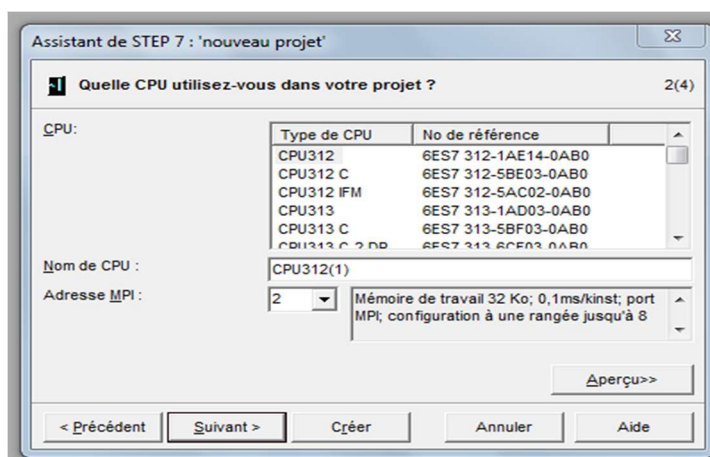
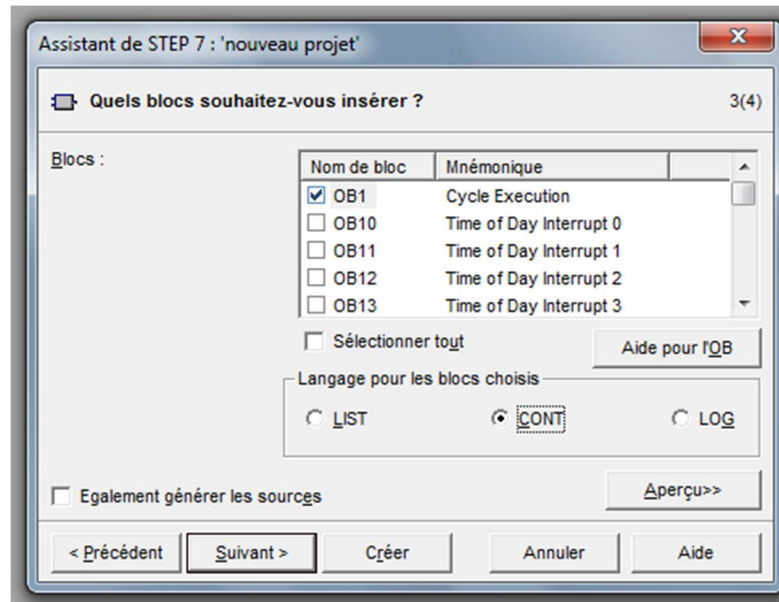


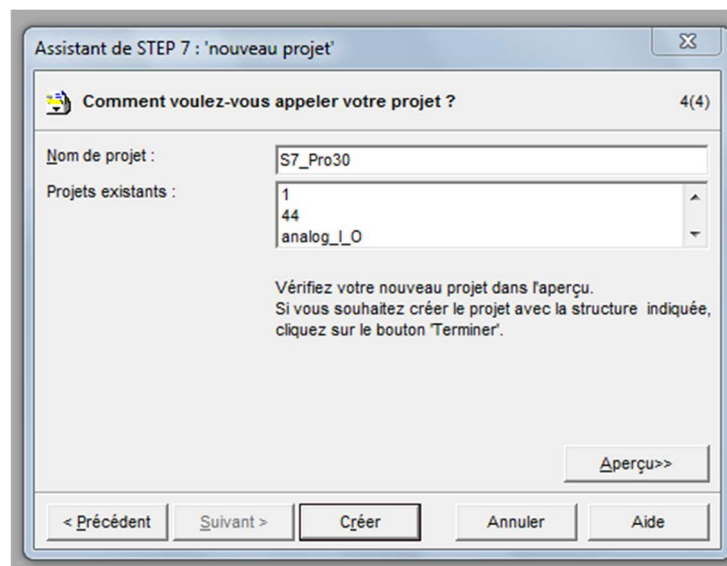
Figure II.9 : Choix de CPU.

- ❖ Après la validation de CPU, la fenêtre qui apparait permet de choisir les blocs à insérer, et choisir le langage de programmation (LIST, CONT, LOG), en choisie le langage « CONT », après en clique ensuit sur « suivant »



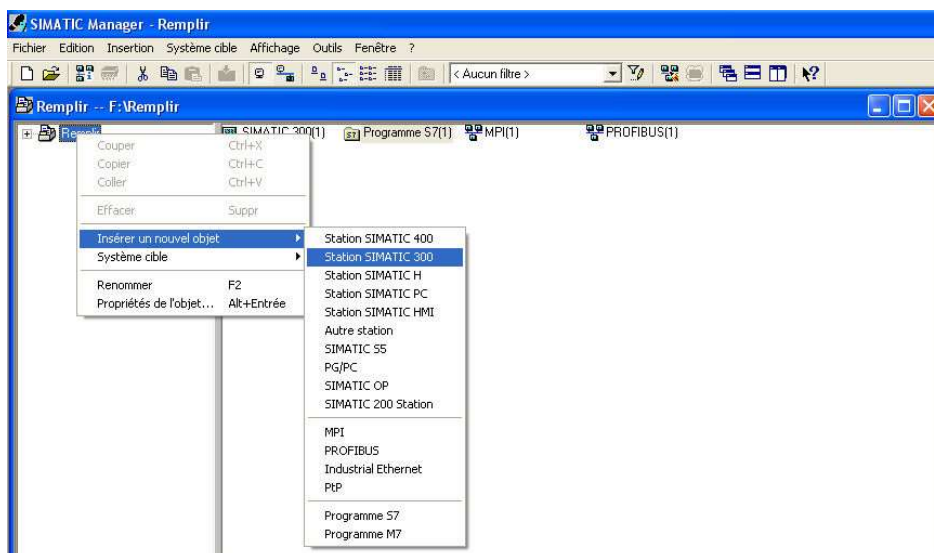
**Figure II.10 :** Choix de bloc et langage utilisé.

- ❖ Dans la boîte de dialogue « Nouveau projet », saisissez un nom pour votre projet par exemple notre projet et nommée «analog\_I\_O » :



**Figure II.11 :** Fenêtre de nommage de projet.

- ❖ Sélection le type de matériel que vous utilisez (par exemple, S7-314)



**Figure II.12 : Choix de matériel.**

Deux approches sont possibles. Soit on commence par la création du programme puis la configuration matérielle ou bien l'inverse.

- ❖ Dans la boîte de dialogue « Configuration de projet », sélectionnez les paramètres de communication avec le matériel « adresse IP, type de connexion... »
  - C'est une étape importante, qui correspond à l'agencement des châssis, des modules et de la périphérie décentralisée. Les modules sont fournis avec des paramètres définis par défaut en usine. Une configuration matérielle est nécessaire pour :
    - Modifier les paramètres ou les adresses pré-réglés d'un module.
    - configurer les liaisons de communication.

Le choix du matériel SIMATIC S300 avec une CPU314 nous conduit à introduire hiérarchisée suivante :

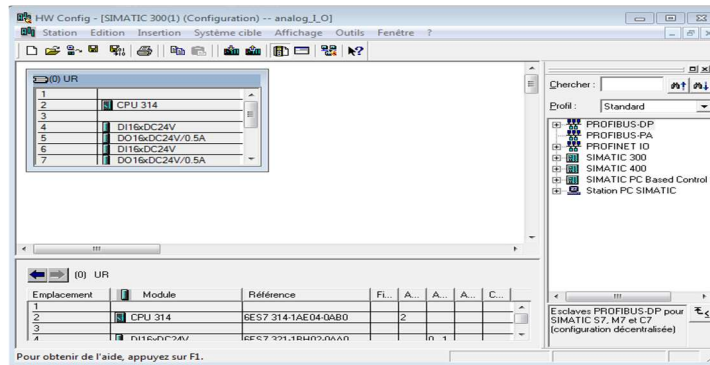


Figure II.13 : Fenêtre de configuration du matériel.

❖ Chargements du tableau mnémorique :

La mnémonique est le donné par l'utilisateur et qui peut remplacer une variable ou un bloc de programmation. La table de mnémonique c'est une table qui permet d'affecter des noms à des adresses données globale accessibles à partir de tous les blocs.

Insérer nouvelle objet  $\Longrightarrow$  Table des mnémoniques

Etat	Mnémonique /	Opérande	Type de do	Commentaire
1	CONT_C	FB 41	FB 41	Continuous Control
2	G7_STD_3	FC 72	FC 72	
3	SCALE	FC 105	FC 105	Scaling Values
4	TIME_TCK	SFC 64	SFC 64	Read the System Time
5	UNSCALE	FC 106	FC 106	Unscaling Values
6	VAT_1	VAT 1		
7	VAT_2	VAT 2		
8				

Figure II.14 : Tableau de mnémonique.

❖ Les blocs utilisent

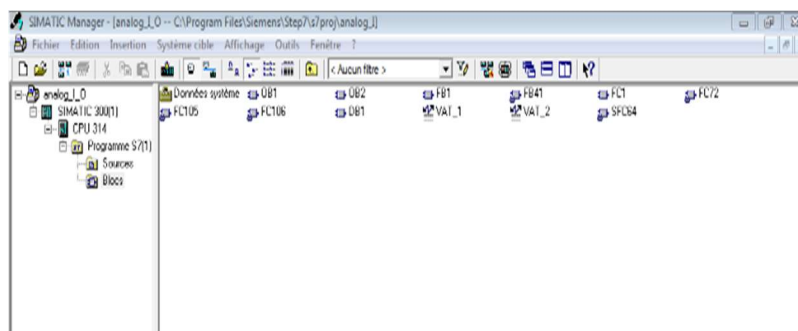


Figure II.15 : Fenêtre des blocs software

❖ La fenêtre de création de projet sur le bloc OB1 :

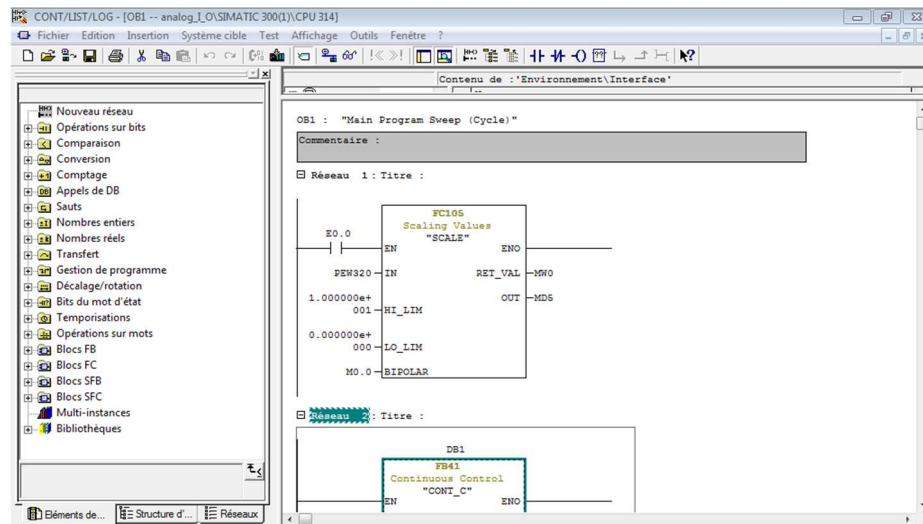


Figure II.16 : Fenêtre de bloc OB1

On commence par le choix du châssis selon la station choisie auparavant, Pour la station SIMATIC S300, on aura le châssis « RACK-300 » qui comprend un rail profilé.

Sur ce profilé, La « CPU 314 » est impérativement mise à l’emplacement n°2.

L’emplacement n°3 est réservé comme adresse logique pour un coupleur dans une configuration multi châssis.

A partir de l’emplacement 4, il est possible de monter au choix jusqu'à 8 modules de signaux(SM), processeurs de communication (CP) ou modules fonctionnels (FM). Après cela il ne nous reste qu’à enregistrer et compiler.

La configuration matérielle étant terminée, un dossier « Programme S7 » est automatiquement inséré dans le projet, comme indiqué dans la figure suivante :

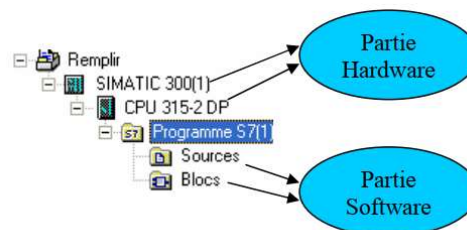


Figure II.17 : Création de programme S7.

## II.8 le logiciel S7-PLCSIM :

L'application de simulation de module S7-PLCSIM nous permet d'exécuter et de tester nos programmes dans l'automate. Etant donné que la simulation est complètement dans le step7, il n'est pas nécessaire d'établir un module S7.

L'objectif de ce logiciel est de tester et de valider les programmes step7 pour les API S7-300 et S7-400.

### II.8.1 Présentation de S7-PLCSIM :

Le S7-PLCSIM est un outil puissant et polyvalent pour la simulation de programmation de contrôleurs Siemens S7, spécialement conçu pour les applications d'automatisation industrielle. Il permet aux développeurs de tester et déboguer leurs programmes sans avoir besoin d'un matériel réel.

### II.8.2 Mise en route du logiciel S7-PLCSIM :

Pour mettre en route le logiciel S7-PLCSIM, suivez les étapes suivantes :




- Dans SIMATIC MANAGER active le logiciel de simulation en cliquant sur l'icône : 
- Dans le programme et les blocs en cliquant sur l'icône : 
- Pour lancer la visualisation en cliquant sur l'icône : 
- Mettre le CPU en mode RUN ou RUN-P :



Figure II.1810 : Mode de simulation.

- Pour sauvegarder la version actuelle de la simulation, cliquez sur : 

## II.9 Exemple d'application de Création et d'édition d'un projet S7 :

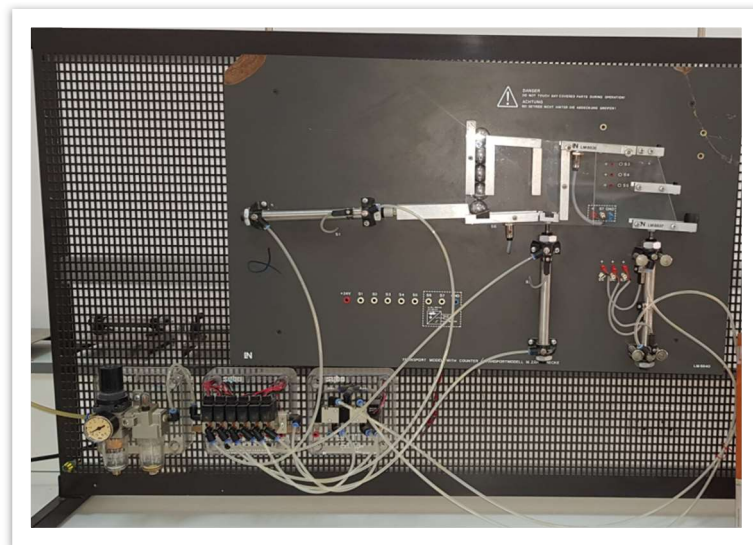
### «Transfert des pièces dans un chaîne par automate programmable API »

#### II.9.1 L'objectif de cette application :

Cette application a pour but de montrer les possibilités qu'offrent les automates programmables pour la commande des systèmes automatiques.

La figure suivante représente une station de transfert de pièces trois vérins pneumatiques : la station est composée par :

- trois vérins A, B et C pour pousser la pièce comme le montre la figure5 suivante.
- cinq capteur (S1, S3, S4, S5, S6 et S7) pour contrôler les sorties et les recules des tiges des vérins.
- Un capteur S1 pour la détection de présence de pièce.
- Un bouton poussoir STOP.



**Figure II.19 :** Station de Transfert des pièces.

Pour bien illustrer les démarches de conception d'un programme dont on a souligné précédemment les étapes, on prend comme exemple d'application l'automatisation d'une chaîne de transfert de pièces, avec le cahier de charge suivant :

### II.9.2 Cahier de charge :

Programmer la gestion des modes fonctionnement sur une installation pneumatique de transfert de pièces d'après les indications suivantes :

- L'installation est mise sous tension avec l'entrée E0.0, un témoin s'allume à la sortie A4.0, et elle est mise hors tension à l'aide de l'entrée E0.1.
- Lorsque l'installation est sous tension, le mode de fonctionnement est sélectionné ainsi :
- En Cas de changement du mode de fonctionnement ou de mise hors tension de l'installation, le mode sélectionné préalablement est désactivé.
- Le nombre de pièce sera compté et affiché pour l'utilisateur, grâce à un afficheur sept segments muni d'un décodeur BCD, et adressé sur AW 12.

### II.9.3 Fonctionnement :

Initialement le système est au repos :

- Les trois vérins sont reculés.

A l'impulsion sur un bouton poussoir START le système effectue le cycle suivant :

Le vérin A poussé le poste de chargement jusqu'à l'action sur S5 (niveau 3) les pièces se déplacent automatiquement un par un vers le niveau 3 c.-à-d. la tige du vérin B pousse la pièce jusqu'à l'action sur S6, simultanément la tige du vérin B se recule et la tige du vérin C pousse la pièce jusqu'à l'action sur S7, lorsque le nombre de pièces programmés est atteint (compteur<sub>3</sub>=3). Une fois les pièces est translitées le vérin A revient le poste de chargement jusqu'à l'action sur S3 (niveau1), les pièces se déplacent automatiquement un par un vers le niveau1. C.-à-d. la tige du vérin B pousse la pièce jusqu'à l'action sur S6 simultanément la tige du vérin B se recule et la tige du vérin C pousse la pièce jusqu'à l'action sur S7, lorsque le nombre de pièces programmés est atteint (compteur<sub>1</sub>=1) ou que l'on a demandé d'arrêt de cycle, le bouton poussoir STOP permet d'arrêter le système par remise à zéro de tout le système. Si le nombre de pièces programmés ne pas atteint (compteur  $\neq$  nombre de cycle) et si on n'a pas demandé d'arrêt de cycle alors le système recommence le cycle chaque présence d'une pièce devant le capteur S1.

#### II.9. 4. L'adressage de notre GRAFECET :

Etape	Capteur	Pré actionneur
X0 : M0.0	S1 : I0.0	$dA^+$ : Q0.0
X1 : M0.1	S2 : I0.1	$dA^-$ : Q0.1
X2 : M0.2	S3 : I0.2	$dB^+$ : Q0.2
X3 : M0.3	S4 : I0.3	$dB^-$ : Q0.3
X4 : M0.4	S5 : I0.4	$dC^+$ : Q0.4
X5 : M0.5	S6 : I0.5	$dC^-$ : Q0.5
X6 : M0.6	S7 : I 0.6	T37 : 5sec
X7 : M0.7	Dcy : I0.7	C3 : 3 fois
X8 : M1.0	Rest_compteur : I1.1	C2 : 2fois
Auto_Man : m2.0	March_Auto : I1.3	
RAZ :M2.1	Semi_automatique :I 1.4	

Tableau II.1 : L'adressage absolu des entrées/sorties de la première application

Il inclue un automate S300, avec une CPU314 et des modules E/S qui seront cités plus tard dans la configuration du matériel, et comporte aussi une interface E/S analogique et numérique pour ses modules, ainsi qu'un afficheur sept segments.

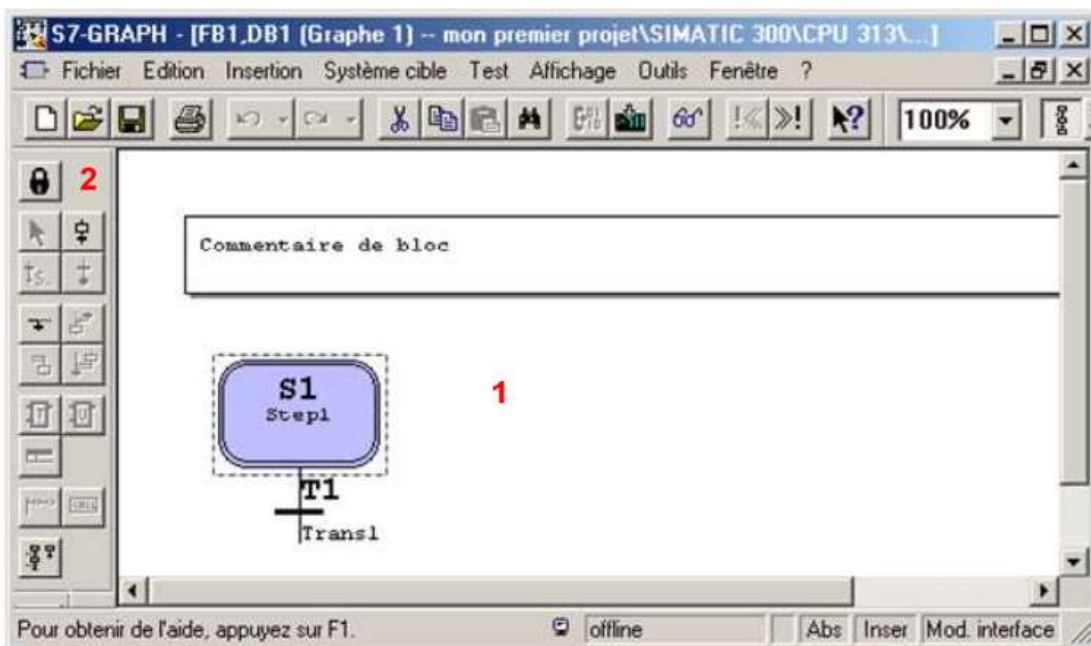


Figure II.20 : Edition d'un programme fonctionnel avec S7-GRAPH.

On remarque deux parties, la première est la zone d'édition, la deuxième inclus les éléments à ajouter (des transitions, des branchements, des étapes...).

Pour ajouter un élément à la zone d'édition, il suffit de cliquer sur l'emplacement souhaité puis sur l'élément.

Notre exemple de transfert de pièces comprend neuf étapes et onze transitions. On commence par les créer.

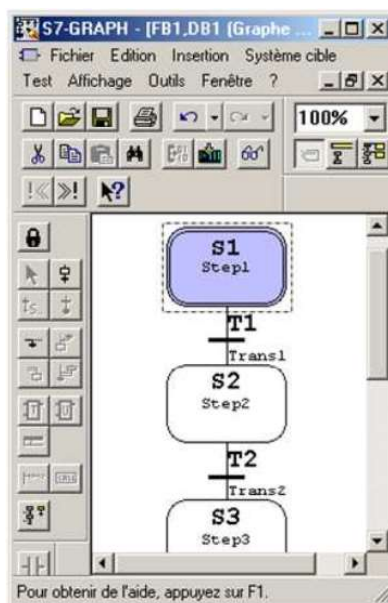


Figure II.21 : Edition du GRAPH.

Ensuite pour l'édition des actions et des transitions on change l'affichage de 'Graph' à 'Étape'. Pour ajouter une action, cliquer par le bouton droit de la souris sur la colonne à droite et choisissez 'insérer un nouvel élément' puis sur 'action'. Il existe différent type d'action, le tableau suivant présente les plus utilisées :

Opération	Opérande	Signification
N	A, E, M ou D	Tant que l'étape est active l'opérande est à 1.
S	A, E, M ou D	SET : Dès que l'étape est active, l'opérande est mis à 1.
R	A, E, M ou D	RESET : Dès que l'étape est active, l'opérande est mis à 0.
D	A, E, M ou D	Delay : n secondes après l'activation de l'étape, l'opérande se mis à 1 pour la durée de l'activation.
L	A, E, M ou D	Impulsion : quand l'étape est active, l'opérande se mis à 1 pendant n secondes
CALL	FB, FC, SFB ou SFC	Appel de bloc : tant que l'étape est active, le bloc spécifié est appelé.

Tableau II.2 :Les actions dans le S7-GRAPH.

Pour les transitions, elles peuvent être présentées en langage CONT ou LOG, cliquer sur la colonne à gauche et insérer des éléments (contacts...) depuis le catalogue.

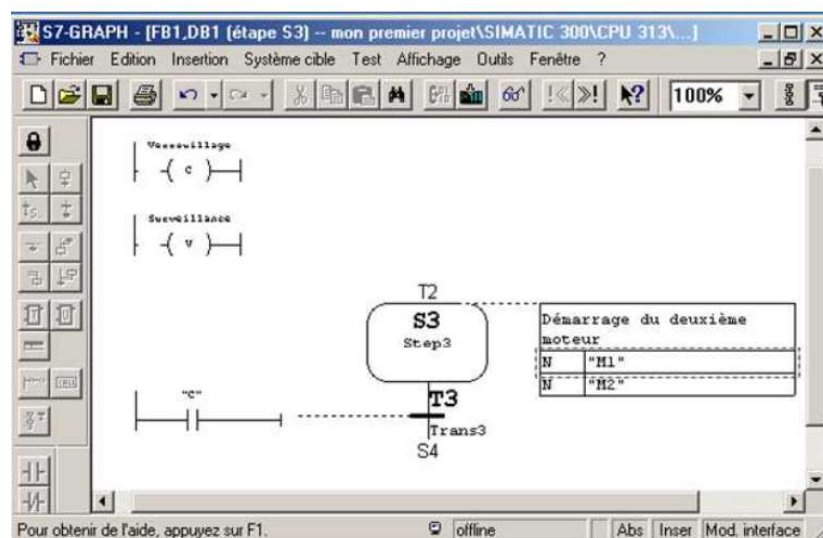


Figure II.22 : Edition des actions & transitions.

### II.9.5. Simulation de modules :

L'application de simulation de modules S7-PLCSIM permet l'exécution et le teste du programme utilisateur destinés aux CPU S7-300 et aux CPU S7-400, ainsi qu'à Win CC. La simulation étant complètement réalisée au sein du logiciel STEP 7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque. Lorsque S7-PLCSIM s'exécute, toute nouvelle liaison est automatiquement dirigée vers la CPU de simulation. S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées).

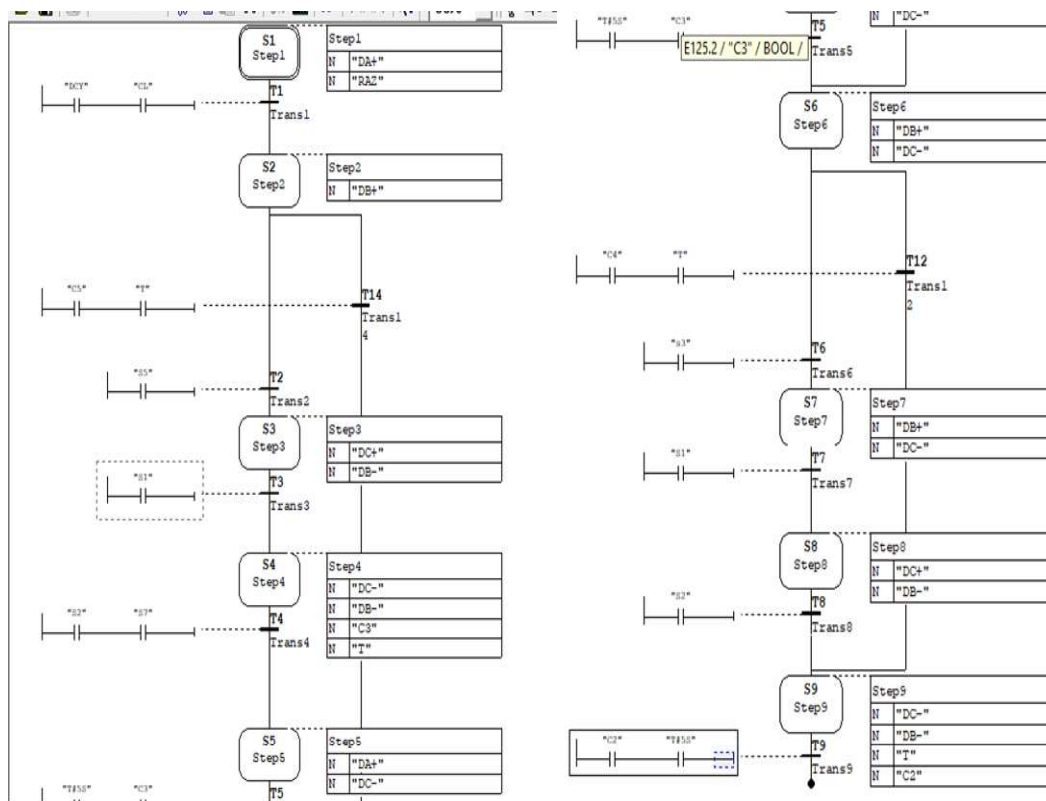


Figure II.23 : Le programme étant écrit, on doit le charger dans le CPU.

### II.9.5. 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).
- 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).

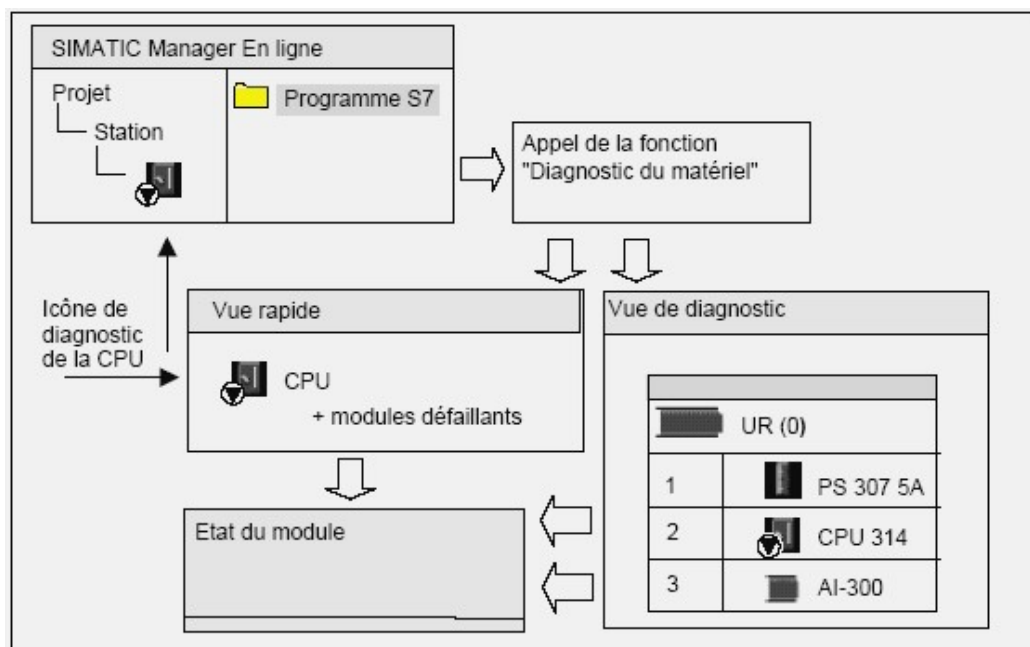
Si la syntaxe est correcte, le bloc est ensuite compilé en code machine, enregistré et chargé par la commande Système cible > Charger, ou en sélectionnant tous les blocs et charger par la fonction de chargement dans la barre d'outils du SIMATIC Manager.

**II.9.6. Surveillance du fonctionnement et diagnostic du matériel :**

Pour le test du bon fonctionnement du programme, le logiciel optionnel de simulation permet d'exécuter et de tester le programme dans un système d'automatisation qu'on simule dans l'ordinateur ou dans la console de programmation en y accédant par la fonction dans la barre d'outils.

Pour la recherche d'erreurs, des icônes de diagnostic permettent de déceler des défauts sur un module et indiquent l'état de ce dernier. Pour une CPU, par exemple, son état de fonctionnement (RUN, STOP ou RUN-P).

Les icônes de diagnostic s'affichent dans la vue en ligne de la fenêtre du projet, dans la vue rapide (présélection) ou encore dans la vue de diagnostic lors d'un appel de la fonction "Diagnostic du matériel".



**Figure II.24 :** Diagnostic du matériel [13].

**II.10 Conclusion :**

La plupart des grands constructeurs d'automates programmables, fournissent des logiciels de configuration et de programmation munis des langages SFC, LD, FBD, ST et IL. Le choix d'un langage s'appuie sur la complexité de l'application et de la tâche de commande. Il est préférable d'utiliser les langages graphiques (SFC, LD et FBD) pour la réalisation des programmes de commande séquentiels. Le SFC est la réalisation directe d'un GRAFCET de commande, les langages LD et FBD sont plus utiles pour les opérations combinatoires sur bits ou mots.

***Chapitre III :***

***Le progiciel de conception des  
interfaces Homme-Machine***

### III.1 Introduction :

Suite à l'automatisation industrielle, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface d'interaction Homme/Machine, flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facilité que l'écran comporte des images avec des synoptiques, des graphes, des paragraphes...etc.

Le progiciel Win CC flexible, reste le plus approprié, en termes de simplicité, pour la conception ce type d'interface de commande/diagnostic.

### III.2 Progiciel de conception et configuration d'interface Win CC flexible

#### III.2.1 Présentation Générale

Win CC flexible, est un progiciel partagé dans l'environnement STEP7, et propose pour la configuration de divers pupitres opérateurs, une famille de systèmes d'ingénierie évolutifs adaptés aux tâches de configuration. Créer l'interface graphique et les variables, c'est pouvoir lire les valeurs du processus via l'automate, les afficher pour que l'opérateur puisse les interpréter, et ajuster, éventuellement, le processus, toujours via l'automate (Figure III.1).

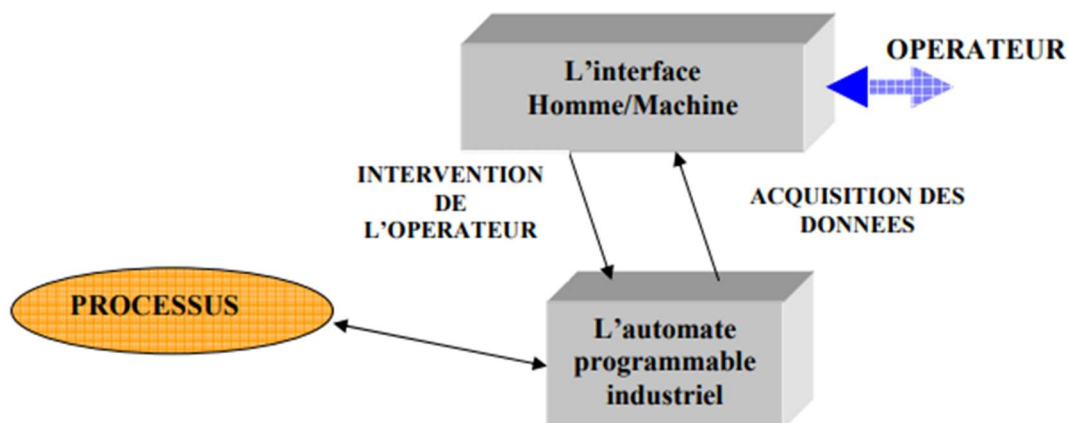
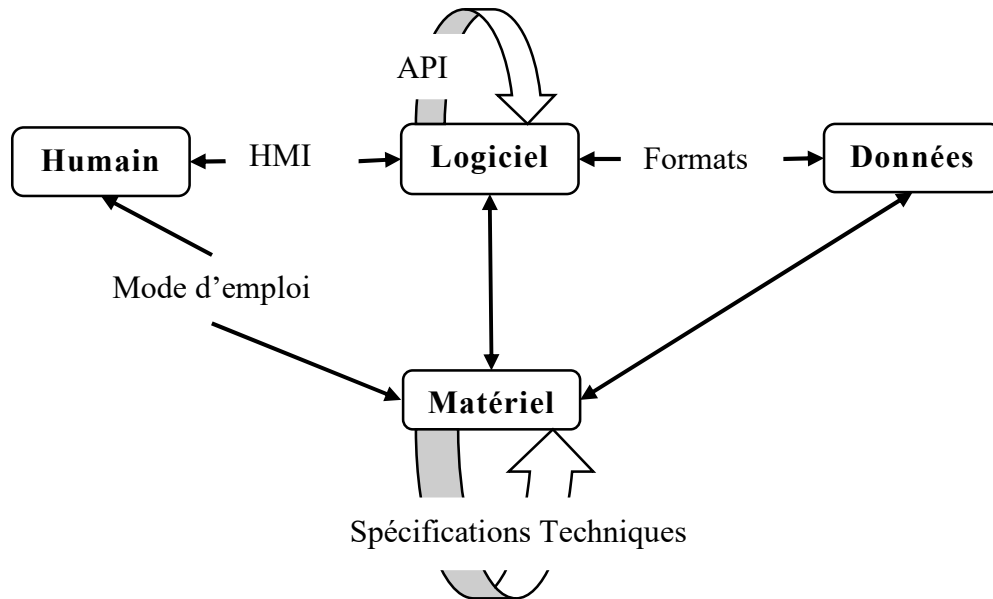


Figure III.1 : L'interface Homme/Machine dans un processus automatisé



**Figure III.2 :** Interface Homme-Machine, API pour programmation d'application.

### III.3.La supervision :

La supervision est un moyen de surveiller l'état de fonctionnement d'un processus. Elle répond à des besoins nécessitant une puissance de traitement importante et assure la communication entre les équipements d'automatisme et les outils informatique. Elle coordonne également le fonctionnement d'un ensemble de machines enchaînées, et assiste l'opérateur dans les opérations de diagnostic et de maintenance.

Les systèmes de supervision industrielle utilisent des capteurs et des actionneurs pour mesurer les paramètres des processus, tels que la température, la pression, le débit et la vitesse de rotation, et pour contrôler les équipements de production tels que les moteurs, les pompes et les vannes.

Les données nécessaires sont ensuite utilisées par des logiciels de supervision pour les afficher sous forme de graphiques, de tableaux de bord et d'alertes [11][15].

#### III.3.1 Architecteur d'UN réseau de supervision :

Un réseau de supervision est généralement composé de plusieurs éléments qui travaillent ensemble pour collecter, traiter et afficher les données de supervision.

Voici les éléments clés d'un réseau de supervision typique :

- **Capteurs :** Ils sont utilisés pour collecter des données à partir de différents systèmes et équipements, tels que des températures, des pressions, des niveaux,

etc..., ces capteurs envoient les données collectées à une unité centrale pour traitement.

- Unité centrale : Elle est chargée de collecter [16].

### **III.3.2. Constitution d'un système de supervision :**

La constitution d'un système de supervision peut varier en fonction des objectifs spécifiques et des exigences techniques de chaque situation. Cependant, voici les étapes générales que vous pouvez suivre pour constituer un système de supervision :

- ✓ Définir les objectifs et les besoins
- ✓ Identifier les capteurs et les dispositifs de surveillance
- ✓ Installer et configurer les capteurs et les dispositifs de surveillance
- ✓ Collecter et stocker les données de surveillance.
- ✓ Analyser les données de surveillance.
- ✓ Agir en fonction des résultats de l'analyse.

En suivant ces étapes, vous pouvez constituer un système de supervision efficace qui vous permet de surveiller les paramètres clés et de prendre des mesures en fonction des résultats de l'analyse des données de surveillance.

### **III.4 Utilisation de SIMATIC Win CC Flexible :**

Grace à Win CC flexible, les ingénieurs peuvent créer des interfaces homme-machine conviviales pour surveiller les machines. Win CC flexible, réunit les avantages suivants :

- ✓ Simplicité.
- ✓ Flexibilité et évolutivité.
- ✓ Diagnostic et analyse avancés.
- ✓ Intégrations avec les systèmes automatisés.

#### **III.4.1. Win CC flexible Runtime:**

Le Principe du Win CC flexible RUNTIME, l'opérateur peut réaliser le contrôle-commande des processus industriels. Voici comment les tâches spécifiques que vous décrit sont exécutées :

- Communication avec les alarmes.
- Affichage des vues à l'écran.
- Commande de processus.

**III4.2 La liaison Win CC avec PLC Sim :**

Le réseau MPI (Multipoint interface) est un protocole de communication utilisé dans les systèmes d'automatisation industriel Siemens SIMATIC S7-300. Il permet la communication entre les contrôleurs S7-300 et les périphériques déportés, tels que les PC de supervision ou modules d'E/S déportés.

Dans notre projet nous allons utiliser un type de communication le type est :

- La communication SIMATIC 300-PC déporté via le MPI.

Le réseau MPI utilisé une topologie en chaine, ou chaque périphérique est connecté à un autre via un câble MPI. Le contrôleur SIMATIC S7-300 agit généralement comme le maitre du réseau, contrôlant la communication avec les périphériques déportés.

La communication via le MPI permet d'échanger des données entre le contrôleur et le PC déporté. Cela peut inclure des informations de supervision, des commandes de contrôle, des valeurs de capteurs, des alarmes, etc. Le protocole MPI offre une communication faible et rapide, adaptée aux exigences de l'automatisation industrielle [17].

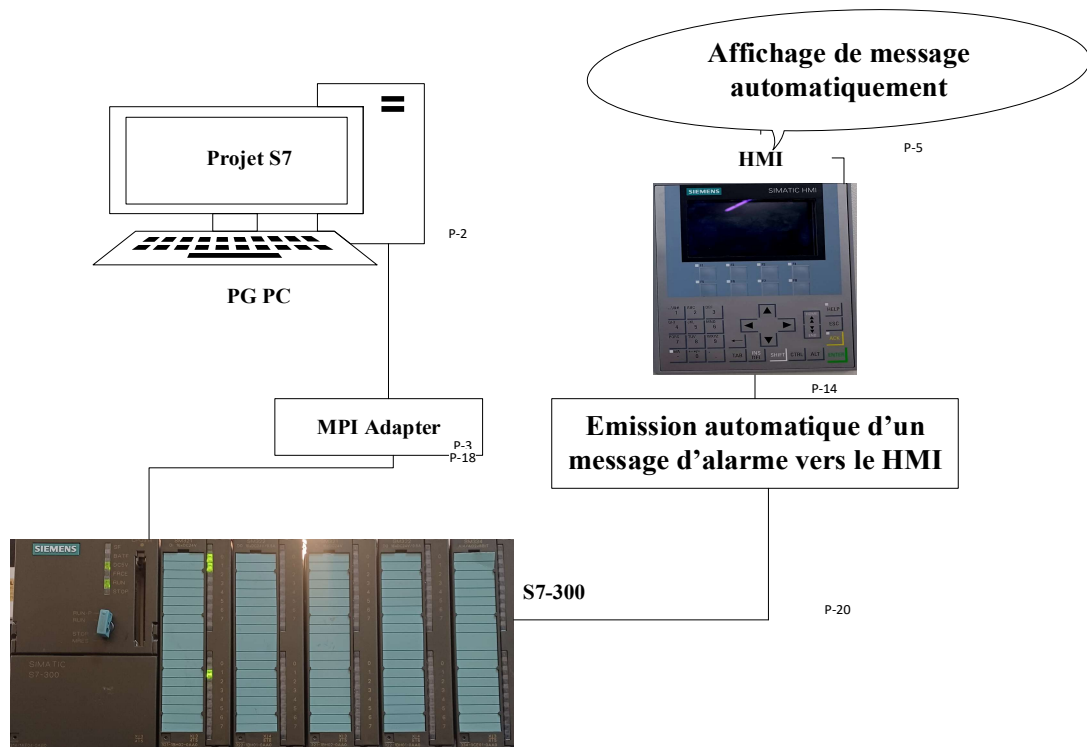


Figure III.3 : Schéma de la liaison et pelotage (MPI)

### III.5 Principalement des vues : [18]

Dans le Win CC Flexible, l'éditeur de vues offre différents outils et barres pour créer des interfaces de contrôle-commande. Sont représentés comme suite :

- **Barre de menu** : la barre de menu contient les options de menu principales pour l'éditeur de vues, telles que fichier, Edition, Affichage, Insertion, Format, etc. Elle permet d'accéder aux différentes fonctionnalités et commandes pour la création et la modification des vues.
- **Barres d'outils** : la barre d'outils affiche tous les outils dont un programmeur a besoin.
- **Zone de travail** : la zone de travail est utilisée pour configurer la vue afin qu'elle soit la plus compréhensible pour l'utilisateur, et très facile à manipuler et à consulter les résultats.
- **Boîte à outils** : une fenêtre d'outils propose un choix d'objets simples ou complexes pouvant être insérés dans une vue, tels que des objets graphiques et des éléments de commande.

- **Fenêtre de propriétés** : le contenu de la fenêtre de propriétés dépend de la sélection actuelle dans la zone de travail, lorsqu'un objet est sélectionné on peut modifier les propriétés de l'objet dans la fenêtre des propriétés.

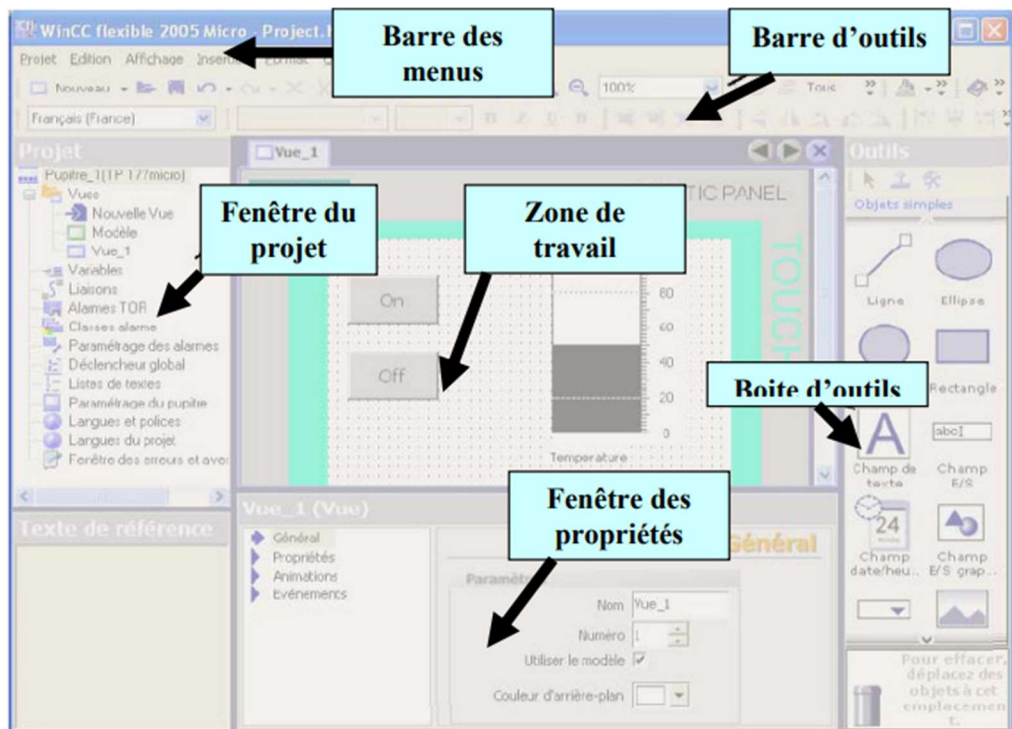
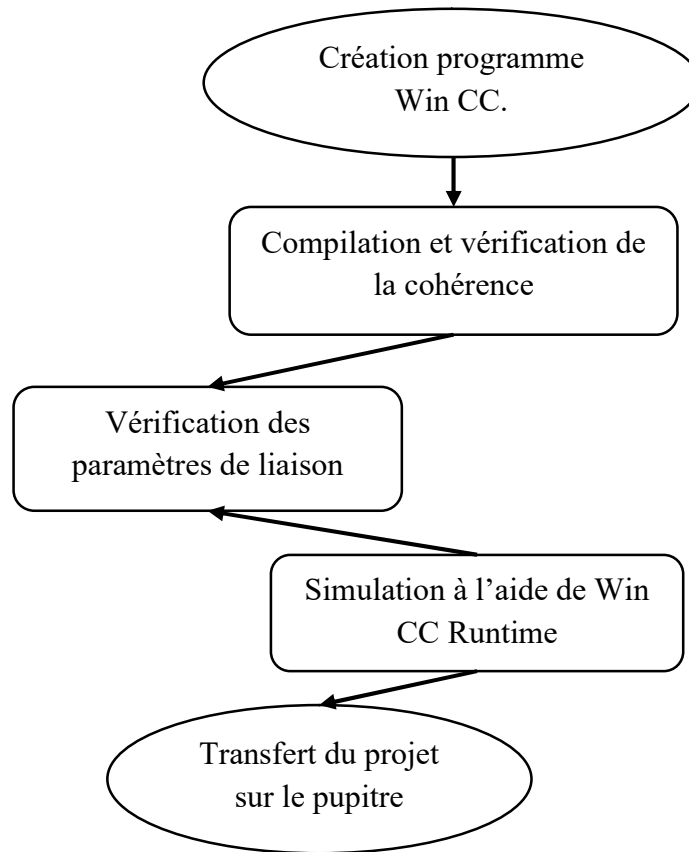


Figure III.4 : Vue d'ensemble du logiciel Win CC flexible

### III.6 Élément du logiciel Win CC flexible

L'utilisation de ce logiciel pour la conception d'une interface, passe par plusieurs étapes résumées dans la Figure III.5 :



**Figure III.5 :** Etapes pour la conception d'une interface via Win CC flexible

Pour créer un projet dans le WinCC, suivez les étapes suivantes :

- Lancez le logiciel Win CC en cliquant sur l'icône correspondante sur le bureau :



**Figure III.6 :** l'assistante de Win CC flexible

- Dans la fenêtre d'accueil, cliquez sur «Crée un nouveau projet ».

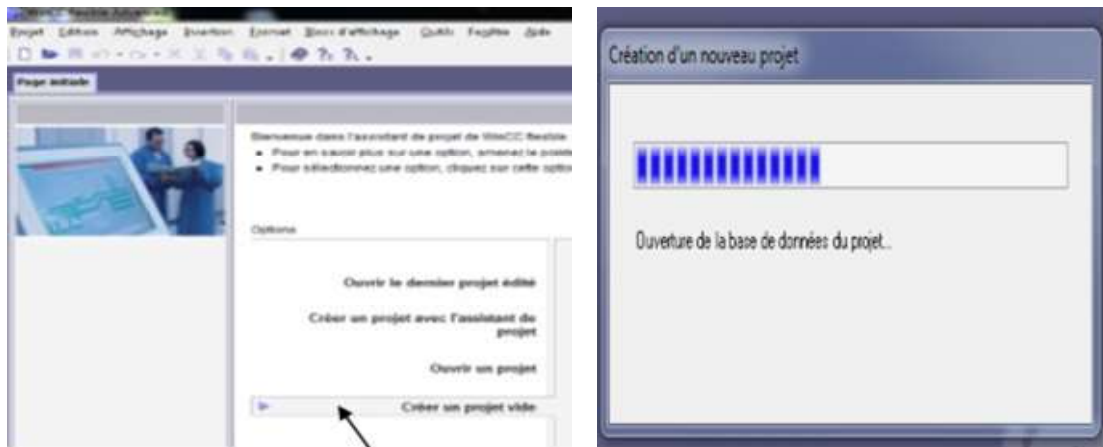


Figure III .7 : Création d'un nouveau projet

- **Sélection du pupitre :**

Sélectionner sur panel choisi

Confirmer la sélection avec **OK**

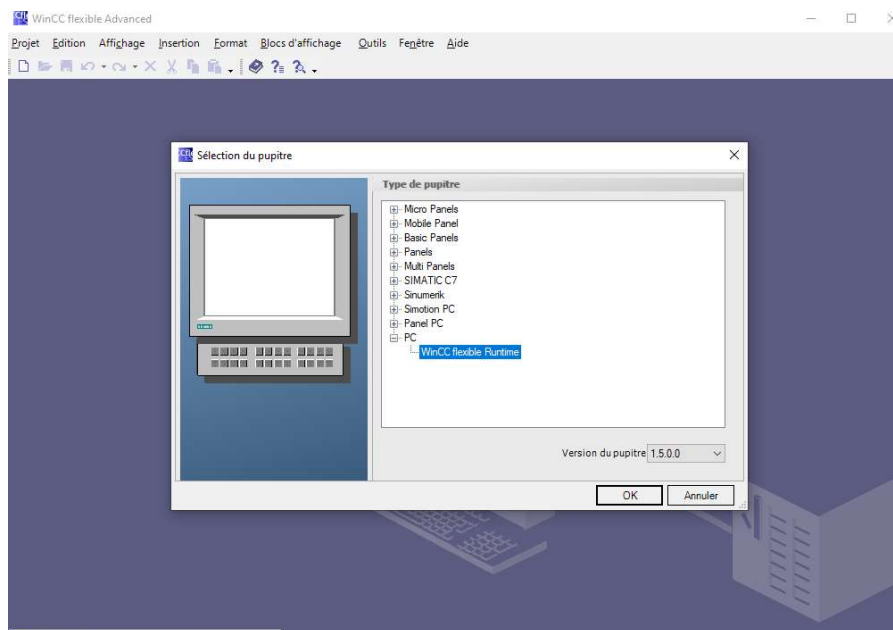


Figure III.8 : Sélection du pupitre

### III.6.1 Intégration de Win cc avec step7 :

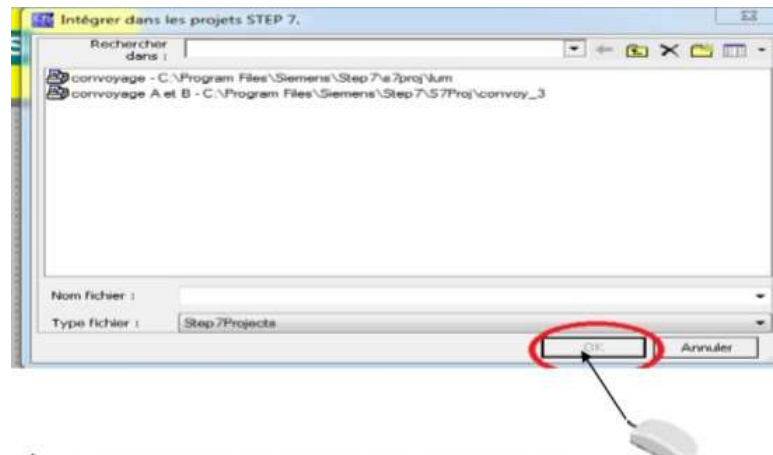
Win CC flexible offre une intégration particulièrement poussée avec les composants de la famille de produits SIMATIC. L'intégration de step7 dans l'interface de configuration réduit la fréquence des erreurs et réduit les tâches de configuration nécessaires. Saisissez directement la table des mnémoniques et les paramètres de communication de step7 lors de la configuration :

- La table mnémonique de step7 contient la définition des points de données. Nous avons paramétré le programme de contrôle lors de sa création.
- Les paramètres de communication incluent l'adresse et le protocole de commande.

**Cliquer sur aperçu puis :**

- Sélection **Projet Intégration dans projet step7**

Confirmer les sélections avec **OK**



**Figure III.9 :** l'intégration dans le step7

### III.6.2 Création des Vues :

Les vues sont l'élément principal du projet. Ils sont utilisés pour contrôler et surveiller les installations. Ces vues contiennent plusieurs éléments, tels que des objets simples. Les objets simples incluent les objets graphiques de base tels que les « lignes » et les éléments de commande de base, ainsi que les « champs d'E\S » ou les « boutons ».

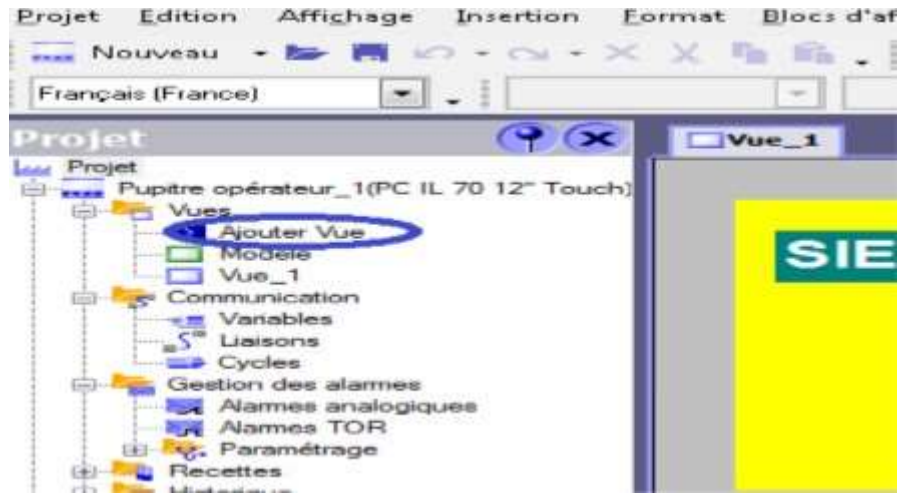


Figure III.10 :fenêtre de création des vues

➤ Choix d'objet :

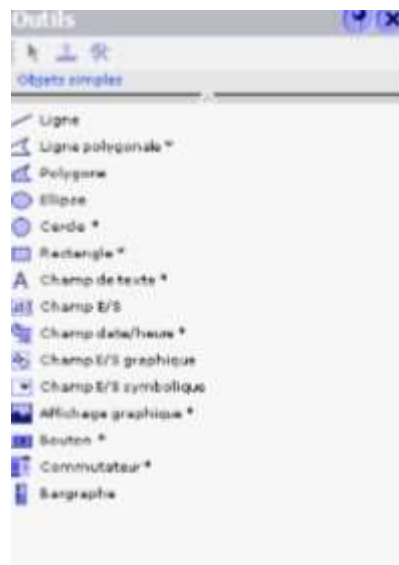


Figure III.11 : Sélection les objets

Dans cette fenêtre-il possible de choisi 1 sur 5 : objets simple, objets complexes, mes contrôles, graphique, bibliothèque.

### III.6.3 Utilisation de variable :

On distingue deux types de variables, les variables externes et les variables internes. Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate p. ex. Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre opérateur que de l'automate. Les variables internes ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre. Le pupitre opérateur en question peut donc accéder en lecture et en écriture aux variables internes. Les variables internes sont créées p. ex. pour exécuter des calculs locaux. La déclaration des variables s'effectue dans l'éditeur "Variables". Lors de leur déclaration, une configuration de base est attribuée aux variables. L'éditeur "Variables" vous permet d'adapter la configuration des variables aux besoins de votre projet ; L'éditeur s'ouvre systématiquement par ajout de variable.

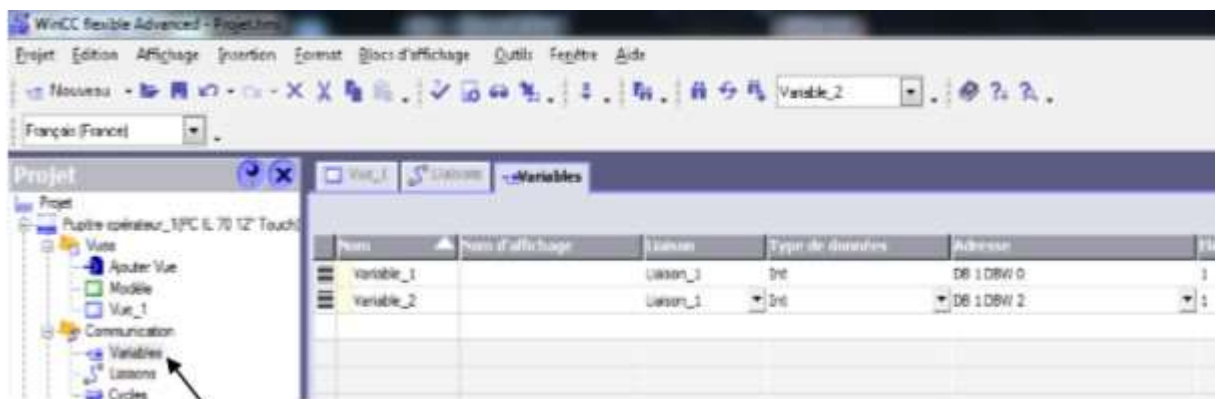


Figure III.12 : Edition de variable dans le Win CC flexible

Chaque variable peut être configurée selon le besoin et cela en accédant à sa fenêtre de priorité après l'avoir sectionnée.

### III.6.4 Etablissement de la liaison automate HMI :

La liaison est établie en choisissant le protocole de communication qui est dans notre cas MPI, en respectant les adresses de chaque élément et la vitesse de transmission.

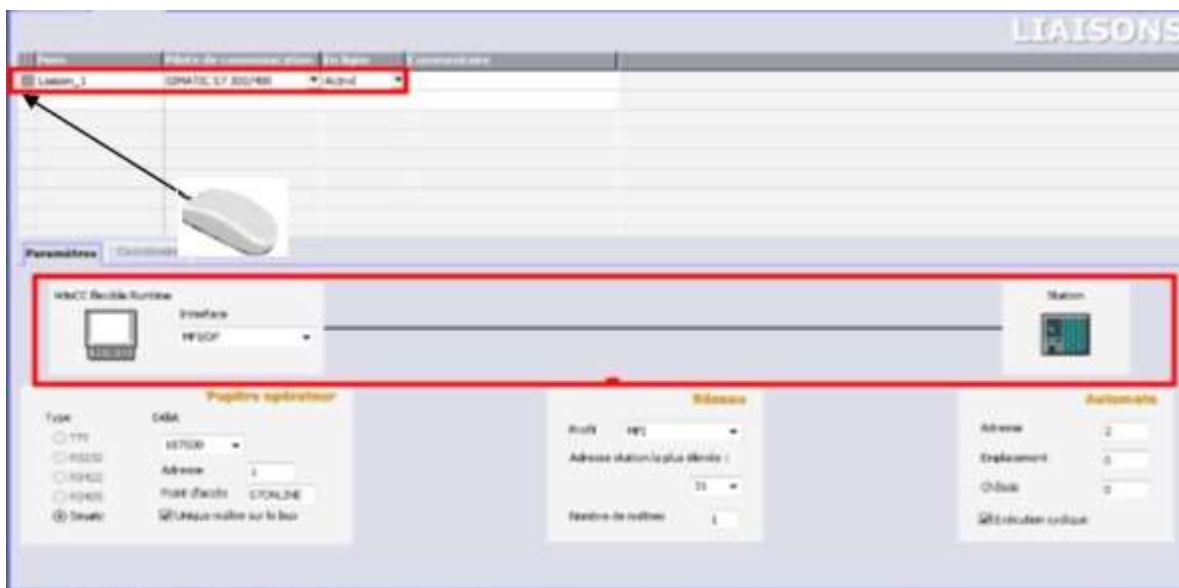


Figure III.13 : éditeur de la liaison

### III.6.5 Compilation et Simulation :

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, de contrôler la cohérence et de chercher les erreurs, à l'aide de la commande sur la barre du menu « contrôle de la cohérence ». Après le contrôle de cohérence, le système crée un fichier de projet compilé. La simulation permet de détecter des erreurs logiques de configuration, par exemple, des valeurs limites incorrectes, et cela à l'aide du simulateur Runtime par la commande « démarrer le système Runtime du simulateur », comme montre dans la figure suivante :

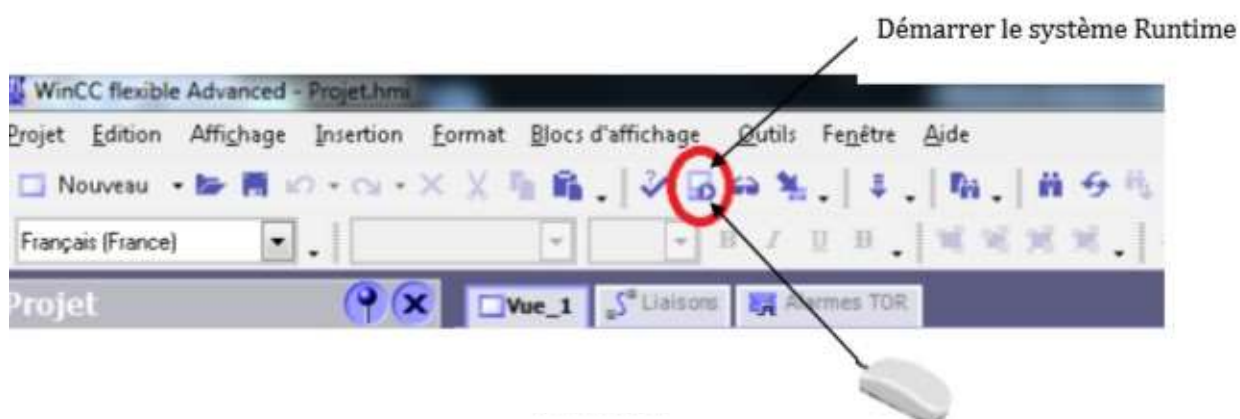


Figure III.14 :Démarrage de système Runtime

Les éléments représentatifs de la station sont utilisés à partir de la bibliothèque du Win CC flexible, et chaque composant est affecté à une variable, une adresse, une représentation et un événement qui lui est assigné.

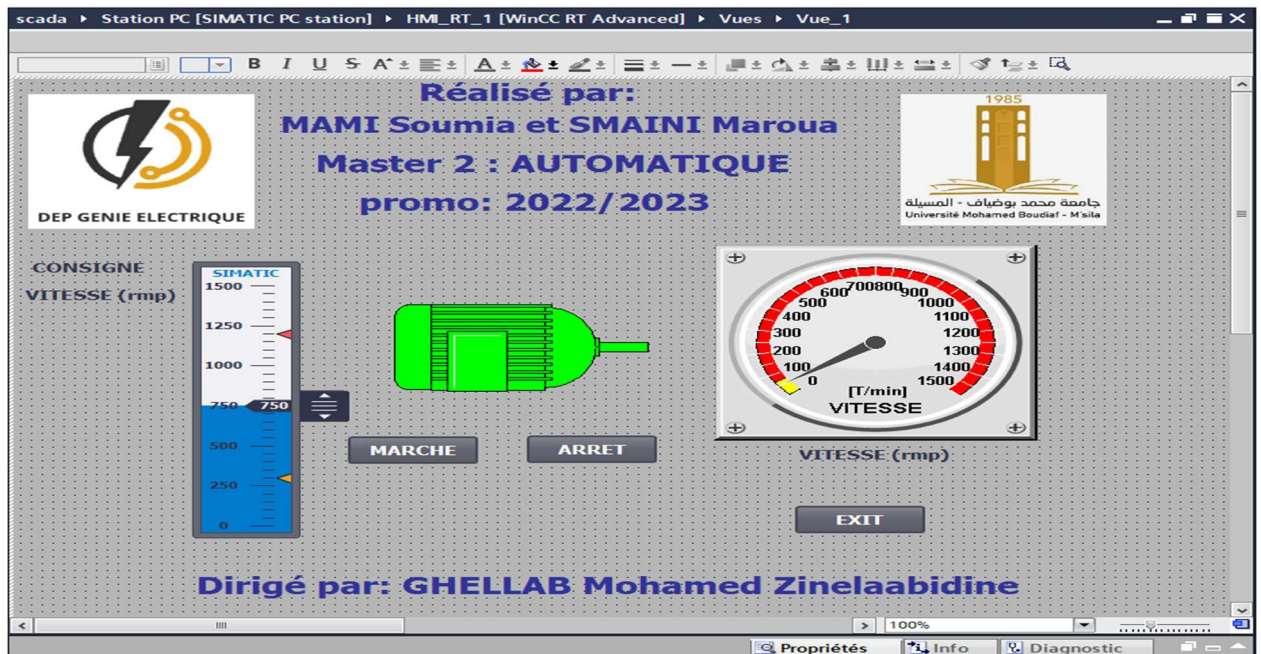


Figure III.15 : L'HMI de notre station

### III.7 Conclusion :

L'HMI et Win CC sont des outils importants pour la surveillance et la supervision des processus automatisés dans l'industrie.

La description précédente de l'outil Win CC flexible, prépare le terrain en vue de l'application

« **Pilotage et supervision programmable d'un Moteur à courant continu par automate programmable SIEMENS** », et pendant laquelle une mise en œuvre complète sera décrite pour élaborer une interface Homme/machine.

## ***Chapitre IV :***

***Simulation et commande d'un  
moteur à courant continu(MCC)***

### IV.1. Introduction :

Dans l'industrie on s'intéresse à la machine à courant continu car il est possible d'obtenir de manière relativement simple sa modélisation dans le cas où le flux agissant sur l'induit est constant. L'apparition et l'évolution des circuits intégrés des microprocesseurs et des techniques de commande ont nous permis d'investir dans de nouveaux domaines techniques [9].

La commande des machines électriques utilise des convertisseurs statiques cette commande nécessite l'association d'une machine dont le fonctionnement est à une vitesse variable ,à un convertisseur statique (hacheur).Le moteur à courant continu est très utilisé dans la chaîne d'automatisme , robotique et dans de nombreuses applications industrielles ,il présente des avantages importants dans les systèmes de commande, en raison de sa très grande souplesse pour l'asservissement de vitesse et de position.

Dans ce chapitre nous présenterons des généralités sur le moteur à courant continu, préciser sa construction, sa modélisation, la commande en BO, Création du projet S7 pour la commande d'un MCC et L'interface par logiciel Win CC.

### IV.1 Définition :

Le moteur électrique à courant continu à collecteur fut la première machine électrique inventée en 1867, suivi par ZENOB Gramme en 1869 [10].

Le moteur à courant continu est un convertisseur électromécanique d'énergie Il permet de transformer une énergie électrique continue en une énergie mécanique pour entrainer une charge en mouvement Un moteur électrique à courant continu est constitué :

- D'un stator qui est l'origine de la circulation d'un flux magnétique longitudinal fixe créer soit par des bobinages soit par des aimants permanents à stator, se trouve la partie porte balais et les contacts électriques avec le rotor(inducteur).
- D'un rotor bobiné relié à un collecteur rotatif inversant la polarité dans chaque enroulement de rotor au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux stator Les enroulement de rotor sont aussi appelés enroulement d'induits, ou communément induit.

### IV.3 Les avantages de Moteur à courant continu :

- Une large gamme de variation de vitesse au-dessous de la vitesse de régime.
- Un fonctionnement avec des couples constants ou variables.
- Une accélération un freinage et une inversion du sens de rotation très rapièce qui est avantageux dans le cas des appareils de lavage et des machines
- Une vitesse de rotation qui peut être réglée par l'intermédiaire d'un système de .
- La possibilité de fonctionner comme générateur lors de freinage par récupération d'énergie.

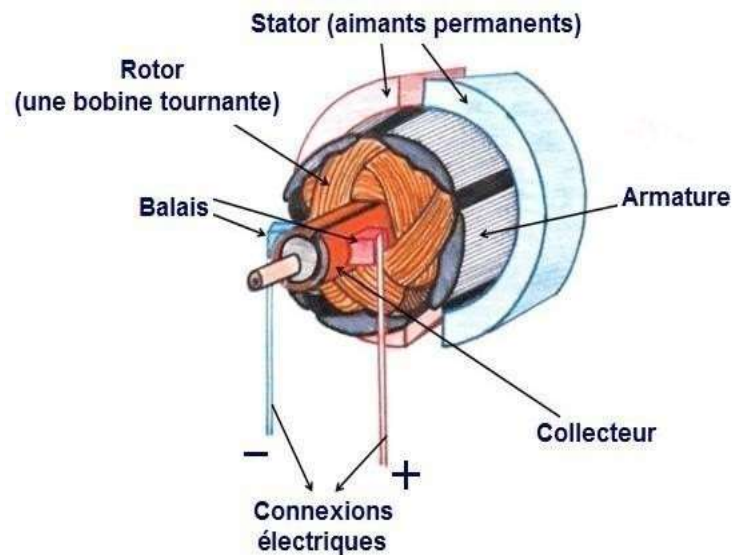


Figure IV.1 : Représentation d'une MCC

### IV.4 Modélisation du moteur à courant continu :

Le modèle électrique du MCC est représenté par la f.e.m. « E », la résistance interne R et l'inductance d'induit L suivant :

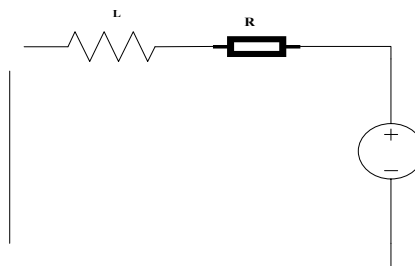


Figure IV.2 : Schéma électrique d'un moteur à courant continu

D'après la loi d'Ohm en négligeant l'inductance L on:

$$U = E + RI$$

U : Tension d'alimentation d'induit

E : Force électromotrice

R : Résistance de bobinage d'induit

I : Courant d'induit

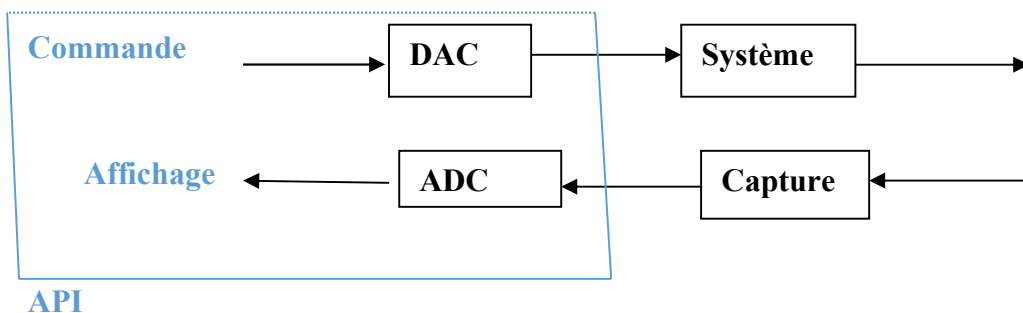
#### IV.5 Commande d'un moteur à courant continu

Cette section a pour but de montrer les possibilités qu'offre un automate programmable pour commander un système continu. Nous avons traité une application pour la commande en boucle ouverte de la vitesse d'un moteur à courant continu (système rapide).

##### IV.5.1 Commande en boucle ouverte avec automate programmable

Cette application a pour but de montrer les possibilités qu'offrent les automates programmables pour la commande des systèmes rapides (dynamique rapide). C'est une commande en boucle ouverte de la vitesse d'un moteur à courant continu avec une boucle de mesure de la vitesse et de la position.

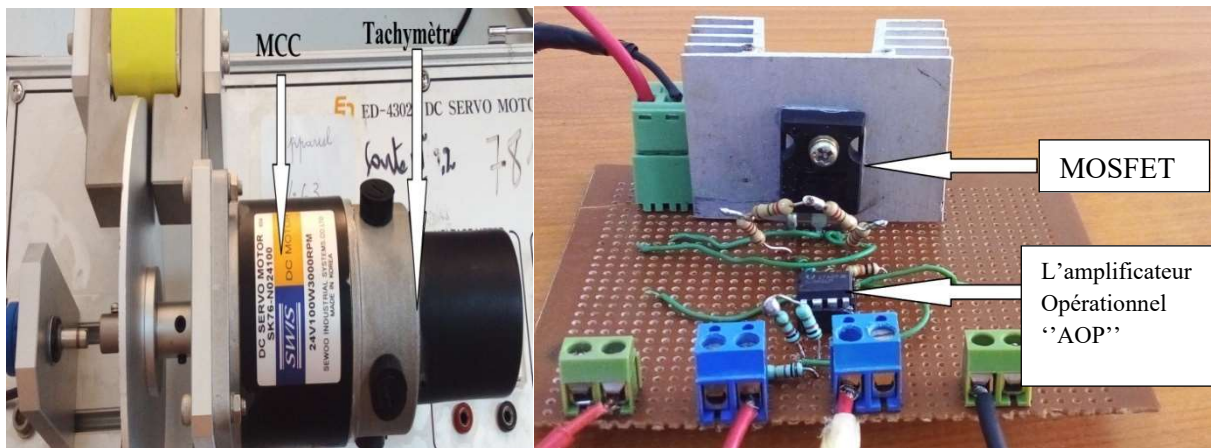
Le schéma de principe d'une commande numérique via automate est le suivant :



**Figure IV.3 :** Commande en BO avec boucle de mesure.

Il s'agit de fixer la vitesse d'un moteur DC, la consigne de vitesse est introduite par le potentiomètre avec le signal de commande est une tension continue comprise entre 0 et 10 V.

Le système est constitué d'un moteur, une tachymétrie et d'un circuit d'amplification.



FigureIV.4 : Le moteur et le circuit d'amplification

### IV.5.2 Rappels sur moteur à courant continu

Les moteurs à courant continu sont des machines réversibles. Ils transforment l'énergie électrique en une énergie mécanique et vice versa. Ils comportent :

- **Le stator (inducteur) :** c'est la partie fixe, constitué d'un aimant permanent ou d'un électroaimant, elle crée un champ magnétique dirigé vers l'axe du rotor.
- **Le rotor (induit) :** c'est la partie mobile, constitué d'un cylindre avec des spires à la périphérie.
- **Le collecteur :** c'est partie mobile constituées de lames, reliées aux spires. L'excitation peut être réalisé de diverses manières : séparée, shunt, série, compound et par aimants permanents. Ce dernier est le plus utilisé pour les petits moteurs.

Pour le moteur à aimants permanents, le schéma de fonctionnement est représenté dans la figure ci-dessous :

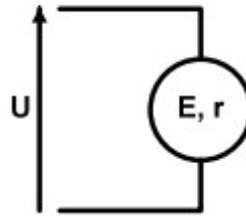


Figure IV.5 : Moteur CC à aimant permanents

$R_c$  est la résistance de la charge.

$$E = r \cdot I + U \text{ Avec } E = K_e \cdot \omega \text{ et } U = R_c \cdot I$$

$$\text{D'où } U = \left( \frac{K_e R_c}{r + R_c} \right) \cdot \omega \text{ donc } U = K_e \cdot \omega$$

$K_e$  est la constante de la f.e.m. il est donné en V/Tr/min. C'est le rapport entre la vitesse de rotation est la tension délivrée. La caractéristique tension-vitesse est donc linéaire (en régime permanent)

### IV.5.3 Description du matériel

C'est un moteur continu à aimants permanents, qui tourne à des vitesses angulaires entre 0 et 1400 tr/mn pour des tensions d'alimentation comprises entre 0 et 12V. La Tachymétrie délivre une tension comprise entre 0 et 2.5V pour des vitesses angulaires entre 0 et 1400 Tr/mn ( $K_e = 1.8 \cdot 10^{-3}$ ).

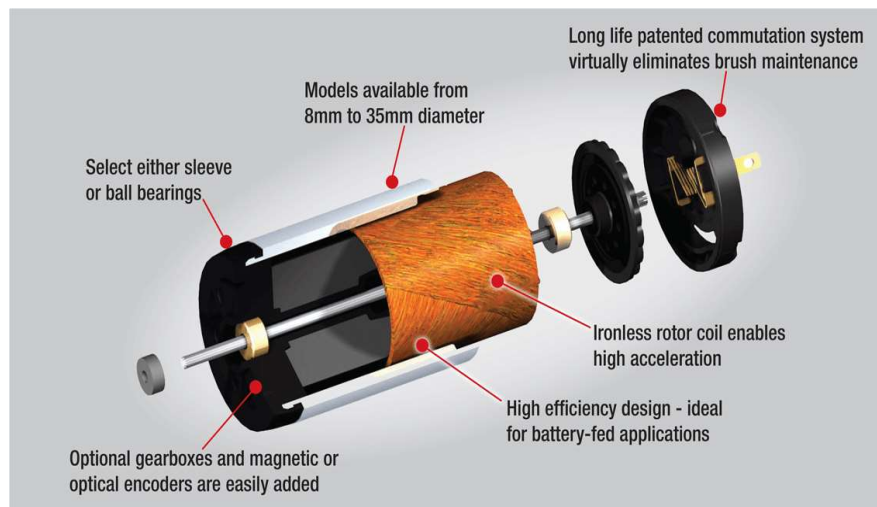


Figure IV.6 : Moteur à courant continu

L'entrée analogique de l'automate accepte une tension entre 0 et 10V, on doit donc faire une Amplification du signal de la tachymétrie pour avoir une meilleure précision.

Le schéma électrique du circuit d'amplification est présenté dans la figure ci-dessous :

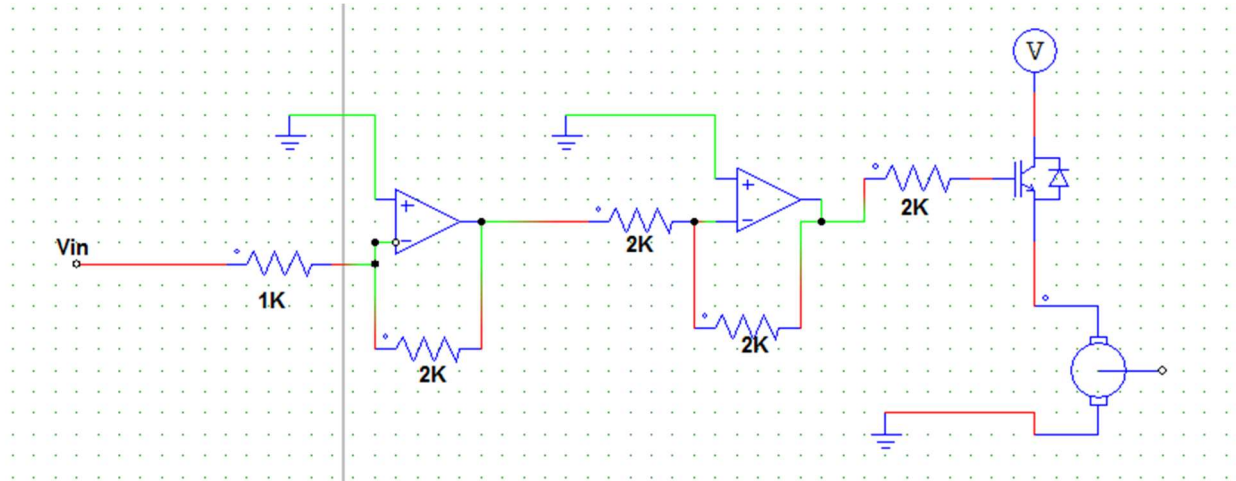
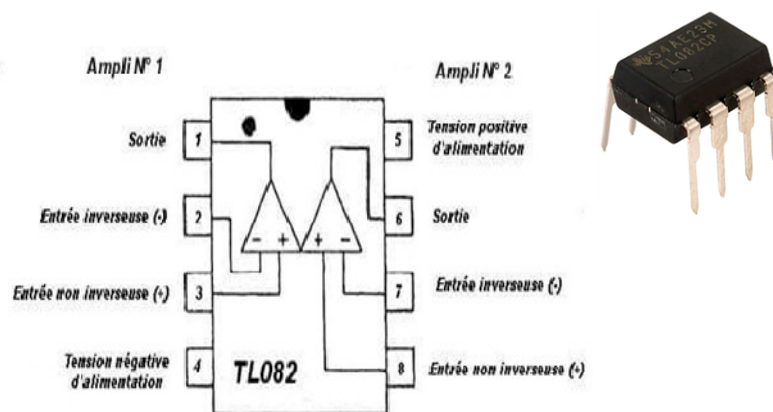
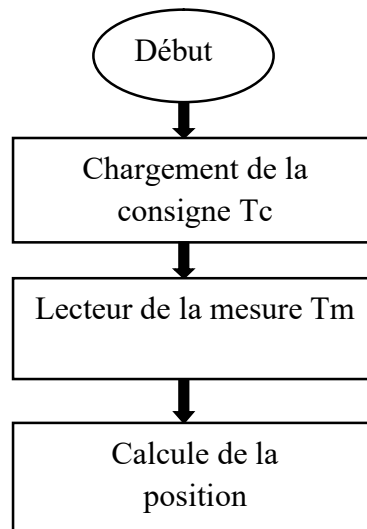


Figure IV .7 : Schéma électrique du circuit d'amplification.

L'alimentation de ce circuit est assurée par un générateur de tension continu.

La datasheet d'amplification est présentée dans la figure ci-dessous :



**IV.6 Description du programme :****Figure IV.8 :** Organigramme de la commande en BO.

Chacun de ces blocs est représenté dans le programme par une fonction FC. Ces blocs sont appelés et exécutés dans le bloc d'organisation.

**IV.6.1 Chargement de la consigne :**

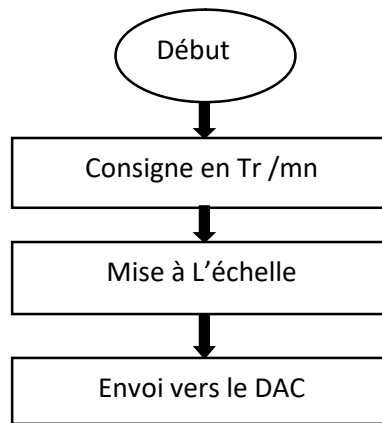
La consigne peut être chargée de deux façons :

**Manuel :** la consigne est introduite dans le programme (mode hors-ligne)

**Automatique :** la consigne est introduite par en manipulant un potentiomètre qui délivre une tension entre 0 et 10V, ce signal est converti en donnée numérique après passage par l'entrée analogique.

La consigne est introduite manuellement dans le programme (hors ligne). Elle est donnée en Tr/mn, il faut la convertir en valeur décimale entre 0 et 27648, ce qui correspond à la sortie analogique de l'automate (DAC) à une tension entre 0 et 10V.

Voici l'organigramme ci-dessous l'organigramme de la fonction chargement consigne :

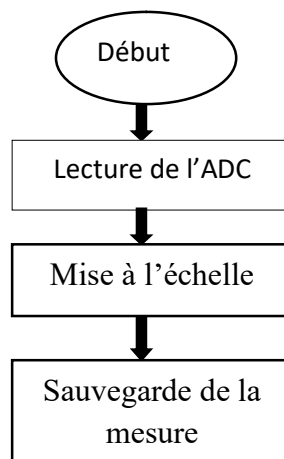


**Figure IV.9 :** Organigramme de chargement de la consigne.

#### IV.6.2 Lecture de la vitesse

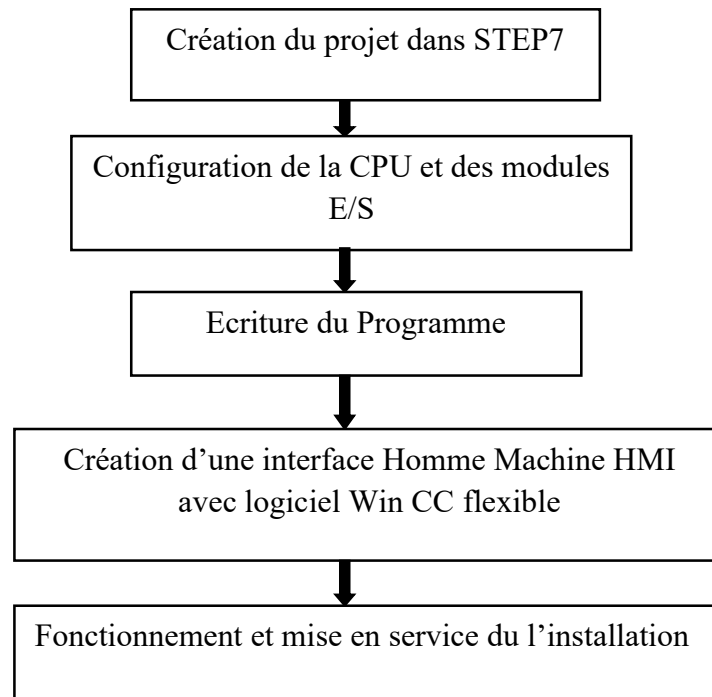
Une solution consiste à prendre le maximum des valeurs lues, tensions correspondantes, lus sur un voltmètre connecté aux bornes du capteur.

La génératrice tachymétrie délivre une tension entre 0 et 10V pour une vitesse angulaire entre 0 et 1400Tr/mn. Cette tension connectée à l'entrée analogique de l'automate. Comme pour le chargement de la consigne une mise à l'échelle doit être mise en place.



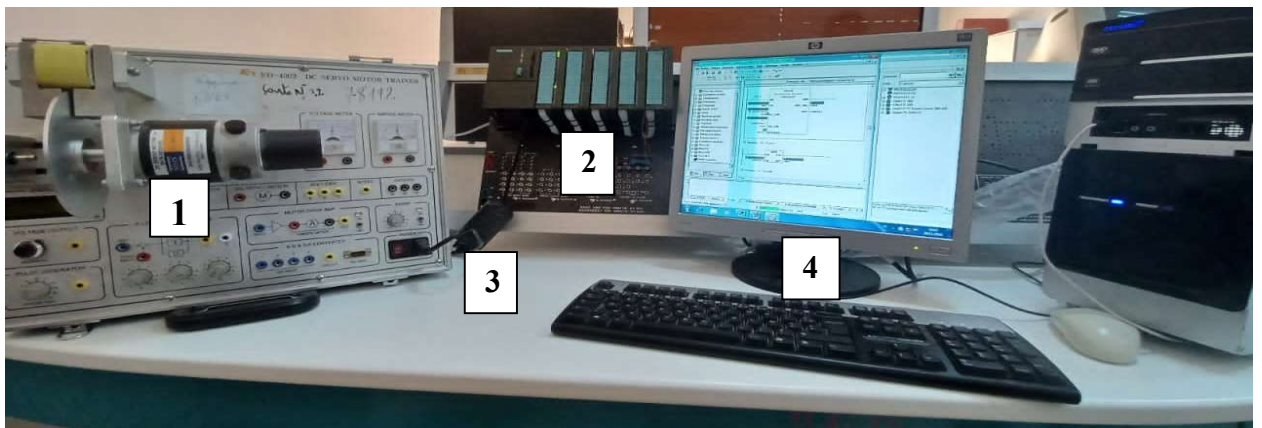
**Figure IV.10 :** Organigramme de lecteur de la vitesse.

L'étape suivante consiste à créer et développer le programme de contrôle et de commande qui sera implémenté dans l'automate afin de gérer et la commande d'un moteur.



**Figure IV.12 :** Organigramme de l'application finale.

Le schéma suivant résume ce qui était dit précédemment :



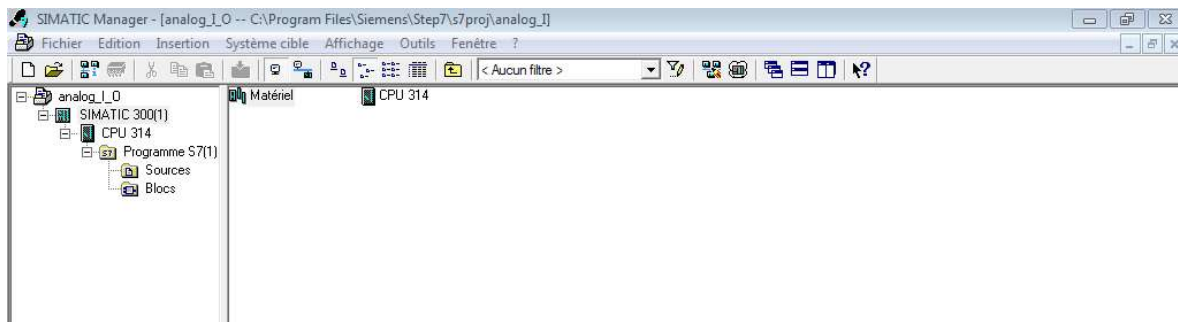
**Figure IV.13 :** prototype de la commande par API.

- 1-Moteur à courant continu ;
- 2- S7-314 ;
- 3-câble MPI ;
- 4- le PC.

**IV.7 Création du projet S7 pour la commande d'un MCC :**

**IV.7.1 Insertion des stations dans SIMATIC MANAGER**

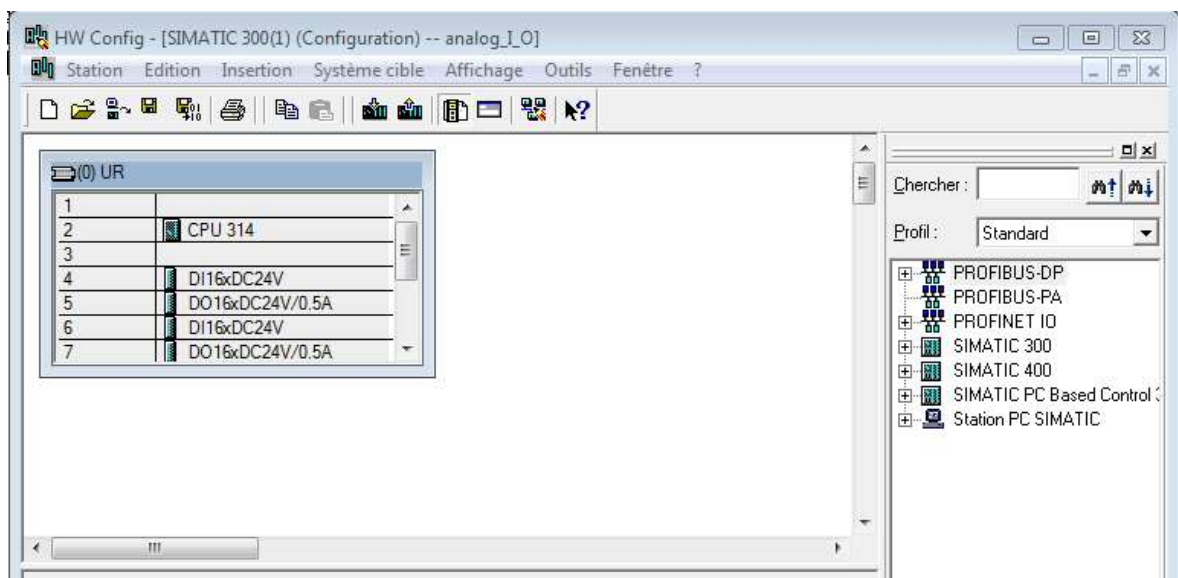
Le projet est créé comme selon la procédure vue dans le chapitre xx. Après insertion de la station SIMATIC300 ainsi qu'une station HMI pour la supervision telle qu'il est montré dans la figure suivante :



**Figure IV.14 : Création du projet**

**IV.7.2. Configuration du matériel dans STEP7**

On sélectionne une CPU 314, avec les caractéristiques citées précédemment, il est montré dans la figure suivante :



**Figure IV.15 : Configuration matériel.**

IV.8 Application et visualisation de la commande :

Nous avons appliqué cette commande pour une consigne en vitesse égale à 360 Tr/mn.



Figure IV.16 : Vue générale crée pour la HMI

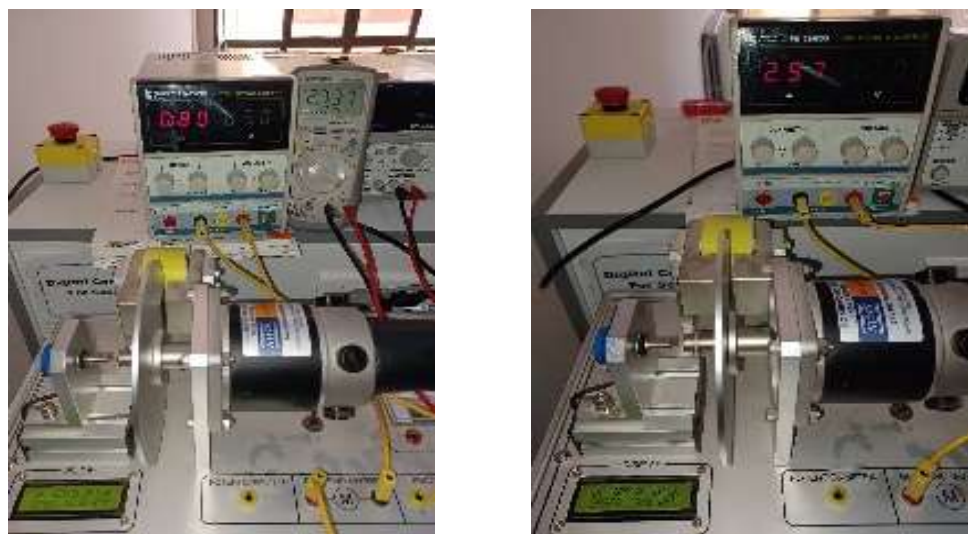


Figure IV.17 : La tension et la vitesse du système 5Entrée/Sortie).

La réponse du système (vitesse + commande) est relevée manuellement à partir d'un multimètre.

- Si  $U=5.0V$ , la vitesse est égale à 408 tr/min.
- Si  $U=12.0V$ , la vitesse 1356 tr/min.

La réponse du système est relevée automatiquement puis visualiser par PC (interface HMI.

Cette figure a été prise à l'aide d'un multimètre branché aux bornes de la sortie analogique de l'automate.

La commande admissible du moteur étant comprise entre 0 et 12V, pour une vitesse angulaire entre 0 et 1400 Tr/mn, la consigne de 400 Tr/mn donnée à l'automate correspond parfaitement à la moitié du signal de commande maximale (12V).

La réponse du système est prélevée à l'aide d'un multimètre branché aux bornes de la génératrice tachymétrie est présenté dans la figure suivante :

**Figure IV.18** : La réponse de la vitesse et tension correspondante



On voit bien que le signal se stabilise autour de la tension 2.73V, ce qui correspond à une vitesse angulaire égale à 408 Tr/mn.

On remarque que l'erreur statique est un très grande 30%. Pour l'annuler il faut introduire un correcteur PI.

IV.9 Programme de la commande du moteur :

- Programme dans FC2 Lecture de la mesure :

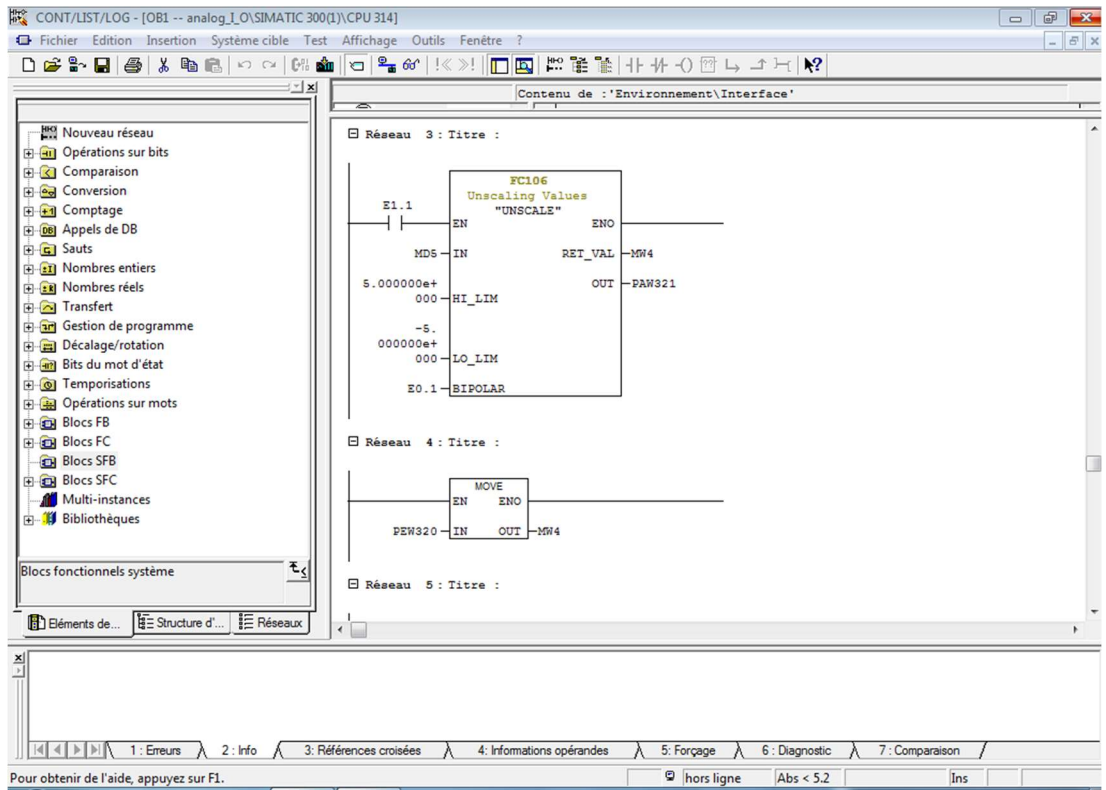


Figure IV.19 : Bloc UNSCALE

- Programme dans FC3 Chargement consigne :

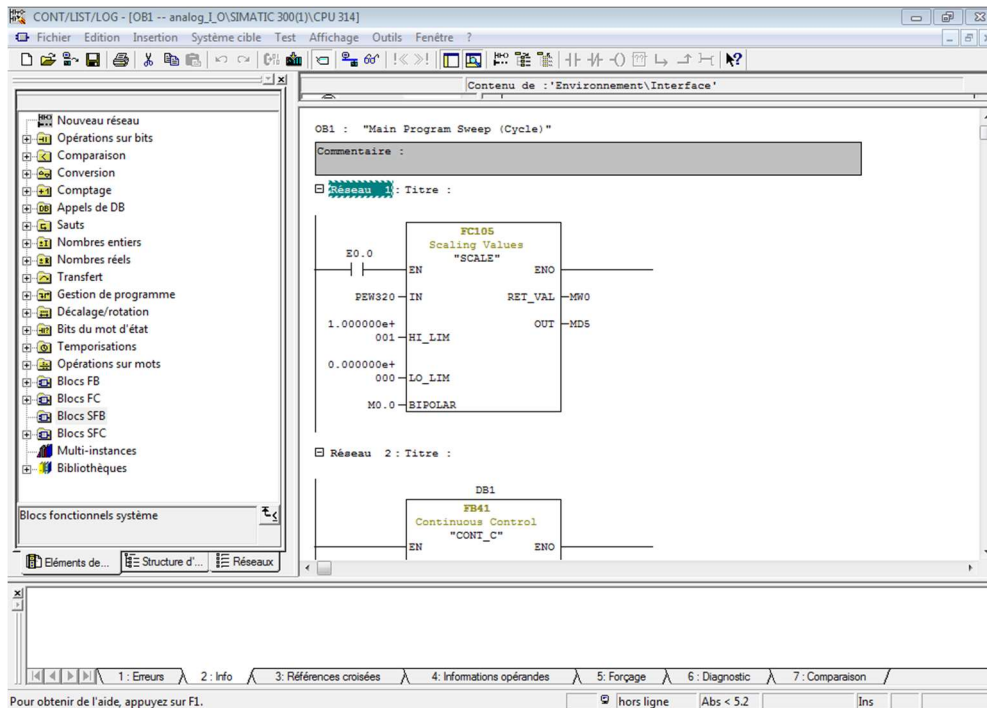


Figure IV.20 : Bloc SCALE

### **IV.10 CONCLUSION**

Le moteur à courant continu est très utilisé dans la chaîne d'automatisme et robotique, Il présente des avantages importants dans les systèmes de commande.

Cette étude pour commande en boucle ouverte, La première étape est chargement de la consigne, Lecteur de la vitesse et calcul de la position. Le programme est créé dans SIMATIC MANAGER et la deuxième étape est création d'une HMI avec logiciel Win cc flexible.

## ***Conclusion Générale***

### CONCLUSION GENERALE

Ce mémoire étudie les différentes étapes d'élaboration d'un projet réel et l'établissement d'une commande à base d'automate programmable SIEMENS grâce au logiciel de programmation STEP7 qui sera chargé dans l'API S7-314 et le logiciel pour la supervision Win CC flexible. A la lumière des résultats obtenus, de nombreuses perspectives s'ouvrent à nous :

- Mettre en œuvre des commandes avancées au niveau de l'automate pour la régulation de vitesse et la position (commande PID, commandes robustes....).
- Traiter en plus approfondi « la télégestion » dans le milieu industriel.
- Mettre en œuvre les connaissances acquises sur des installations réelles.

En fin, nous espérons que notre travail sera une meilleure solution à la problématique posée et servira comme une base de départ pour notre vie professionnelle et être bénéfique aux promotions futures.



## REFERANCES BIBLIOGRAPHIES

- [1] Benrezia Hocine. « Automatisation D'une Porte Coulissante Programmation Avec Step7 », Memoire De Master, Universite Mohamed El Bachir-Bordj Bou Arreridj, 2021.
- [2] Mekrouf Samira, « Conception Et Supervision D'une Commande Programmable A Base D'une Api », Memoire De Master, Universite Mouloud Mammeri De Tizi-Ouzou, 2009.
- [3] Alain Gonzaga, Les Automates Programmables Industriels.
- [4] G.Michel. « Les A.P.I Architecteur Et Application Des Automates Programmables Industriels », Edition Dunod, 1987.
- [5] Boubaaya Nadjib, « Etude Et Realisation De Systeme Automatisees Diduction Mise En Œuvre De L'automate Schneider Tsx37 », Memoire De Master, Universite Constantine 1, 2014
- [6] El Hammoumi, « Les Automates Programmables Industriels », Memoire De Master, Universite Sidi Mohamed Ben Abdellah De Fès.
- [7] M. Sedrati, « Interfaces Homme -Machine », 2015.
- [8] Mohamed Tounsi (Isims), « Interfaces Homme -Machine », Memoire De Master, Institut Superieur D'informatique Et De Multimedia Sefax, Fevrier 2015.
- [9] Diche Bouchra, « Modelisation Et Simulation De La Commande D 'Un Moteur A Courant Continu », Memoire De Master, Higher School In Applied Sciences, Tlemcen, 2020.
- [10] Mehimmedetsi Bou Djemaa, « Application Du Formalisme Bond Graph A Une Chaîne De Conversion D'énergie Photovoltaïque », Memoire De Master, Universite Mentouri De Constantine, 2007.
- [11] [www.Wikipedia.com](http://www.Wikipedia.com)
- [12] [www.Siemens.com](http://www.Siemens.com)
- [13] Siemens, "Programmation Avec Step7", Simatic, 2000.
- [14] [www.Plctutorialpoint.com](http://www.Plctutorialpoint.com)
- [15] [www.Flexio.fr](http://www.Flexio.fr)
- [16] Kebaili Ouiza, « Automatisation De La Presse Profiles-U Par Un Automate Programmable S7\_300 Et La Supervision Par Le Win Cc », Memoire De Master, Universite Mouloud Mammeri De Tizi-Ouzou
- [17] Manuels Siemens : « Wincc Flexible », Simatic, 2008.
- [18] Touiker Wafide, « Supervision Du Systeme Anti-Incendie De La Turbine Man Thm-1304 Dln Au Niveau De La Region Rhourde Nouss Dans L'usine Csc A L'aide Des Logiciels Step7 Et Win cc », Memoire De Master, Universite Kasdi Merbah Ouargla.

## REFERENCES BIBLIOGRAPHIES

---