

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE MINISTÈRE DE  
L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOHAMED BOUDIAF - M'SILA

FACULTE : TECHNOLOGIE

DÉPARTEMENT : GENIE ÉLECTRIQUE

N° : ELM\_13



DOMAINE : SCIENCES ET TECHNOLOGIES

FILIÈRE : ELECTROMECHANIQUE

OPTION : ELECTROMECHANIQUE

Mémoire de fin d'études en vue de l'obtention du diplôme de

**MASTER Académique**

**Spécialité : Electromécanique**

**Intitulé**

**SIMULATION ET COMMANDE EN TEMPS REEL  
D'UN ENTRAINEMENT ELECTRIQUE PAR  
L'AUTOMATES PROGRAMMABLES SIEMENS**

**Présenté par :**

CHENITI Haroun

YAHY Abdelhadi

Examiné par le jury composé de :

**Mr : ZOUGUAR Eloualid**

**Mr : GHELLAB Mohamed Zinelabidine**

**Mr : KHETTAB Khatir**

**Président**

**Rapporteur**

**Examineur**

Année universitaire : 2022/2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# إهداء

وصلت رحلتي الجامعية إلى نهايتها بعد تعب ومشقة، ومع ذلك حاولت أن أتخطاها بثبات بفضل من الله وحده، اللهم لك الحمد على نعمك

اللهم صلي وسلم على نبيك وحبيبك سيدي محمد ابن عبد الله العربي صلى الله عليه وسلم  
اللهم اغفر لأبي عزيز واحمه واسكنه فسيح جناتك، اللهم إن كان في طلبي للعلم أجرا اجعلها في ميزان حسناته: أبي مفتاح شنييتي

إلى التي بالأمني حملتني، وبالتهاني إستقبلتني، إلى من أفضلها على نفسي ولم لا فلقد ضحت من أجلي، ولم تدخر جهداً في سبيل إسعادي على الدوام لكي: أمي الغالية

إلى الذي أظهر لي الطريق الصحيح بتذكيري بأن الإرادة والعمل تصنع رجال عظماء، إلى الشخص الذي أفخر بأن أقول انه أخي، وأعتز به في كل مكان : أخي أنور شنييتي

إلى الذين كانوا دائما سندي وحافزي وعزي في الحياة عائلتي الصغرى أختي وعائلتها وعائلة أخي وعمتي الصغرى أو بالأحرى أمي الثانية وعائلتها

إلى من كانوا عزي وجاهي وافتخاري ومفخرتي عائلتي الكبرى كل باسمه وعزه من أعمام وعمات وأحوال وخالات بكامل عائلاتهم

إلى الذين ساهموا بالكثير في المسار الدراسي من دعم مادي ومعنوي وتربوي بكل فخر أقول لهم بوركتم جهودكم : عمي بوزيد شنييتي، خالي صابر شنييتي

إلى البركة والعز وأم الكبرى جدتاي أطل الله في عمركما ومنا علينا برويتكم معنا

إلى كل أحبائي أصدقائي إخواني كل باسمه ومقامه (م،م،ل،ب،إ،ع،م،م)

أستاذي ومشرفي وقدوتي في التعليم العالي غلاب محمد زين العابدين

اللهم أرحم وأغفر لأجدادي عمر وجمعي ربي أرحمهما

هارون شنييتي

# إهداء

الحمد لله على منه و امتنانه والشكر له على نعمه وإنعامه حمدا كثيرا طيبا، الذي أنعم علي  
بنعمته العلم وسهل لي طريقا أبغى فيه علما ووفقتني في إنهاء عملي.

إلى من بلغ الرسالة وأدى الأمانة ونصح الأمة إلى نبي الرحمة ونور العالمين، سيدنا  
محمد عليه الصلاة و أزكى التسليم وسلم.

إلى التي بالأمانى حملتني ، وبالتهاني إستقبلتني، إلى من أفضلها على نفسي ولم لا فلقد ضحت من  
أجلي، ولم تدخر جهداً في سبيل إسعادي على الدوام لكي: أُمي الغالية.

إلى الذي أظهر لي الطريق الصحيح بتذكيري بأن الإرادة والعمل تصنع رجال عظماء، إلى الشخص الذي  
أحمل أسمه بكل فخر، وأعتز به في كل مكان:أبي العزيز.

إلى سندي وقوتي وملاذي، إلى من علموني علم الحياة إلى القلوب الطاهرة :إخوتي و أخواتي وإلى كل  
الأهل والأقارب إلى لأصدقاء الأوفياء والزملاء الأعزاء التي جمعتني بهم الحياة.

ممن كان لهم الدور الأكبر في ولا ينبغي أن أنسى أستاذنا المشرف غلاب محمد زين العابدين  
مساندتي ومدي بالمعلومات القيمة.

إلى أولئك الذين شجعونا من بعيد أقول لكم شكرا جزيلا لكم جميع.

ياحي عبد الهادي

# شكر وتقدير

سبحانك اللهم لا علم لنا إلا ما علمتنا ، نشكر الله ونحمده فضل نعمه علينا، نعمة العقل التي أنار بها  
دربنا وفكرنا، والصلاة والسلام على قدوة المرابين نبينا محمد عليه الصلاة و أزكى التسليم وسلم.

إن من تمام شكر الله ،شكر أهل الفضل والبر، قال رسول الله (صلى الله عليه وسلم): "مَنْ صَنَعَ إِلَيْكُمْ  
مَعْرُوفًا فَكَافَيْتُوهُ، فَإِنْ لَمْ تَجِدُوا مَا تَكْفِيُونَهُ فَادْعُوا لَهُ حَتَّى تَرَوْا أَنْكُمْ قَدْ كَافَأْتُمُوهُ". (رواه أبو داوود).

ونخص بالشكر الجزيل لأساتذة الخير الذين علمو بلا شك أن العلم من أجمل العبادات وأفضلها، وإن  
من أماننا وتطلعاتنا في هذا الصرح أن أتقدم بجزيل الشكر إلى كل من ساعدنا وساهم في تكويننا  
طيلة مشوارنا الدراسي من أساندة التعليم الابتدائي وصولا إلى أساندة التعليم العالي والبحث العلمي.

ثم الشكر الموصول للأستاذ المشرف غلاب محمد زين العابدين فهو كان التيسير للعسير، ولك الشكر  
حار جدا على كل ما قدمته لنا وساعدتنا عليه أستاذنا الفاضل، شكر يفوق حدود الخيال على  
مساعدتك لنا في تحقيق الآمال.

كما لا يفوتنا أن نشكر الذين ساعدونا في إنجاز هذا المشروع المتواضع، كل بإسمه ومقامه منهم :  
الأستاذ زوقار وليد ، الأستاذ خوجة جلال الدين ، الطالب مسلمي إسماعيل ، الأخ محجوبي لخضر ،  
عميد كلية علوم والتكنولوجيا ، رئيس قسم الهندسة الكهربائية شوشو عبد المجيد وكل الطاقم  
الإداري كل بإسمه ومقامه .

## Table des matières

<b>Introduction Générale :</b> .....	<b>1</b>
<b>I. Automate programmable industriel</b> .....	<b>3</b>
<b>I.1 Introduction</b> .....	<b>3</b>
<b>I.2 Historique</b> .....	<b>3</b>
<b>I.3 Définition générale</b> .....	<b>4</b>
<b>I.4 Domaines d'utilisation des automates</b> .....	<b>4</b>
<b>I.5 Les Avantages d'un API</b> .....	<b>4</b>
<b>I.6 Les Inconvénients d'un API</b> .....	<b>5</b>
<b>I.7 Architecture des automates programmables</b> .....	<b>5</b>
<b>I.8 Présentation de la gamme SIMATIC de SIEMENS</b> .....	<b>8</b>
<b>I.8.1 Les différentes variantes dans la gamme SIMATIC</b> .....	<b>9</b>
<b>Conclusion :</b> .....	<b>11</b>
<b>II Le STEP7-SIMATIC MANAGER</b> .....	<b>12</b>
<b>II.1 Introduction</b> .....	<b>12</b>
<b>II.2 Description du STEP7</b> .....	<b>12</b>
<b>II.3 Gestionnaire de projets SIMATIC Manager</b> .....	<b>12</b>
<b>II.4 Editeur de programme et les langages de programmation</b> .....	<b>13</b>
<b>II.5 Interface de communication PC- APC</b> .....	<b>14</b>
<b>II.6 Le simulateur des programmes PLCSIM</b> .....	<b>14</b>
<b>II.7 Stratégie pour la conception d'une structure programme complète et optimisée</b> .....	<b>15</b>
<b>II.7.1 Création du projet SIMATIC Step7</b> .....	<b>16</b>
<b>II.8 Création d'un projet STEP 7</b> .....	<b>17</b>
<b>II.9 Configuration matérielle (Partie Hardware)</b> .....	<b>18</b>
<b>II.10 Création de la table des mnémoniques (Partie Software)</b> .....	<b>20</b>

<b>II.11 Elaboration du programme S7 (Partie Software)</b> .....	<b>20</b>
<b>II .12 Exemple d'application de Création et d'édition d'un projet S7</b> .....	<b>22</b>
II.12.1 La mise en œuvre de l'application.....	25
II.12.2 Description les éléments de l'installation pour l'application.....	25
II.12.3 Création du projet dans STEP7.....	26
II.13.4 Elaboration du programme S7 (Partie Software).....	28
II.13.5Simulation de modules .....	32
<b>CONCLUSION</b> .....	<b>33</b>
<b>III. Le Progiciel De Conception Des Interfaces Homme-Machine</b> .....	<b>34</b>
<b>III.1 Introduction</b> .....	<b>34</b>
<b>III.2 Interface homme-machine</b> .....	<b>34</b>
<b>III.2 Description générale de SIMATIC HMI</b> .....	<b>34</b>
<b>III.3. Eléments de WinCC flexible</b> .....	<b>35</b>
<b>III.4 Mise en service un projet de WinCC flexible</b> .....	<b>37</b>
III.4.1 Création d'un nouveau projet.....	38
III.4.2 Sélection de pupitre.....	38
III.4.3 sélections projet WinCC sur STEP7 Manager .....	39
III.4.4 Etablissement de la liaison Automate-HMI .....	40
III.4.5 Utilisation des objets .....	41
III.4.6 Utilisation des variables .....	43
III.4.7 Compilation et Simulation .....	44
<b>Conclusion</b> .....	<b>45</b>
<b>IV : Commande d'un moteur à courant contenu</b> .....	<b>47</b>
<b>IV.1 Introduction :</b> .....	<b>47</b>
<b>IV.2Commande en boucle ouverte avec automate programmable</b> .....	<b>47</b>
<b>IV.3 Rappels sur moteur à courant continu</b> .....	<b>48</b>
<b>IV.4Description du matériel</b> .....	<b>50</b>
<b>IV.5Description du programme :</b> .....	<b>51</b>

IV.6Chargement de la consigne .....	52
IV.7Lecture de la vitesse .....	52
IV.9Création du projet S7 pour la commande d'un MCC.....	54
IV.10Application et visualisation de la commande .....	57
IV.11Programme de la commande du moteur : .....	59
Conclusion.....	60
Conclusion générale.....	55
Références bibliographiques.....	57

### *Liste de Figure:*

Figure. II. 22:Edition du GRAPH.....	32
Figure. II. 23 : édition des action &transitions .....	35
Figure .III.1 : L'interface Homme/Machine dans un processus automatisé .....	35
Figure .III.2 : La fenêtre principale de WinCC flexible.....	37
Figure III.3 : Etapes pour la conception d'une interface via Win CC flexible.....	37
Figure. III.4 :L'interface principale du logiciel WinCC flexible.....	38
Figure. III.5 : Sélection du pupitre.....	39
Figure. III.6 : Fenêtre d'une vue dans le WinCC flexible.....	39
Figure .III.7 :L'éditeur de liaison.....	40
Figure .III.8 : Création d'un bouton.....	42
Figure .III.9 : Fenêtre d'outils d'un bouton.....	42
Figure. III.10 : La fenêtre d'outils pour les objets de la bibliothèque.....	43
Figure III.11 : éditeur de variable dans le Win CC flexible.....	44
Figure .III.12 : le système Runtime.....	44
Figure .III.13Supervision par WINCC Runtime.....	45
Figure IV.1 : Commande en BO avec boucle de mesure .....	47

<b>Figure. IV.2</b> : Le moteur et le circuit d'amplification .....	<b>48</b>
<b>Figure. IV.3</b> : Moteur CC à aimant permanents.....	<b>48</b>
<b>Figure. IV.4</b> : Schéma de fonction d'une génératrice bathymétrique.....	<b>49</b>
<b>Figure .IV.5</b> : Moteur courant contenu .....	<b>50</b>
<b>Figure .IV.6</b> : Schéma électrique du circuit d'amplification.....	<b>50</b>
<b>Figure IV.7</b> : Organigramme de la commande en boucle ouverte .....	<b>51</b>
<b>Figure IV.8</b> : Chargement de la consigne.....	<b>52</b>
<b>Figure IV.9</b> : Lecteur de la vitesse.....	<b>53</b>
<b>Figure IV.11</b> : Organigramme de l'application finale.....	<b>53</b>
<b>Figure IV.12</b> : Synoptique de la commande par API.....	<b>54</b>
<b>Figure IV.13</b> : Création du projet .....	<b>55</b>
<b>Figure IV.14</b> : Configuration matériel .....	<b>55</b>
<b>Figure IV.15</b> : La tension et la vitesse du système.....	<b>56</b>
<b>Figure IV.15</b> : Vue générale crée pour la HMI.....	<b>56</b>
<b>Figure IV.16</b> : La tension et la vitesse du système (Entrée/Sortie).....	<b>56</b>
<b>Figure IV.17</b> :La réponse de la vitesse et tension correspondante .....	<b>57</b>
<b>Figure IV.18</b> : bloc UNSCALE.....	<b>58</b>
<b>Figure IV.19</b> : Bloc SCALE.....	<b>58</b>

### ***Liste de Tableau:***

<b>Tableau II. 01</b> : Adressage des entrées/sorties processus .....	<b>30</b>
<b>Tableau II.02</b> : Les actions dans le S7-GRAPH.....	<b>34</b>

## *Liste de l'abréviation*

**API : Automate Programmable Industriel.**

**MCC : Moteur à Courant Contenu.**

**PLC:Programabl Logic control.**

**SIMATIC: Siemens automatique.**

**PLCSIM: Programmable logic Controller simulator.**

**E/S : Entrée et sortie.**

**HMI : Human Machine Interface.**

**CONT : langage Contact.**

**LIST : langage liste.**

**LOG : langage logigramme.**

**GRAF : langage grafcet.**

**TOR : tout ou rien.**

**LD : LADDER diagramme.**

**CPU : Central Processing Unit.**

**PC : ordinateur.**

**PG : Programming consol.**

**MPI: Multi point interface.**

**DAC: Digital analog converter.**

**ADC: analog Digital converter.**

**RAM: random access memory.**

**ROM: random read memory.**

**EPROM: Erasable programmable random read memory.**

**PID: proportional integral derive**

# ***INTRODUCTION GENERALE***

### **Introduction Générale :**

Le développement massif des techniques de l'automatisme a permis le passage de la machine automatisée à celui des systèmes automatisés de production, qui gèrent l'alimentation en énergie et qui permettent d'avoir une meilleure qualité des produits en plus de la sécurité et de la flexibilité des processus, mais cela entraîne un accroissement des besoins, en particulier la manipulation d'un grand nombre de variables et la gestion de véritables flux de communication [1].

Cela explique que les systèmes câblés deviennent trop volumineux et trop rigides pour de telles applications, et que l'on se tourne donc vers des solutions utilisant les techniques de traitement de l'information par processeurs programmables ou les automates programmables industriels occupent une place de choix [2].

Un automate programmable est un système électronique, destiné à être utilisé dans un environnement industriel, il utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en œuvre de fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties divers types de machines ou de processus [3].

Les API (ou Programmable Logic Controller PLC) sont aujourd'hui les constituants les plus répandus des automatismes. On les trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services (gestion de parkings, d'accès à des bâtiments, contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes) et dans l'agriculture (composition et délivrance de rations alimentaires dans les élevages) et cela est dû surtout à leurs performances de sécurité [4].

Dans tous les secteurs où ils sont utilisés les API doivent remplir des tâches de commande en élaborant des actions suivant une algorithmique appropriée, à partir des informations données par des détecteurs (Tout ou Rien) ou des capteurs (analogiques ou numériques), et des tâches de communication avec des opérateurs humains ou avec d'autres processus (automates, calculateurs de gestion de production,).

## INTRODUCTION GENERALE

---

Bien que les automates programmables incluent des modules et des options permettant le contrôle ou la commande d'un système continu, beaucoup d'utilisateurs pensent que les APIs ne sont destinées qu'à la commande séquentielle.

Le but de ce travail est de faire la mise en œuvre d'automates programmables industriels de la famille SIMATIC S7-300 pour la commande d'un moteur à courant continu avec boucle de mesure et de traitement (commande en boucle ouverte).

Ce document englobe l'étude du logiciel de programmation STEP7, des automates utilisés ainsi que l'implémentation de quelques applications.

***Chapitre I :***

***Les Automates Programmables***

***Industriel***

## I. Automate programmable industriel

### I.1 Introduction

L'Automate programmable Industriel API, (PLC) est aujourd'hui le constituant le plus répandu dans les automatismes. On le trouve pratiquement dans tous les domaines industriels vu sa grande flexibilité.

Dans ce chapitre nous donnerons une présentation générale sur les automates programmables industriels et les domaines d'utilisation et on va étudier ses avantages et ses inconvénients et la connaissance d'architecture de (API), langage de programmation, langage de LADDER, les principales fonctions, Les caractéristiques d'un (API), Les marques de (API), Les critères de choix d'un (API), Des logiciels de programmation et leur description et les caractéristiques de l'automate S7-300

### I.2 Historique

Les automates programmables industriels sont apparus à la fin des années soixante aux États-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 grandes 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 Unidata. 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 Unidata, CII fusionne avec Honeywell-Bull, Philips délaisse l'informatique et Siemens rejoint Fujitsu pour devenir, aujourd'hui, un des plus grands constructeurs mondiaux [5]

### **I.3 Définition générale**

Un automatisme désigne en général le dispositif assurant le fonctionnement d'une machine ou d'une installation de production avec un minimum d'intervention humaine.

La réalisation et l'exploitation des automatismes industriels font appel à divers outils matériels et logiciels qui doivent être convenablement choisis et utilisés en vue de remplir certaines fonctions.

L'automate programmable est un Ordinateur industriel permettant d'effectuer des fonctions control, utilisé pour l'automatisation de machines ou de processus industriel et remplace les Systèmes automatisés câblés à base de relais.

### **I.4 Domaines d'utilisation des automates**

Les automates programmables sont utilisés dans de nombreux domaines industriels tels que l'industrie manufacturière, pétrolière et gazière, alimentaire et des boissons, pharmaceutique, automobile, minière, etc. Ils sont utilisés pour contrôler et automatiser les processus de production, de fabrication, de transport et de gestion des bâtiments. Les automates programmables offrent une grande flexibilité et une capacité à s'adapter à différents environnements et configurations, ce qui les rend très utiles pour améliorer l'efficacité et la productivité dans ces domaines. [6]

### **I.5 Les Avantages d'un API**

- Rapidité.
- Fiabilité professionnelle.

- Facilité de maintenance.
- Enormes possibilités.
- Modifications du programme faciles à une logique câblée.
- Simplicité du câblage.[2]

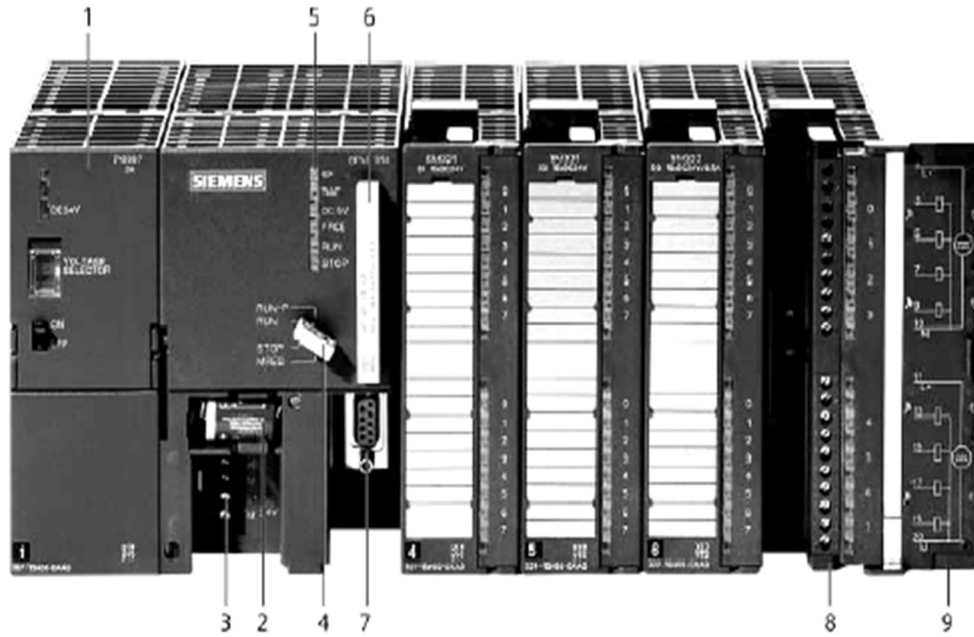
### **I.6 Les Inconvénients d'un API**

- Cherté.
- Nécessite une formation spécifique.
- En cas de « plantage » (très rare heureusement), c'est une belle pagaille.
- Son prix qui comme nous l'avons vu plus haut ne le met pas à la portée [6].

### **I.7 Architecture des automates programmables**

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 +/- 5V, +/-12V ou +/-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 [7].
- 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.



**Figure I. 1:** Automate SIMATIC S7-314

- |  |                               |
|--|-------------------------------|
| 1- Module d'alimentation                     | 6- Carte mémoire              |
| 2- Pile de sauvegarde                        | 7- Interface multipoint (MPI) |
| 3- Connexion au 24V                          | 8- Connecteur frontal         |
| 4- Commutateur de mode à clé                 | 9- Volet en face avant        |
| 5- LED de signalisation d'état et de défauts |                               |

❖ **structure Interne**

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire.

Un API se compose donc de trois grandes parties :

- Le processeur.
- La zone mémoire.
- Les interfaces Entrées/Sorties.

La structure interne d'un API peut se représenter comme suit :

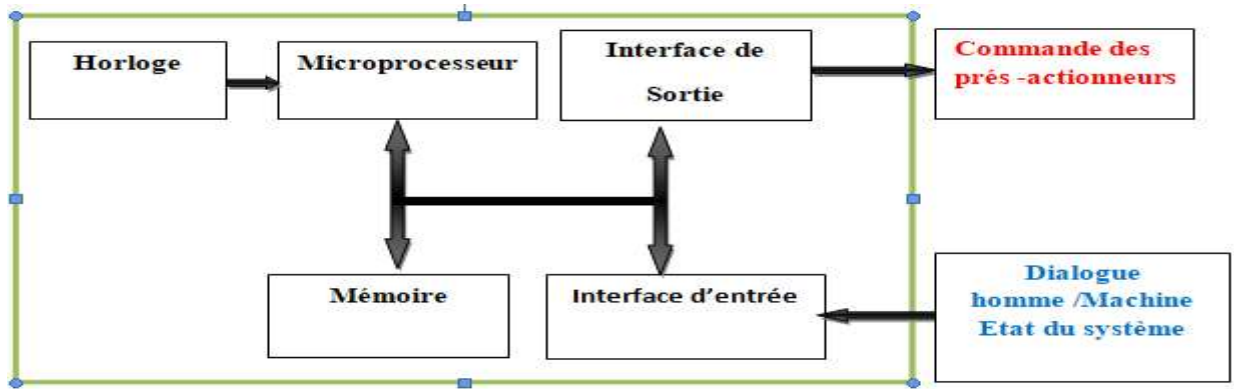


Figure I.3 : structure interne d'un API

### ➤ Le processeur

Il constitue le cœur de l'appareil dans l'unité centrale ; En fait, un processeur devant être automatisé, se subdivise en une multitude de domaines et processeurs partiels plus petits, liés les uns aux autres.

### ➤ Les modules d'entrées/sorties

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions. Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée :

- **Modules TOR** : l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 ...) C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir ...etc.
- **Modules analogiques** : l'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débit, niveau, pression, température...etc.).
- **Modules spécialisés** : l'information traitée est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

### ➤ Les mémoires : Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent :

- De stocker le système d'exploitation dans des ROM ou PROM.

- Le programme dans des EEPROM.
- Les données système lors du fonctionnement dans des RAM. Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA.
- **L'alimentation** : Elle assure la distribution d'énergie aux différents modules. L'automate est alimenté généralement par le réseau monophasé 230V-50 Hz mais d'autres alimentations sont possibles (110V ...etc.).
- **Liaisons de communication** : Elles Permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions. Les liaisons s'effectuent :
  - avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant le signal électrique ;
  - avec l'intérieur par des bus reliant divers éléments, afin de d'échanger des données, des états et des adresses.

### I.8 Présentation de la gamme SIMATIC de SIEMENS

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC. L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à:

- ✓ Une configuration et une programmation homogène des différentes unités du système.
- ✓ Une gestion cohérente des données.
- ✓ Une communication globale entre tous les équipements d'automatisme mis en œuvre

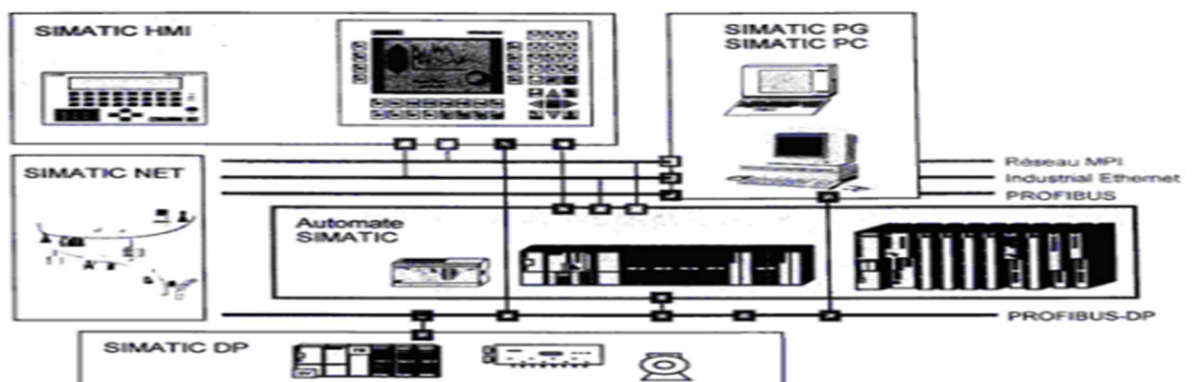


Figure I .4: Présentation de la gamme de SIMATIC [8]

### I.8.1 Les différentes variantes dans la gamme SIMATIC

a. **SIMATIC S7** Cette gamme d'automates comporte trois familles :

**S7 200**, qui 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 I.5:**L'API S200 [9]

**S7300** est un Mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et Industriel Ethernet



**Figure I.6:**L'APIS300[9]

- **S7400** est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industriel Ethernet.



**Figure I.7:** L'API S400 [9]

**b. 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 I .8:** La gamme SIMATIC C7 [9]

**c. SIMATIC M7**

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

Le M7 300/400 est capable d'effectuer simultanément avec une seule CPU des opérations en temps réel, par exemple des algorithmes complexes de commande, de régulation ainsi que des tâches de visualisation et de traitement informatique. Les logiciels sous DOS ou Windows sont exploitables sur le M7-300. Par ailleurs, avec son architecture normalisée PC, il permet une extension programmable et ouverte de la plate-forme d'automatisation S7.



**Figure I.9 :**La gamme SIMATIC M7 [9]

**I.9 Conclusion :**

Les automates programmables sont des outils essentiels pour l'automatisation des processus industriels. Ils offrent une grande flexibilité et une grande efficacité dans la gestion des tâches de contrôle en temps réel, ce qui permet d'optimiser la production et d'améliorer la qualité des produits. De plus, les automates programmables améliorent la sécurité des opérateurs et des équipements en intégrant des fonctionnalités de sécurité avancées.

Le SIMATIC S7-300 est une partie très importante dans l'automate programmable industriel polyvalent et fiable, capable de répondre aux exigences des environnements industriels les plus difficiles. Cependant, il est important de noter que chaque application a ses propres exigences spécifiques en matière d'automatisation. Il est donc essentiel d'analyser correctement les besoins de chaque application avant de choisir l'automate programmable le plus approprié.

***Chapitre II :***

***Le STEP7***

## II Le STEP7-SIMATIC MANAGER

### II.1 Introduction

STEP 7-SIMATIC Manager est un logiciel utilisé pour programmer les automates programmables Siemens (PLC) dans l'automatisation industrielle. Dans ce chapitre nous donnerons une présentation générale sur le STEP7 SIMATIC MANAGER et étude description step7 et l'interface de communication PC-APC et la connaissance de programmation sur STEP7 et étudiez ses avantages.

### II .2Description du STEP7

STEP 7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC S300 et S400. Il fait partie de l'industrie logicielle SIMATIC. Le logiciel de base assiste dans toutes les phases du processus de création de la solution d'automatisation, La conception de l'interface utilisateur du logiciel STEP 7 répond aux connaissances ergonomiques modernes et son apprentissage est très facile [10].

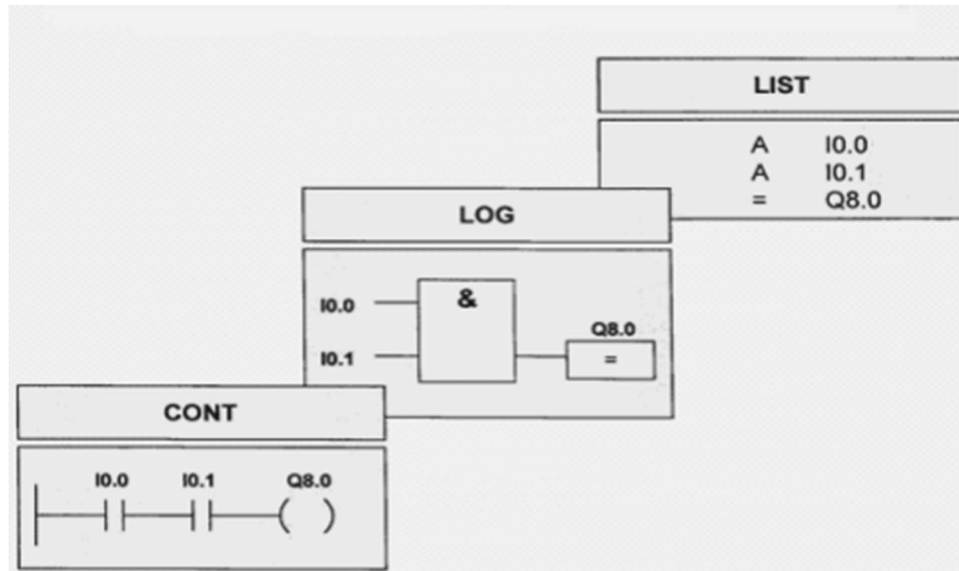
STEP 7 comporte quater sous logiciels de base suivants :

### II.3 Gestionnaire de projets SIMATIC Manager

SIMATIC Manager constitue l'interface d'accès à la configuration et à la programmation. Ce gestionnaire de projets présente le programme principal du logiciel STEP7 il gère toutes les données relatives à un projet d'automatisation, quel que soit le système cible (S7/M7/C7) sur lequel elles ont été créées. Le gestionnaire de projets SIMATIC démarre automatiquement les applications requises pour le traitement des données sélectionnées.

## II.4 Editeur de programme et les langages de programmation

Les langages de programmation CONT, LIST et LOG, font partie intégrante du logiciel de base.



**Figure II.1:** mode de représentation des langages basiques de programmation STEP 7

- **Langage CONT :** C'est un langage à contacts, les entrées sont représentées par des interrupteurs et les sorties par des bobines ou des bascules, c'est le plus utilisé par les automaticiens.
- **Langage LOG :** Les langages logigramme est un langage graphique, utilisant les symboles de l'électronique numérique comme les portes logiques.
- **Langage LIST :** Se présente sous forme d'une List d'instruction est correspond à peu près à l'assembleur dans un ordinateur.
- **Langage GRAFCET :** Le GRAFCET est un outil graphique pour la description des applications de commande et utilisé généralement pour spécifier et concevoir les comportements souhaités de la partie commandé.

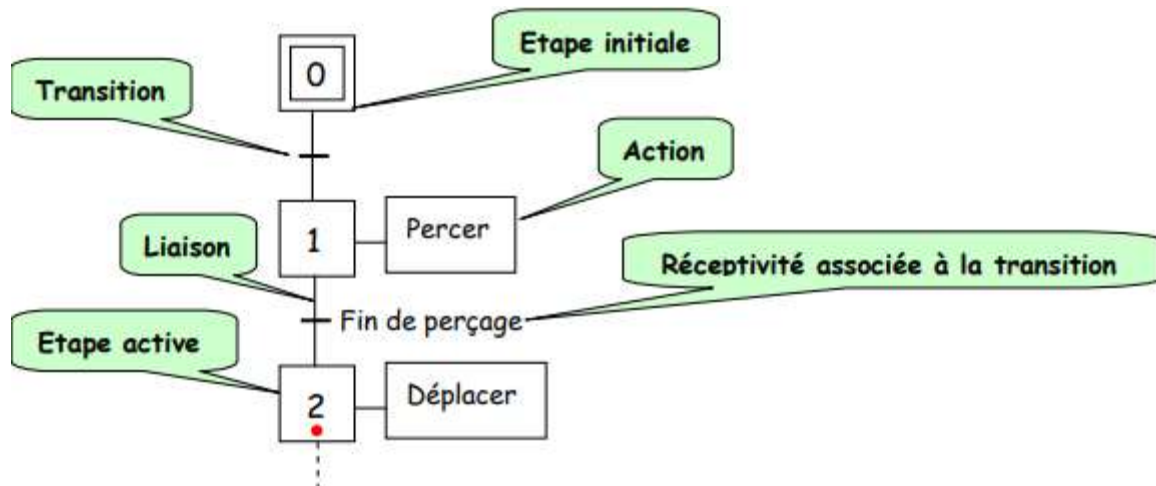


Figure II.2:L'élément de GRAFCET

### II.5 Interface de communication PC- APC

En fait, l'interface MPI (Multi Point Interface) est une interface de communication intégrée à chaque automate programmable SIMATIC. Elle permet de communiquer l'automate avec le PC via USB



Figure II. 3 : Interface de communication PC- APC

### II.6 Le simulateur des programmes PLCSIM

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme dans un automate programmable (AP) qu'on simule dans un ordinateur ou dans

une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). L'AP S7 de simulation permet de tester des programmes destinés aux CPU S7-300 et aux CPU S7-400, et de remédier à d'éventuelles erreurs [10].

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). Tout en exécutant le programme dans l'AP de simulation, on a également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

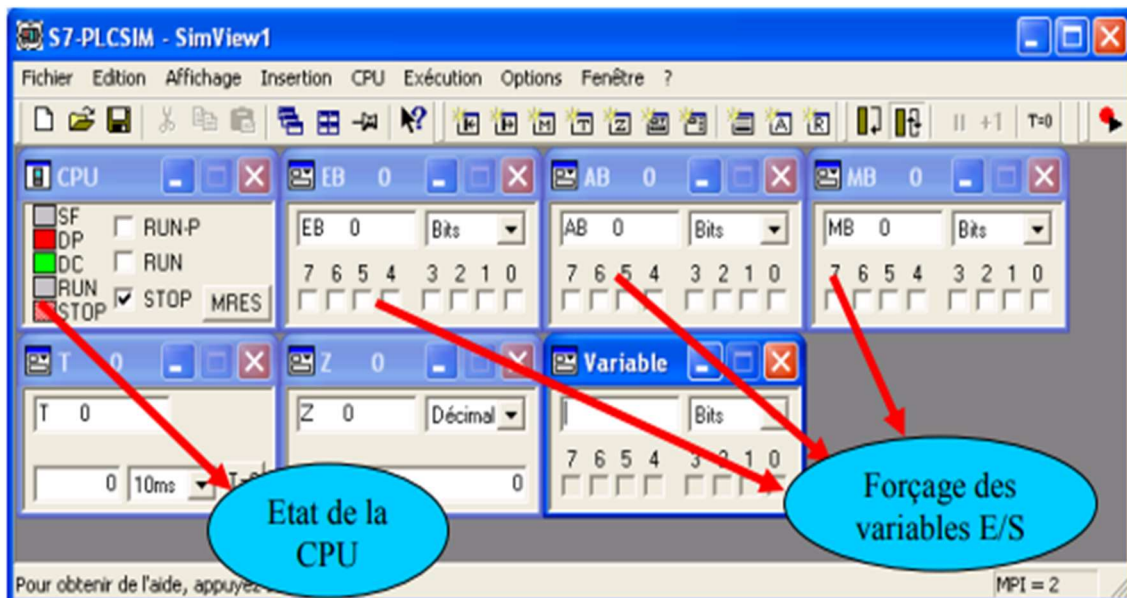


Figure II .4 : logiciel de simulation PLC-SIM

## II.7 Stratégie pour la conception d'une structure programme complète et optimisée

La mise en place d'une solution d'automatisation avec STEP 7 nécessite la réalisation des tâches fondamentales suivantes :

### II.7.1 Création du projet SIMATIC Step7

- **Configuration matérielle HW Config** : Dans une table de configuration, on définit les modules mis en œuvre dans la solution d'automatisation ainsi que les adresses permettant d'y accéder depuis le programme utilisateur, pouvant en outre, y paramétrer les caractéristiques des modules.
- **Définition des mnémoniques** : Dans une table des mnémoniques, on remplace des adresses par des mnémoniques locales ou globales de désignation plus évocatrice afin de les utiliser dans le programme.
- **Création du programme utilisateur** : En utilisant l'un des langages de programmation mis à disposition, on crée un programme affecté ou non à un module, qu'on enregistre sous forme de blocs, de sources ou de diagrammes

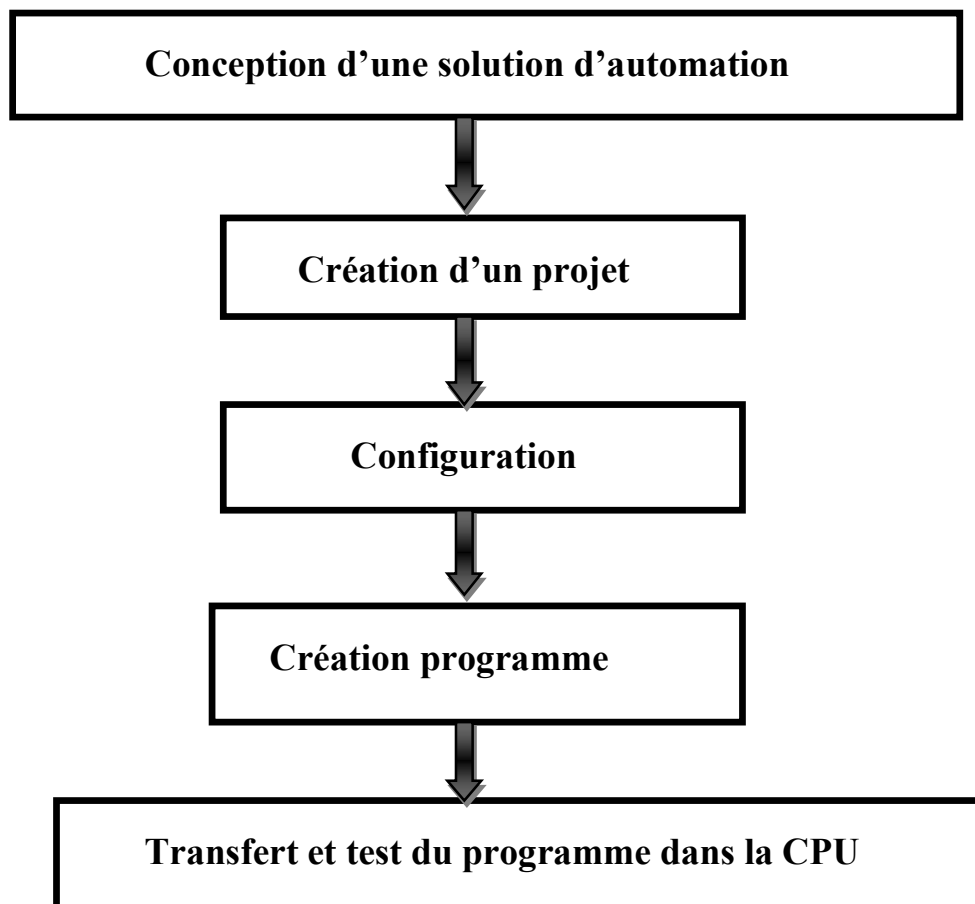


Figure II.5 : Création projet STEP7

## II.8 Création d'un projet STEP 7

Un projet comprend deux données essentielles, les programmes et la configuration du matériel. On peut commencer par définir l'une ou l'autre, mais tout d'abord, il faut démarrer le programme SIMATIC Manager. Ce programme, est l'interface graphique qui permet la manipulation du projet et l'accès aux autres programmes de STEP7.

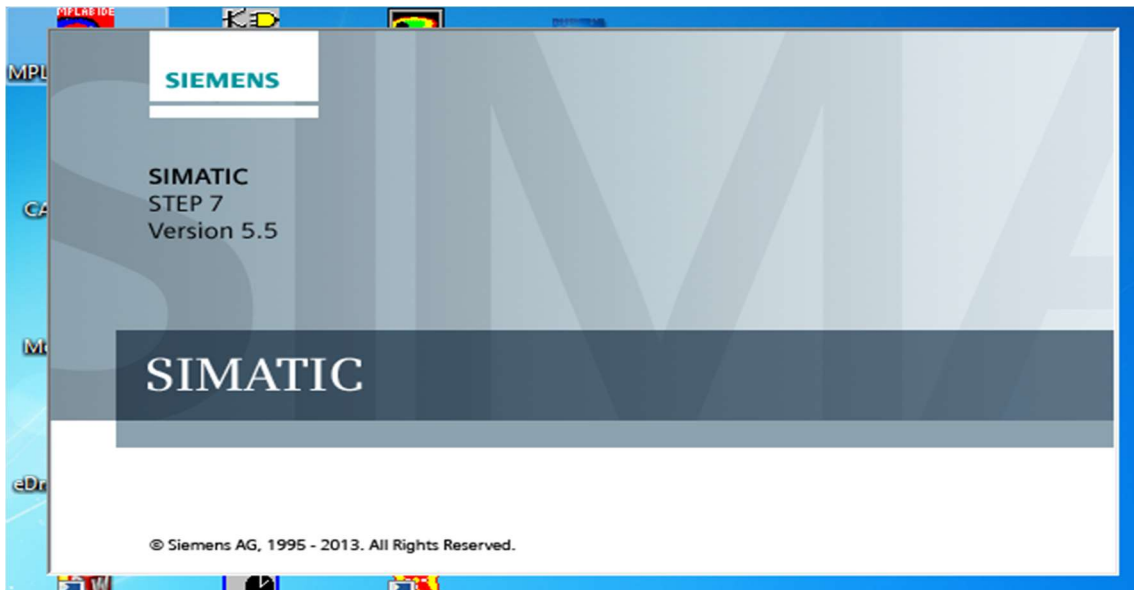


Figure II .6: SIMATIC Manager.

En sélectionnant l'icône SIMATIC Manager, on aura la fenêtre principale qui s'affiche, pour sélectionner un nouveau projet et valider.

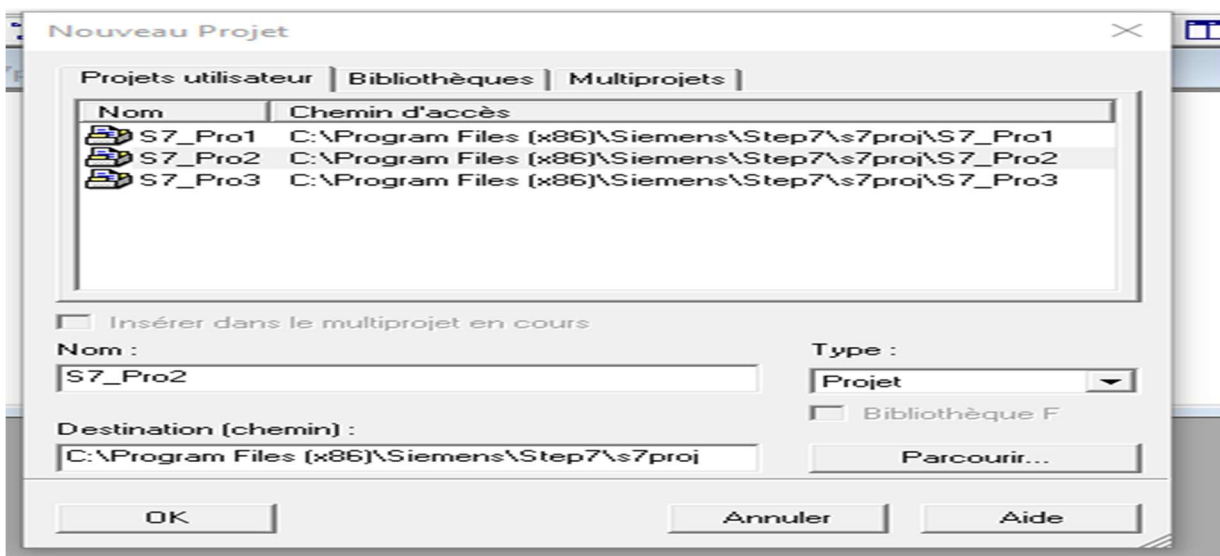
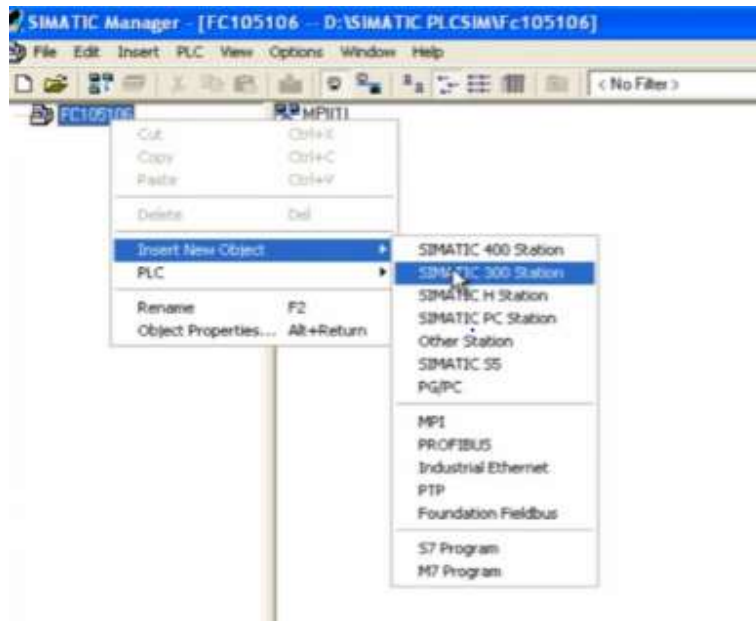


Figure II .07: Choisir un nom de projet

Comme le projet est vide il nous faut insérer une station SIMATIC 300/400.

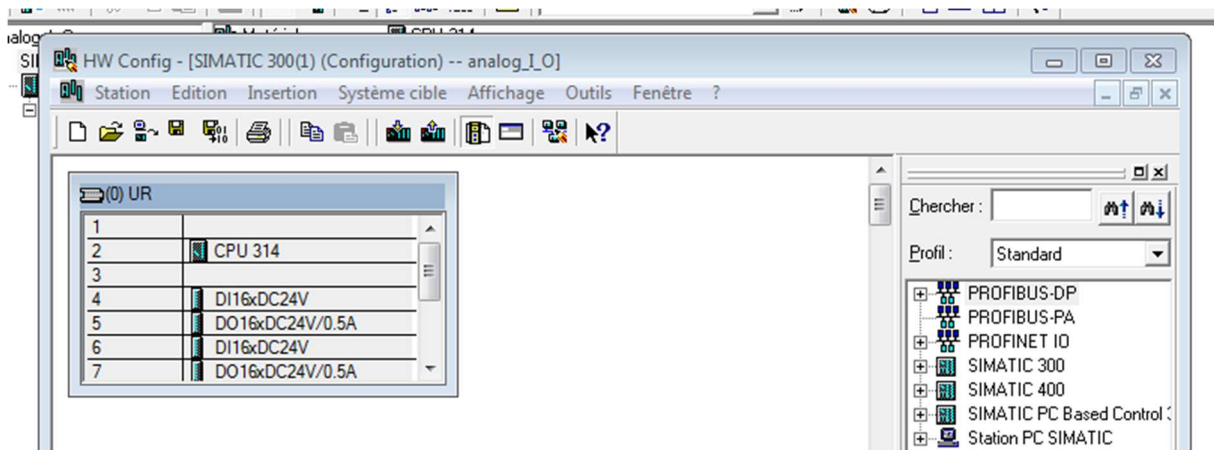


**Figure II .8:** Choix de API SIMATIC -300

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

## II.9 Configuration matérielle (Partie Hardware)

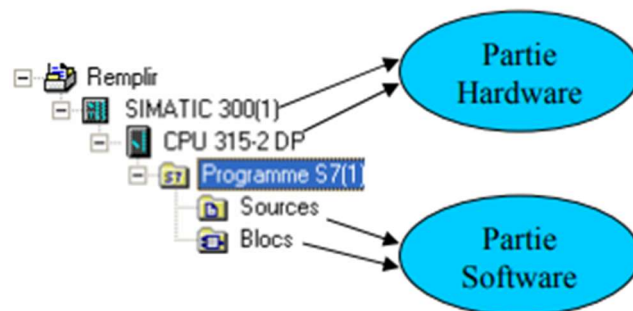
C'est une étape importante, qui correspond à l'agencement des châssis, des modules et de la périphérie décentralisée. Les modules sont fournis avec des paramètres définis par défaut en usine. Une configuration matérielle est nécessaire pour : • modifier les paramètres ou les adresses pré-réglés d'un module, • configurer les liaisons de communication. Le choix du matériel SIMATIC S300 avec une CPU315-2DP nous conduit à introduire hiérarchisée suivante :



**Figure II .09 :** Configuration matérielle

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.10:** Création du programme S7

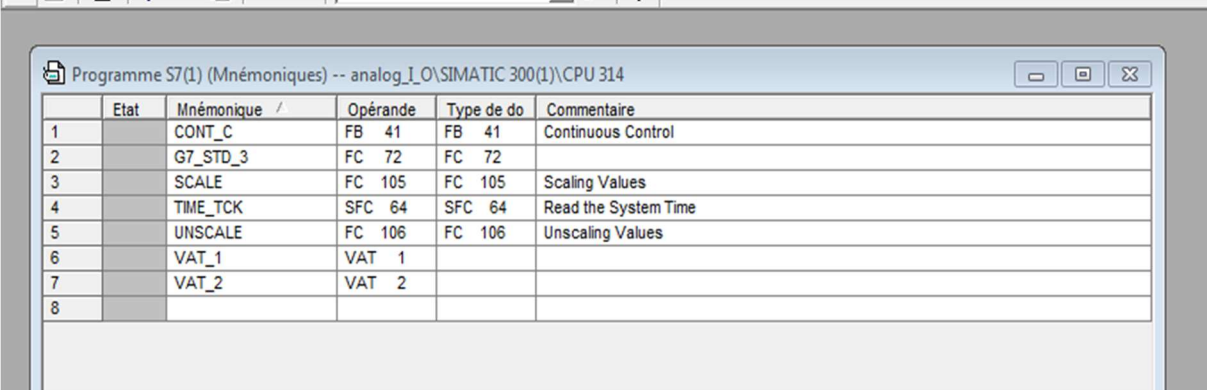
## II.10 Création de la table des mnémoniques (Partie Software)

Dans tout programme il faut définir la liste des variables qui vont être utilisées lors de notre programmation. Pour cela la table des mnémoniques est créée. L'utilisation des noms appropriés rend le programme plus compréhensible est plus facile à manipuler. Ce type d'adressage est appelé « relatif »

Pour créer cette table, on suit le cheminement suivant :

**Insérer nouvel objet**  **table des mnémoniques**

On édite la table des mnémoniques en respectant notre cahier de charge, pour les entrées et les sorties.



	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 .11:** Table de mnémonique

## II.11 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.

### a. 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 [11].

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

### b. Les blocs fonctionnels (FB), (SFB)

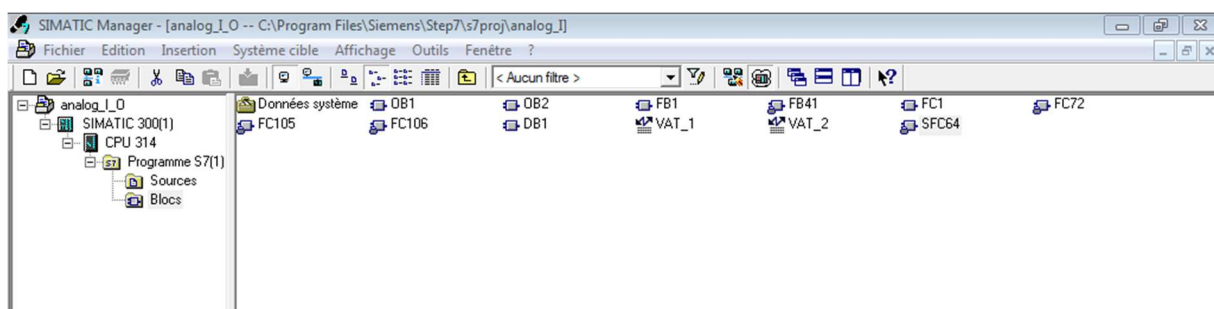
Le FB est un sous-programme écrit par l'utilisateur et exécuté par des blocs de code. On lui associe un bloc de données d'instance relatif à sa mémoire et contenant ses paramètres. Les SFB système sont utilisés pour des fonctions spéciales intégrées dans la CPU [11].

### c. La fonction (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 [11].

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

Dans notre exemple on a utilisé le FC105, FC106, FC1 et FC72 pour les routines, et FB1 pour la partie séquentielle. Systématiquement et automatiquement, les SFC correspondantes ont été créés.



**Figure II .12 :** blocs programme dans SIMATIC Manager

Une fois les mnémoniques et les blocs déclarés, on est prêt pour écrire le programme.

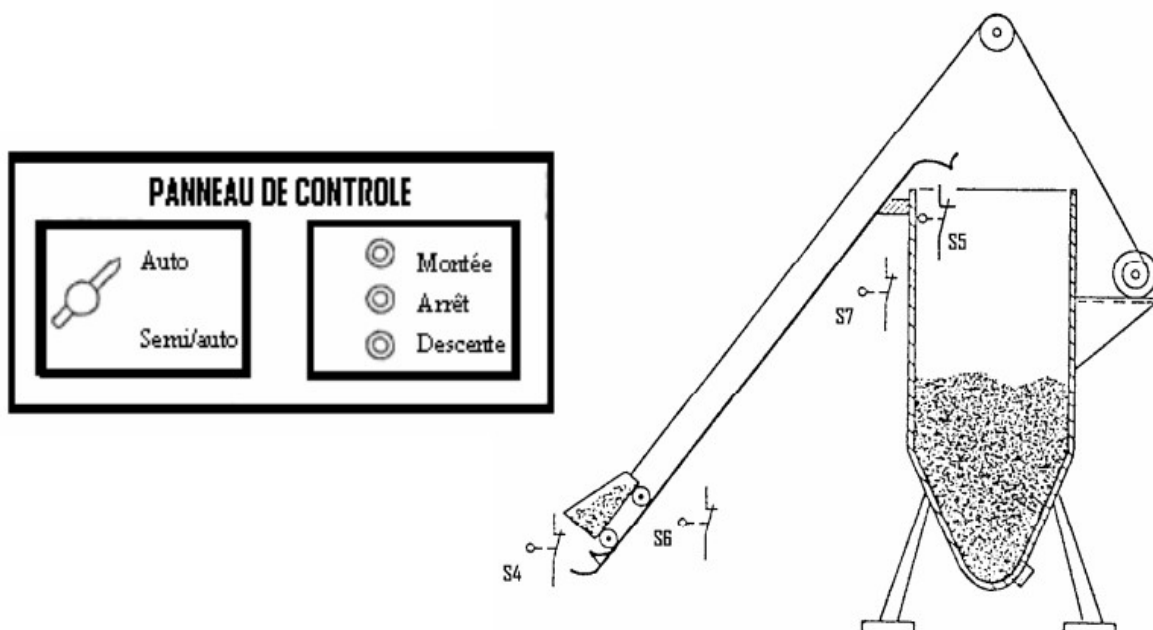
## II .12 Exemple d'application de Création et d'édition d'un projet S7

### ‘Déplace un chariot sur un plan incliné ‘

Pour illustrer ce qu'on a vu précédemment, on va mettre en œuvre une application d'entraînement d'un moteur asynchrone, qui déplace un chariot sur un plan incliné. Le moteur asynchrone à courant alternatif va être commandé par le variateur de vitesse Schneider.

Une station SIMATIC S7-300 (CPU314) avec des modules entrées sorties numériques intégrée va être utiliser pour la gestion du « Monte/descente-chariot ». Le cahier de charge

Voici le schéma du « Monte/descente-chariot » et son panneau de contrôle :



**Figure II.13** : Schéma de principe de l'application

Le chariot se déplace jusqu'à rencontrer le capteur de fin de course (S4, resp, S5). En appuyant sur la touche Arrêt, le déplacement du chariot est stoppé.

Si le commutateur automatique / semi-automatique est sur "Semi-automatique", alors le chariot monte (descend) tant que l'utilisateur appuie sur le bouton "Montée" ("Descente").

Dans la phase de montée, le chariot accélère jusqu'à la vitesse requise. Quand le chariot atteint le capteur de limite S7, il entre dans la phase de freinage. Le chariot sera donc freiné

jusqu'à la vitesse faible exigée et continuera à cette même vitesse jusqu'à atteindre le capteur de fin de course S5.

Dans la phase de descente, le chariot accélère jusqu'à sa vitesse maximale. Quand le chariot atteint le capteur de limite S6, alors le chariot est freiné et continue à cette basse vitesse jusqu'au capteur de fin de course S4.

Voici les normes à respecter pour cela :

- Détecteur de position S7 actif : Le moteur freine et atteint la fréquence requise de  $F3=5\text{Hz}$  en 7,5 secondes.
- Détecteur de position S5 actif : Le moteur s'arrête en 3 secondes environ.  
Descendre le chariot : La durée pour atteindre le régime haut de fréquence (50Hz) doit valoir 10 secondes.

A durée pour revenir de la fréquence (50Hz) au repos doit valoir 15 secondes :

- Le moteur doit s'arrêter en 3 secondes (pour une fréquence de 5Hz).
- Le chariot sera remonté conformément à la fréquence  $F1 = 30\text{Hz}$ .
- Le chariot sera redescendu conformément à la fréquence  $F2 = 50\text{Hz}$ .
- Lors de la phase de décélération, la nouvelle consigne devient  $F3 = 5\text{Hz}$ .

En détaillant chaque phase comme suit .

Monter le chariot :

- Actionnez « Montée » : Le moteur démarre et atteint la fréquence requise  $F1=30\text{Hz}$  en 6secondes.
- Actionnez « Descendre » : Le moteur démarre et atteint la fréquence max.  $F2 = 50\text{Hz}$  en 10secondes.
- Détecteur de position S6 actif : Le moteur freine en 13,5 secondes jusqu'à la fréquence de 5Hz.
- Détecteur de position S4 actif : Le moteur s'arrête au bout d'environ 3 secondes.

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

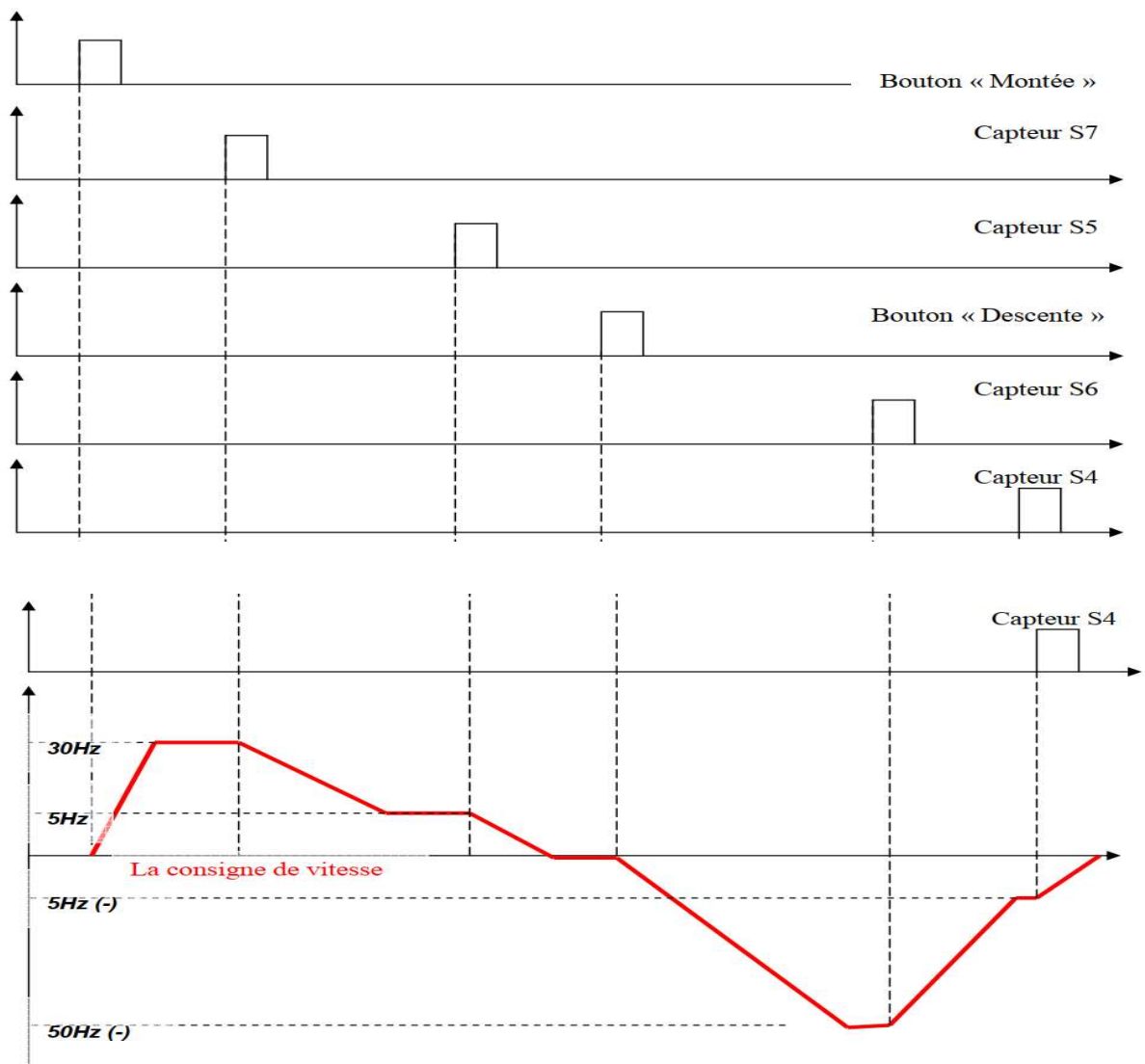


Figure II.14 : Chronogramme de fonctionnement

### II.12.1 La mise en œuvre de l'application

Tout au long de cette application, la méthodologie est représentée dans l'organigramme suivant :

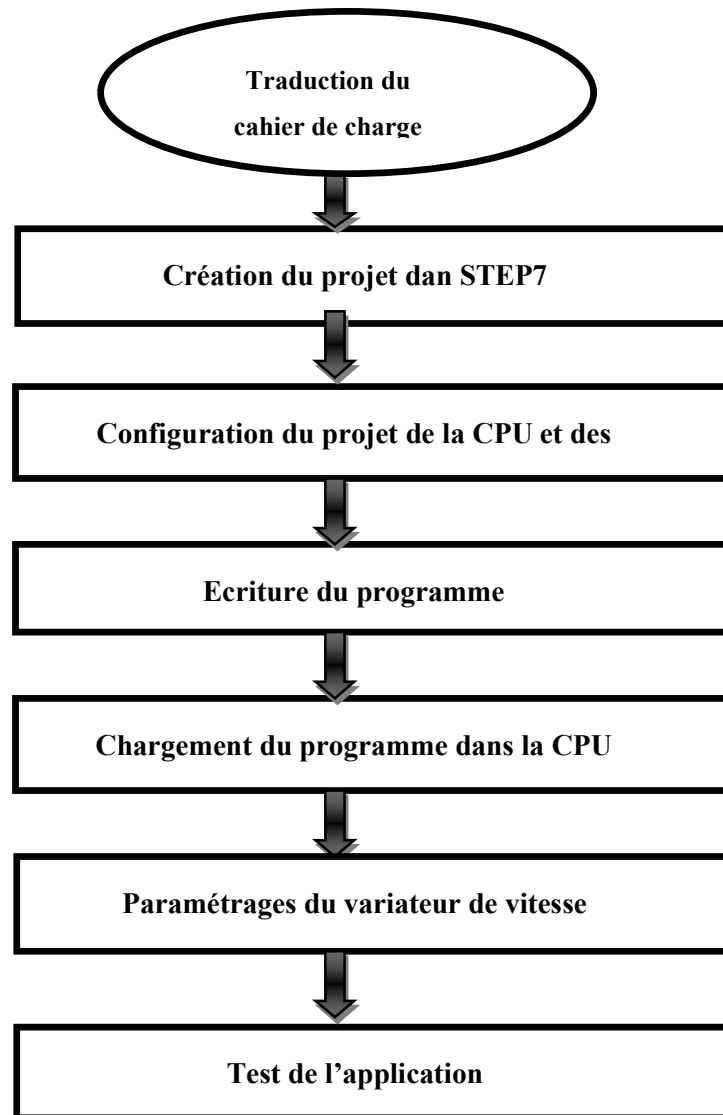


Figure II. 15 : l'organigramme de l'application du chariot

### II.12.2 Description les éléments de l'installation pour l'application

Pour cette application, les éléments du réseau sont les suivant :

1 PC, système d'exploitation Windows XP.

2 Progiciel STEP7 V 5.5.

3 Interface ordinateur MPI (Ex : PC- Adapter) pour configurer l'automate.

4 API SIMATIC S7-300 avec la CPU 314, pour cette application.

5 Variateur de vitesse.

6 Moteur asynchrone.

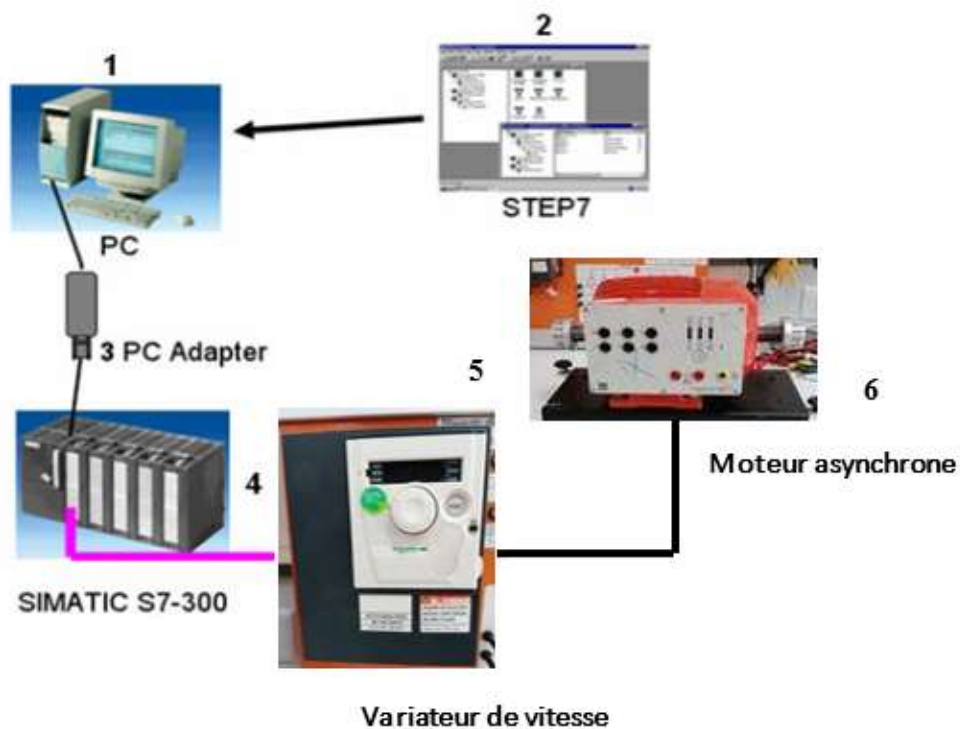


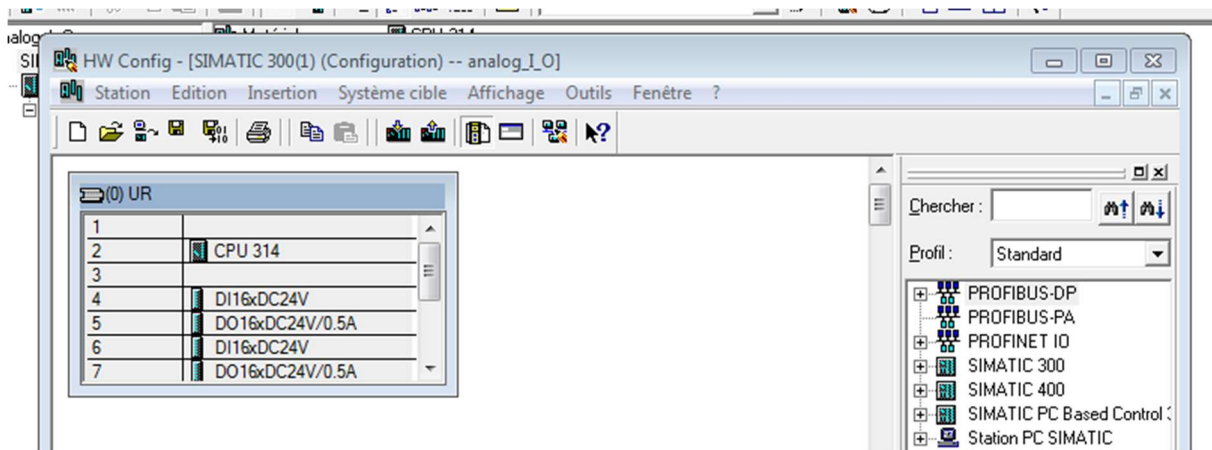
Figure II .16: Schéma de principe de l'application

### II.12.3 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 I.17 :** Configuration matériel pour la CPU314 et ces modules E/S

**b. Conception et écriture du Programme**

Dans cette partie, nous allons attribuer des adresses absolues dans la CPU pour les entrées du processus, comme indiqué dans le tableau suivant :

Entrées (Boutons/ capteurs)	Adresse absolue
Auto/Manuel	E 0.0
Montée	E 0.1
Descente	E 0.2
STOP	E 0.3
S4	E 0.4
S5	E 0.5
S6	E 0.6
S7	E 0.7
Affichage	AW 12

**Tableau II. 01 :** Adressage des entrées/sorties processus

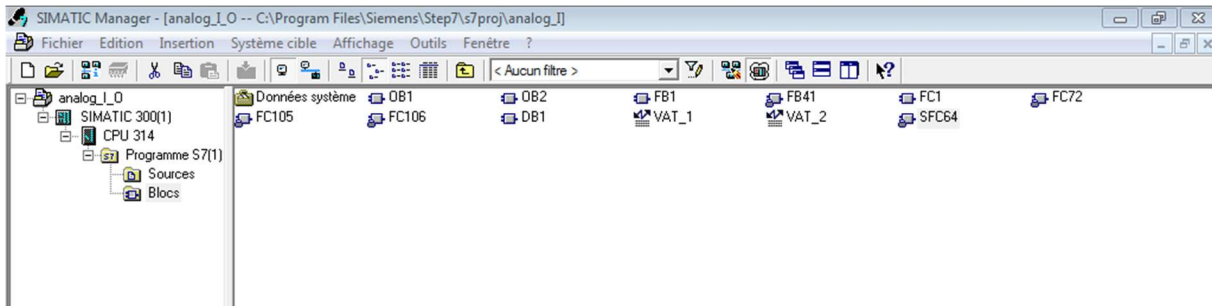


Figure II.18 : blocs programme dans SIMATIC Manager

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

	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.19 :Table de mnémonique

### II.13.4 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.

#### a. 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 [11].

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

### b. Les blocs fonctionnels (FB), (SFB)

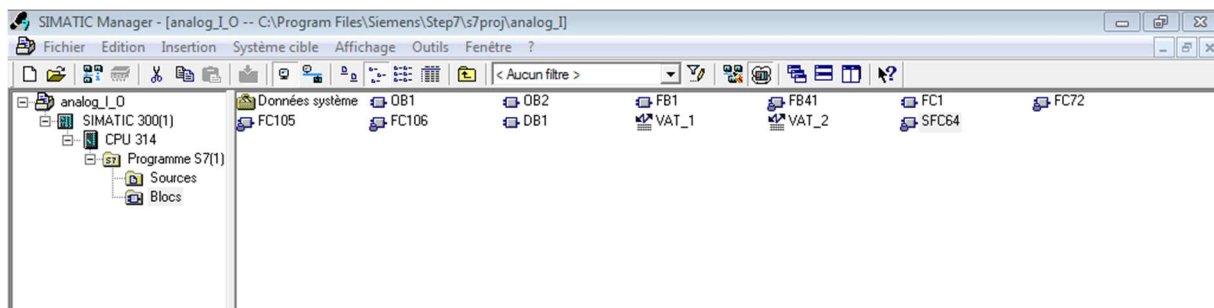
Le FB est un sous-programme écrit par l'utilisateur et exécuté par des blocs de code. On lui associe un bloc de données d'instance relatif à sa mémoire et contenant ses paramètres.

Les SFB système sont utilisés pour des fonctions spéciales intégrées dans la CPU [8].

### c. Les fonction (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 [8].

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

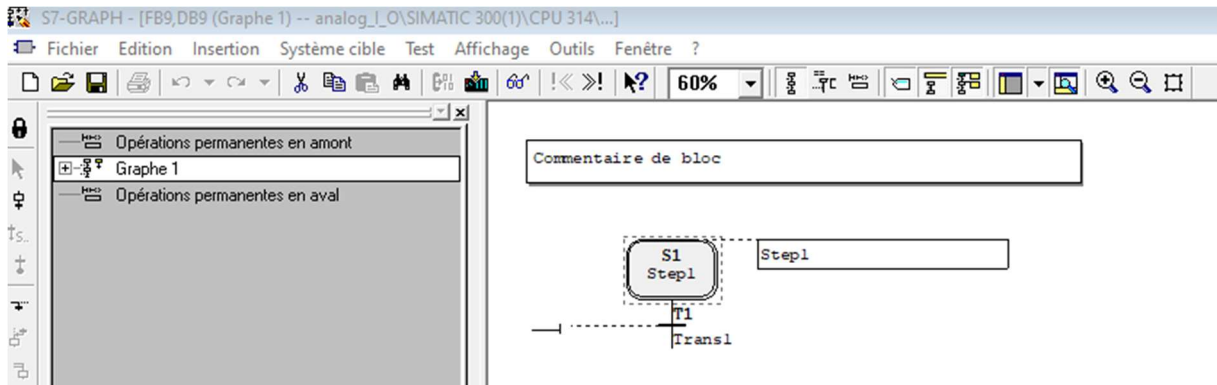


**Figure II.20:** Vue d'ensemble des blocs programme dans SIMATIC Manager

Une fois les mnémoniques et les blocs déclarés, on est prêt pour écrire le programme.

On commence par créer un bloc fonctionnel et un autre donné d'instance dans le dossier Blocs du SIMATIC Manager

On choisit le GRAPH comme langage de programmation et on lance l'application.

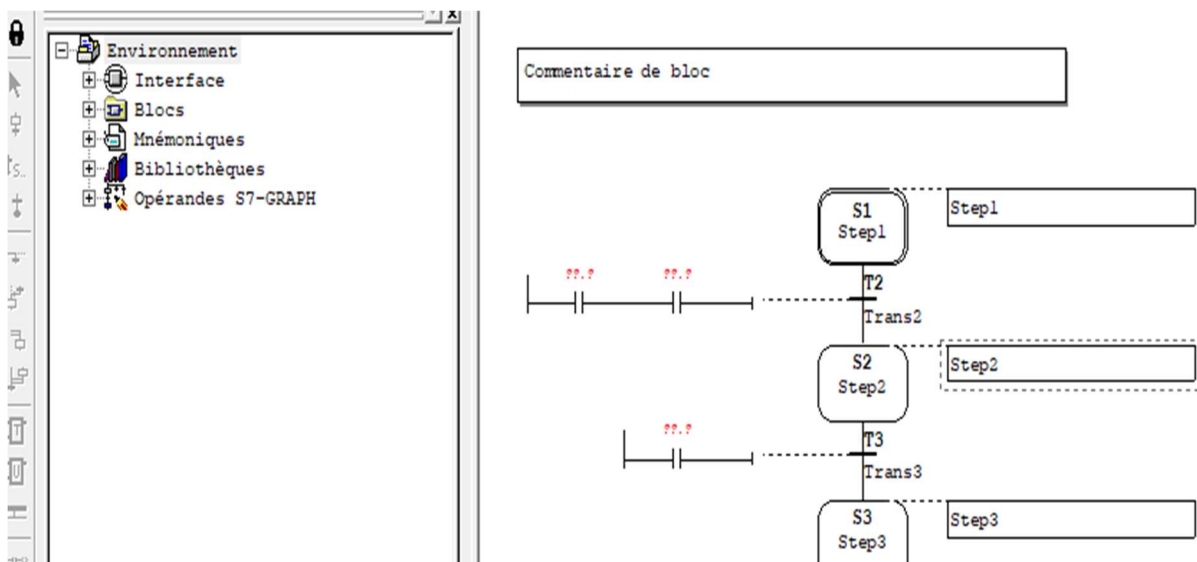


**Figure II. 21 :** édition d’un diagramme fonctionnel avec S7-Graph

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 commande de chariot comprend cinq étapes et quatre transitions. On commence par les créer.



**Figure II. 22:** 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 différents types 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 met à 1 pour la durée de l'activation.
L	A, E, M ou D	Impulsion : quand l'étape est active, l'opérande se met 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.02 :** 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...).

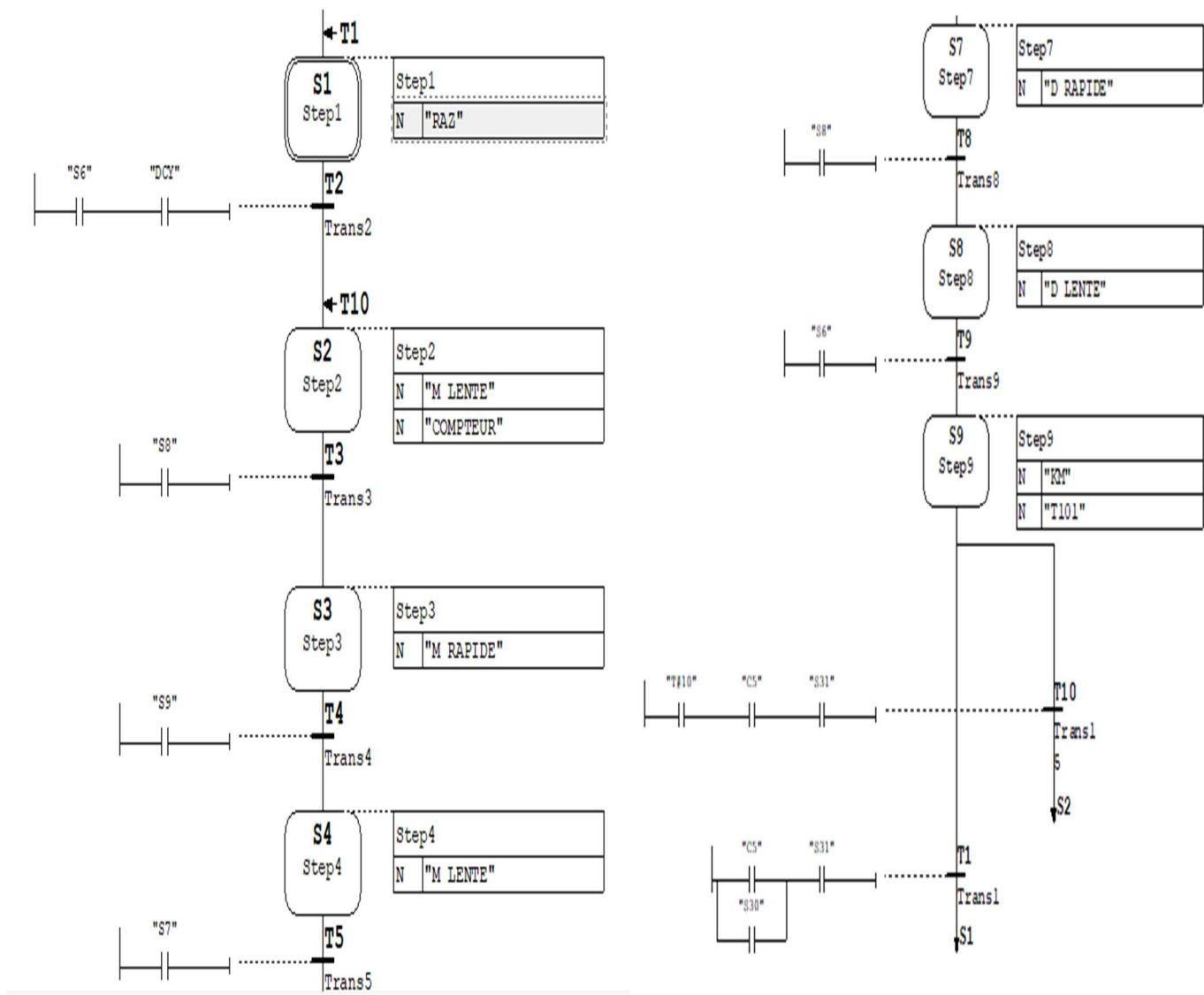


Figure II. 23 : édition des actions& transitions

### II.13.5 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 WinCC.

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).

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.

### **II.14 Conclusion**

On a constaté la facilité et la souplesse qu'offre L'A.P. I pour sa programmation, connexion, et adaptation aux conditions industrielles, avec toutes les fonctionnalités à l'automatisation des processus. Les résultats et tests obtenus répondent bien à l'objectif de commande, la simplicité et la performance pour de telles installations. L'automatisation, le traitement de l'information ont été traités durant ce projet.

***Chapitre III :***

***Le Progiciel De Conception Des  
Interfaces Homme-Machine***

### **III. Le Progiciel De Conception Des Interfaces Homme-Machine**

#### **III.1 Introduction**

Le logiciel de configuration et de contrôle industriel Win CC, fourni par Siemens, est utilisé comme un système d'acquisition de données et de surveillance en temps réel, ainsi qu'une interface homme-machine [12]. Il offre aux opérateurs une assistance intuitive, englobant une diversité de produits et de systèmes standardisés, flexibles et échelonnables, tels que la surveillance des processus industriels

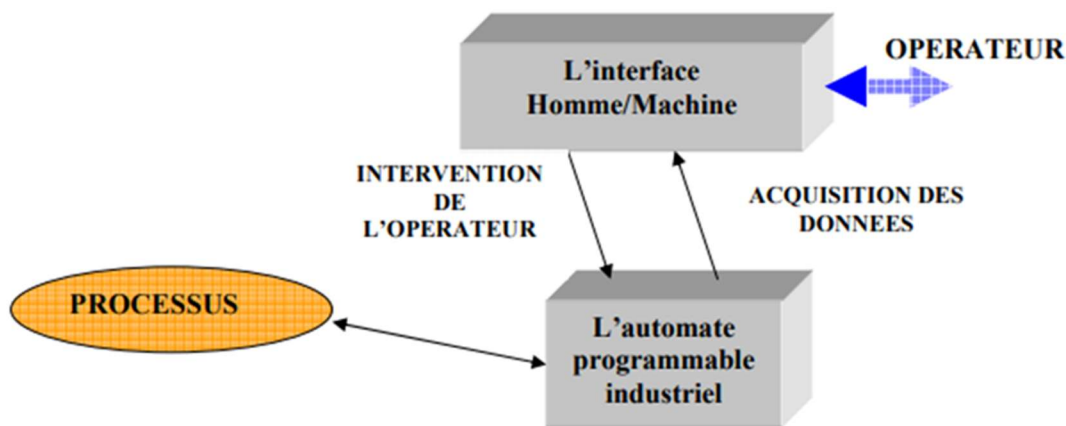
#### **III.2 Interface homme-machine**

L'évolution de l'industrie a engendré des besoins croissants en matière d'exploitation des installations industrielles, passant des schémas synoptiques aux pupitres de contrôle et de commande, puis aux écrans tactiles dont l'objectif principal est de faciliter au maximum la supervision et l'exploitation des équipements industriels.

Dans notre étude, nous nous intéresserons aux HMI dédiées à la supervision des systèmes industriels.

#### **III.2 Description générale de SIMATIC HMI**

SIMATIC HMI propose une gamme complète qui permet de répondre à toutes les tâches de contrôle-commande. SIMATIC HMI permet de surveiller en permanence le processus et de maintenir les machines et les installations en état de fonctionnement. Un système IHM constitue l'interface entre l'opérateur et le processus de la machine ou de l'installation. Le contrôle réel du processus est assuré par le système d'automatisation. Il existe donc une interface entre l'opérateur et Win CC flexible (sur le pupitre opérateur) et une interface entre Win CC flexible et le système d'automatisation, toujours via l'automate



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

Un système IHM prend en charge les tâches suivantes [13] :

- Représentation du processus : le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue, par exemple, l'affichage sur le pupitre opérateur est mis à jour.
- Commande du processus : l'opérateur peut commander le processus via l'interface utilisateur graphique. Par exemple, il peut définir une consigne pour un automate ou démarrer un moteur.
- Affichage des alarmes : lorsqu'il y a des états critiques dans le processus, une alarme est déclenchée immédiatement, par exemple, lorsque la valeur limite est dépassée.
- Archivage des valeurs de processus et des alarmes : les alarmes et les valeurs de processus peuvent être archivées par le système IHM. Cela nous permet de documenter le déroulement du processus et d'accéder ultérieurement aux données de production passées.

### III.3. Eléments de WinCC flexible

On commence par l'environnement de cet outil, pour terminer et expliquer les différentes étapes dans les détails précédemment énumérés. L'environnement de travail de Win CC flexible se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque cet éditeur est activé. Il met à disposition un éditeur spécifique pour chaque tâche de configuration. On peut configurer par exemple, l'interface utilisateur graphique d'un pupitre opérateur avec l'éditeur "Vues". Pour la configuration d'alarmes, on peut utiliser p. ex. l'éditeur "Alarmes TOR" ou autre. Dans Win CC flexible, chaque projet créé contient principalement des vues que l'on crée pour le contrôle-commande de machines et d'installations. Lors de la création des vues, vous disposez d'objets prédéfinis

permettant d'afficher des procédures et de définir des valeurs du processus. Les différents outils et barres de l'éditeur de vues sont représentés dans la figure qui suit :

- **Barre de menus** : La barre de menus regroupe toutes les commandes nécessaires à l'utilisation de WinCC flexible. Des raccourcis sont également disponibles à côté de chaque commande du menu.
- **Barres d'outils** : Les barres d'outils affichent les outils et les fonctionnalités indispensables pour les programmeurs.
- **Zone de travail** : La zone de travail permet de configurer les vues de manière à les rendre compréhensibles et faciles à manipuler pour les utilisateurs, facilitant ainsi la consultation des résultats.
- **Boîte à outils** : La fenêtre des outils propose une sélection d'objets simples ou complexes qui peuvent être insérés dans les vues, tels que des éléments graphiques et des éléments de commande.
- **Fenêtre des propriétés** : Le contenu de la fenêtre des propriétés dépend de la sélection en cours dans la zone de travail. Lorsqu'un objet est sélectionné, ses propriétés peuvent être modifiées dans la fenêtre des propriétés [13].

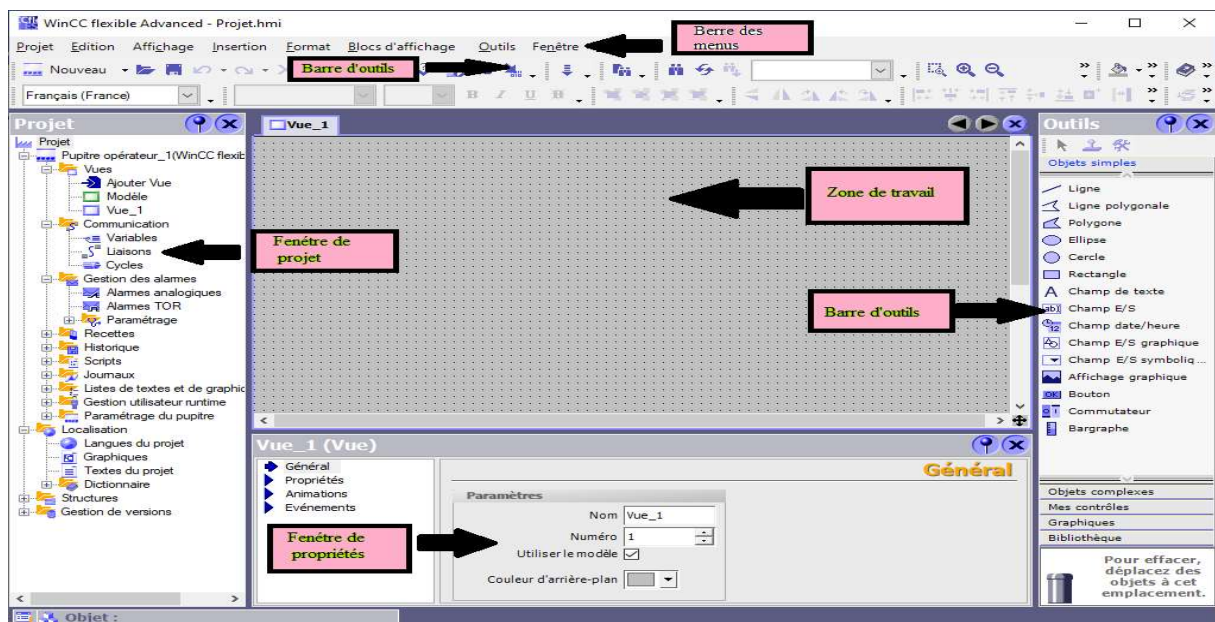


Figure .III.2 : La fenêtre principale de WinCC flexible.

### III.4 Mise en service un projet de WinCC flexible

L'utilisation de ce logiciel pour la conception d'une interface, passe par plusieurs étapes résumées dans le projet de WinCC flexible :

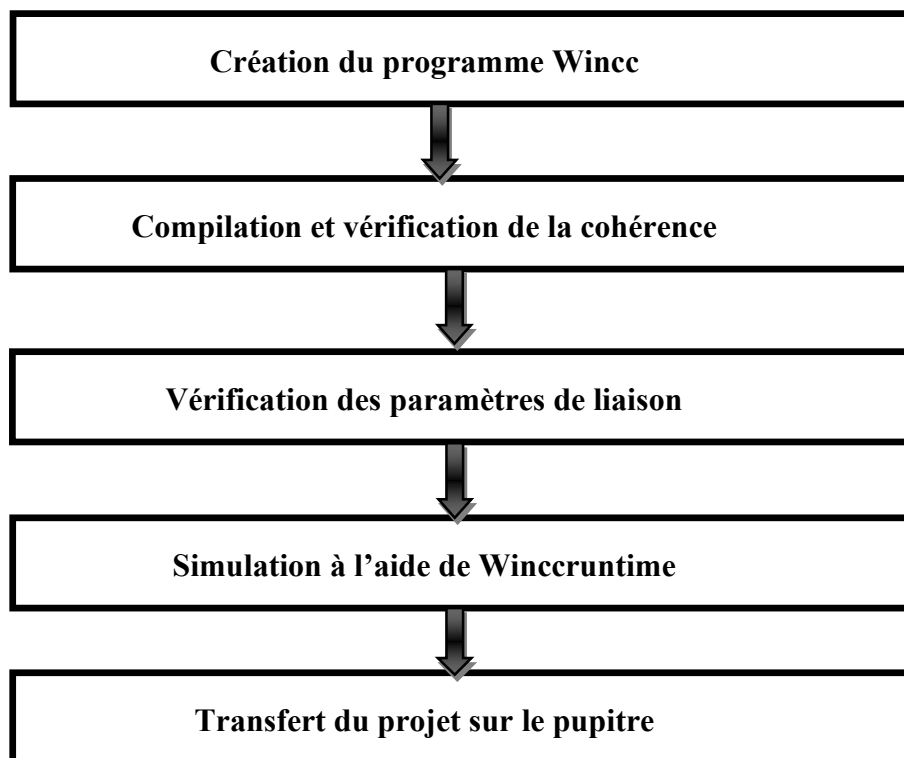


Figure III.3 : Etapes pour la conception d'une interface via Win CC flexible

### III.4.1 Création d'un nouveau projet

- Sélectionnez la commande du menu.
- Projet → Nouveau.
- Cliquez sur Aperçu, puis :
- Choisissez "Créer un projet vide".

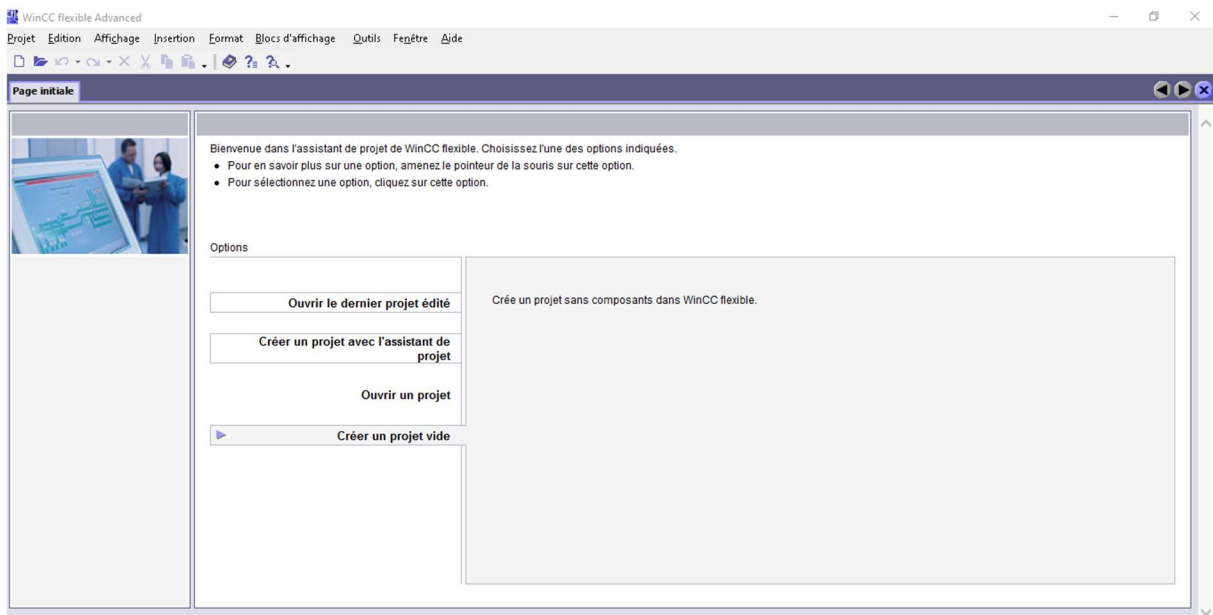


Figure. III.4 :L'interface principale du logiciel WinCC flexible.

### III.4.2 Sélection de pupitre

Après la création d'un projet, nous passons à la sélection du pupitre opérateur pour lequel nous allons créer le projet. Dans la fenêtre du projet, il est possible de modifier le type de pupitre opérateur ou d'ajouter d'autres pupitres. Les propriétés suivantes des vues dépendent du pupitre opérateur choisi [12] :

- Mise en page.
- Résolution de l'écran.
- Profondeur des couleurs.
- Polices de caractères.
- Objets disponibles.

Pour ce projet, nous avons choisi "WinCC flexible Runtime".

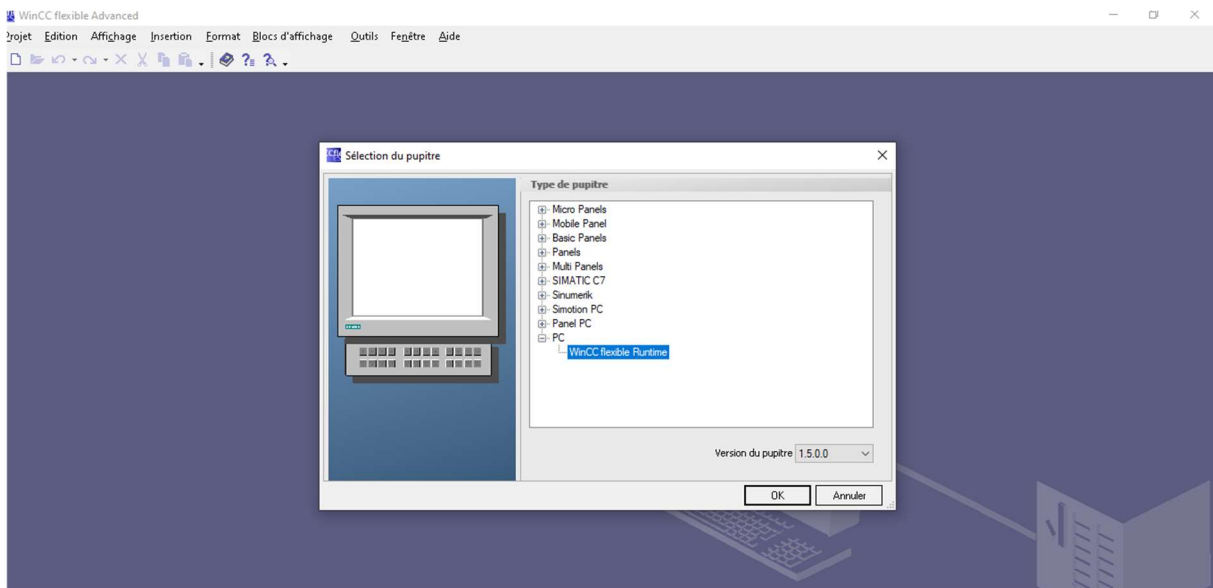


Figure. III.5 : Sélection du pupitre.

### III.4.3 sélections projet WinCC sur STEP7 Manager

Après les deux étapes précédentes, la fenêtre de WinCC flexible est illustrée dans la figure suivante.

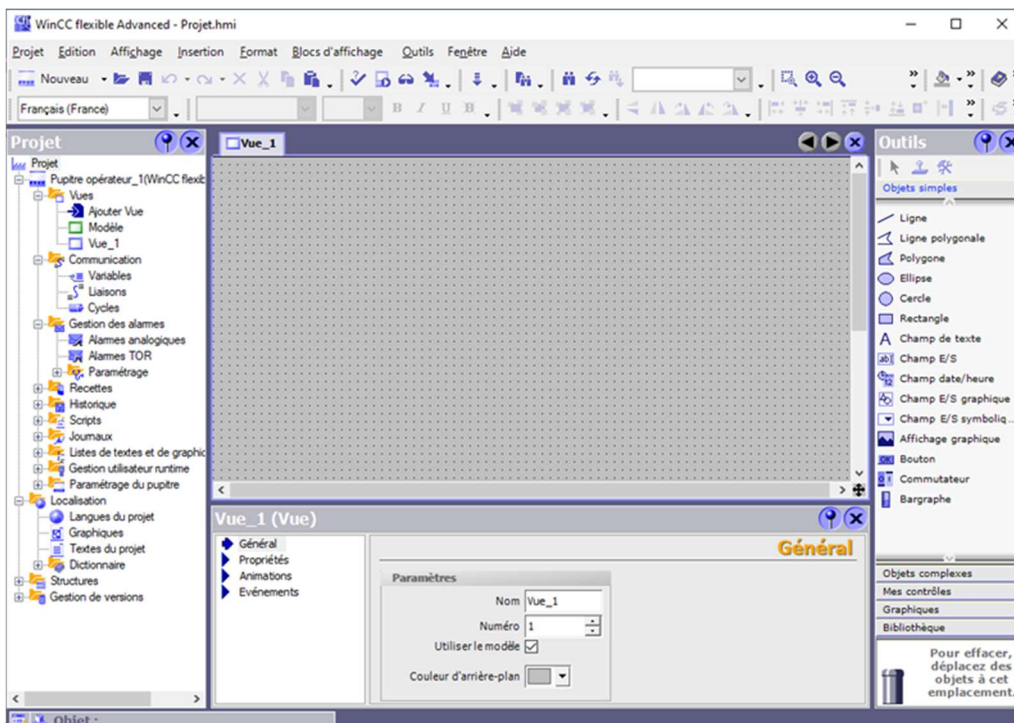


Figure. III.6 : Fenêtre d'une vue dans le WinCC flexible.

On peut accéder aux données de configuration créées lors de la configuration de l'automate. Les avantages sont les suivants :

- Pour l'édition et la gestion des automates SIMATIC, on utilise le gestionnaire MANAGER comme poste central de création.
- Toute modification dans STEP 7 se traduit par une mise à jour des paramètres de communication dans WinCC flexible.
- Les alarmes peuvent être affichées sur le pupitre opérateur.
- L'étape d'intégration n'est pas nécessaire pour créer un projet dans WinCC ; on peut créer un projet WinCC flexible sans intégration dans STEP 7 et l'intégrer ultérieurement dans STEP 7.
- Il suffit de définir les mnémoniques une seule fois dans STEP 7 pour pouvoir les utiliser à la fois dans STEP 7 et dans WinCC flexible.

#### III.4.4 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.

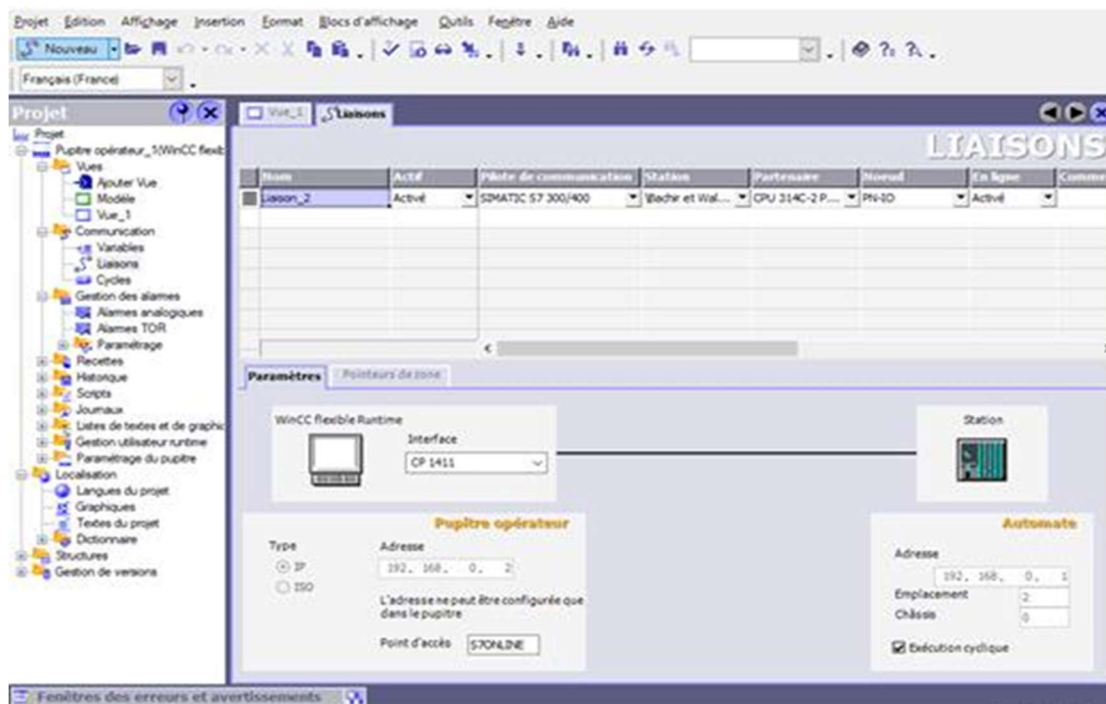


Figure .III.7 :L'éditeur de liaison.

### III.4.5 Utilisation des objets

Les objets sont des éléments graphiques utilisés pour configurer l'apparence des vues de processus dans notre projet. Lors de la création des vues, nous avons à notre disposition des objets prédéfinis qui nous permettent de représenter nos installations, d'afficher des procédures et de définir des valeurs de processus. La boîte à outils regroupe différents types d'objets tels que des objets simples, des objets complexes, des graphiques et une bibliothèque.

Voici quelques exemples des objets les plus couramment utilisés :

- **Polyline** : il s'agit d'une ligne composée de sections reliées les unes aux autres et pouvant contenir un nombre variable de points d'angle. Nous pouvons les modifier ou les supprimer. Nous avons également la possibilité de choisir différents types de terminaisons pour les lignes, tels que des lignes droites, des lignes arrondies ou des flèches.
- **Graphique** : il permet d'afficher des graphiques créés à partir de programmes graphiques externes. Nous pouvons afficher des graphiques dans différents formats tels que "\*.emf", "\*.wmf", "\*.dib", "\*.bmp", "\*.jpg", "\*.jpeg", "\*.gif" et "\*.tif".
- **Boutons** : ils sont utilisés pour contrôler un processus. Nous pouvons configurer l'exécution de fonctions ou de scripts spécifiques associés à chaque bouton.
- **Objets de la bibliothèque** : nous pouvons utiliser des objets préexistants à partir d'une bibliothèque pour faciliter la création de vues et d'interfaces utilisateur.

Ces objets offrent une grande flexibilité et permettent de concevoir des interfaces graphiques conviviales pour le contrôle et la surveillance des processus.

- **Configuration des boutons** : Chaque objet présent dans la vue et le modèle doit avoir un nom unique (Général > Texte > Texte OFF/Texte ON). Il est possible de modifier les propriétés par défaut pour la mise en page, le texte et la représentation des boutons.

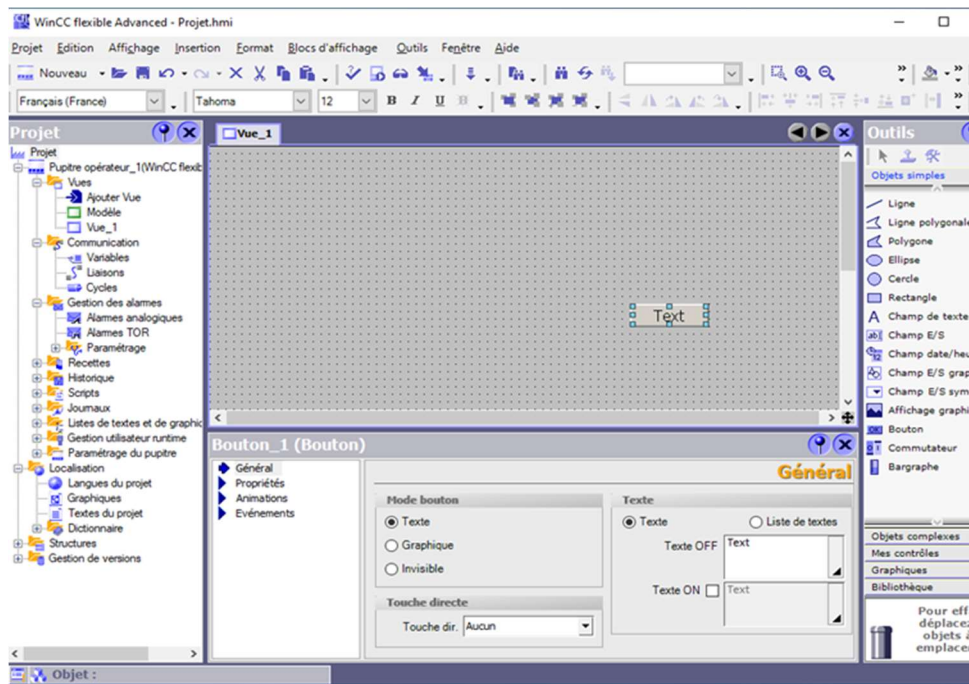


Figure .III.8 : Création d'un bouton.

La fenêtre des propriétés inclut le groupe "Animations" qui permet et de dynamiser la visibilité, la représentation et le mouvement d'un objet en utilisant la variable sélectionnée. Un bouton est une touche du pupitre opérateur à laquelle une fonction configurable est assignée. Une liste de fonctions peut être configurée pour les événements "Presser", "Relâcher" et "Clic" du bouton.

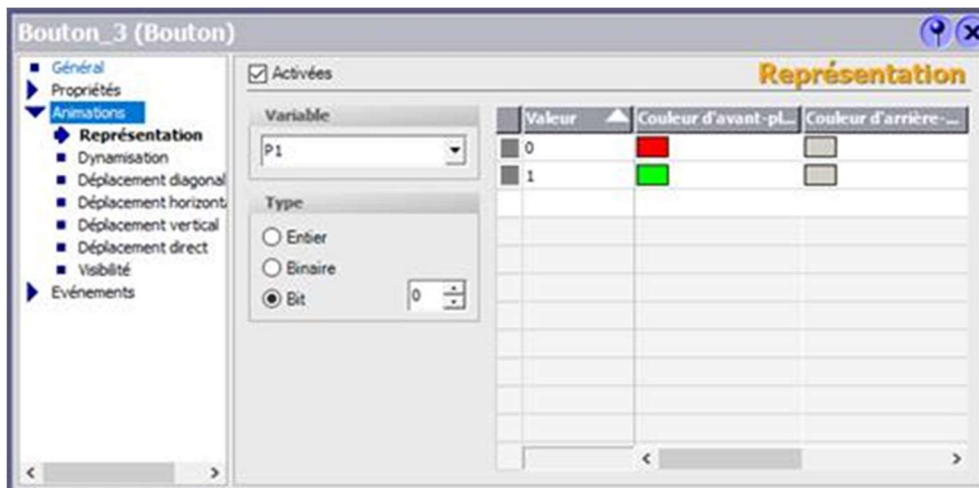
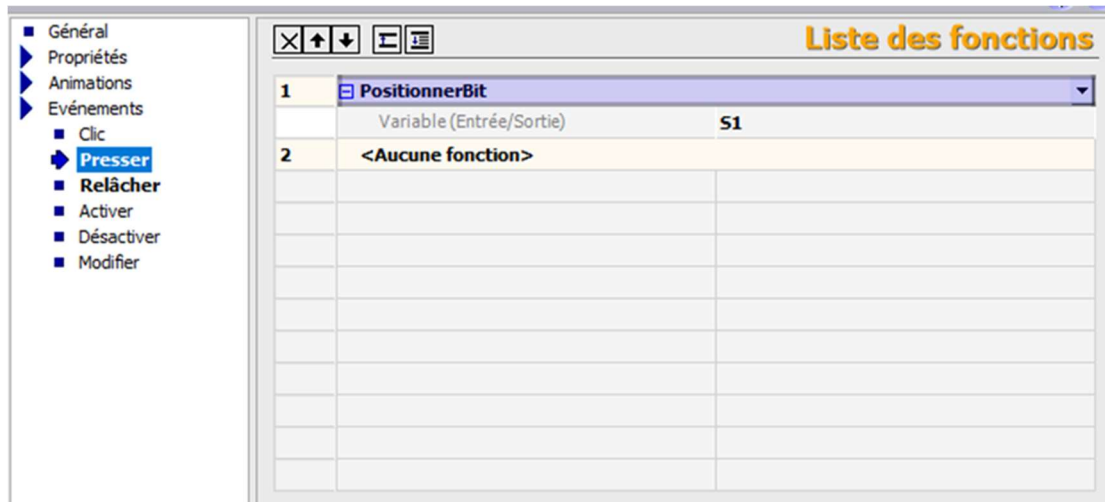


Figure .III.9 : Fenêtre d'outils d'un bouton.

- Configuration des icônes de la bibliothèque :



**Figure. III.10** : La fenêtre d'outils pour les objets de la bibliothèque.

Dans la fenêtre des propriétés, vous pouvez écrire du texte, modifier les propriétés et ajouter des animations. Dans la fonctionnalité des événements, vous pouvez activer la position du bit pour une variable spécifique (0/1) lors de la simulation. Veuillez-vous référer à la figure précédente pour plus de détails.

#### III.4.6 Utilisation des variables

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.11 : éditeur 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.4.7 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 :

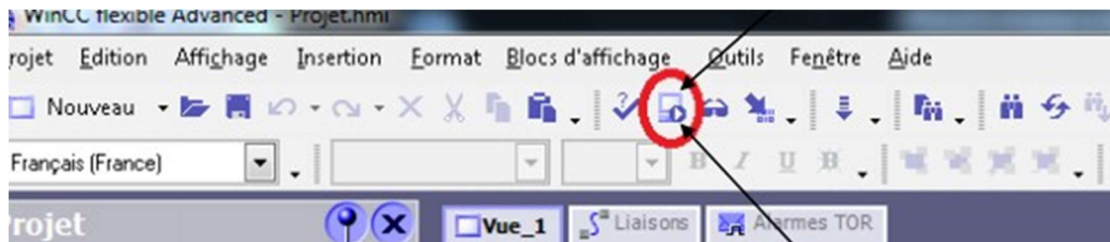


Figure .III.12 : le système Runtime.

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.14 : Supervision par WINCC Runtime

### III.4.8 Conclusion

Dans ce chapitre, nous avons exploré l'ingénierie des interfaces homme-machine (HMI) et nous avons abordé la présentation du logiciel de supervision Win CC flexible ainsi que ses outils simples et efficaces pour la réalisation d'un projet.

***Chapitre IV :***  
***Commande d'un Moteur à***  
***courant contenu***

## IV : Commande d'un moteur à courant continu

### IV.1 Introduction :

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.2 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 :

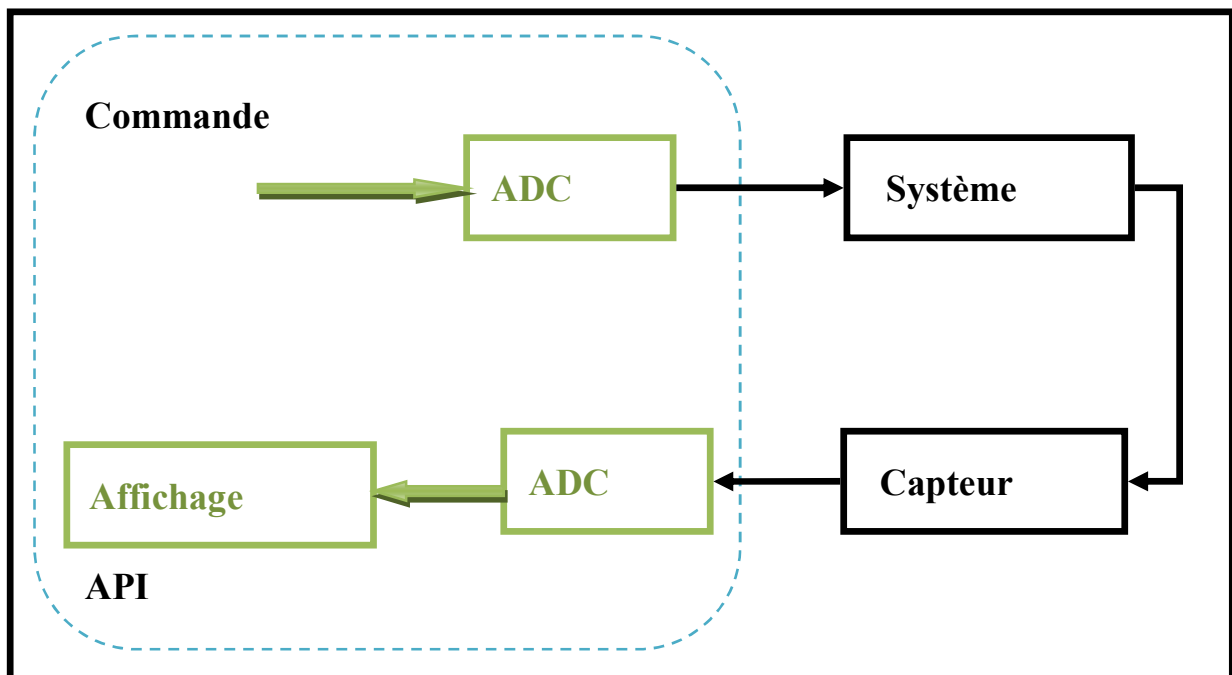


Figure IV.1 : Commande en BO avec boucle de mesure

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 10V.

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

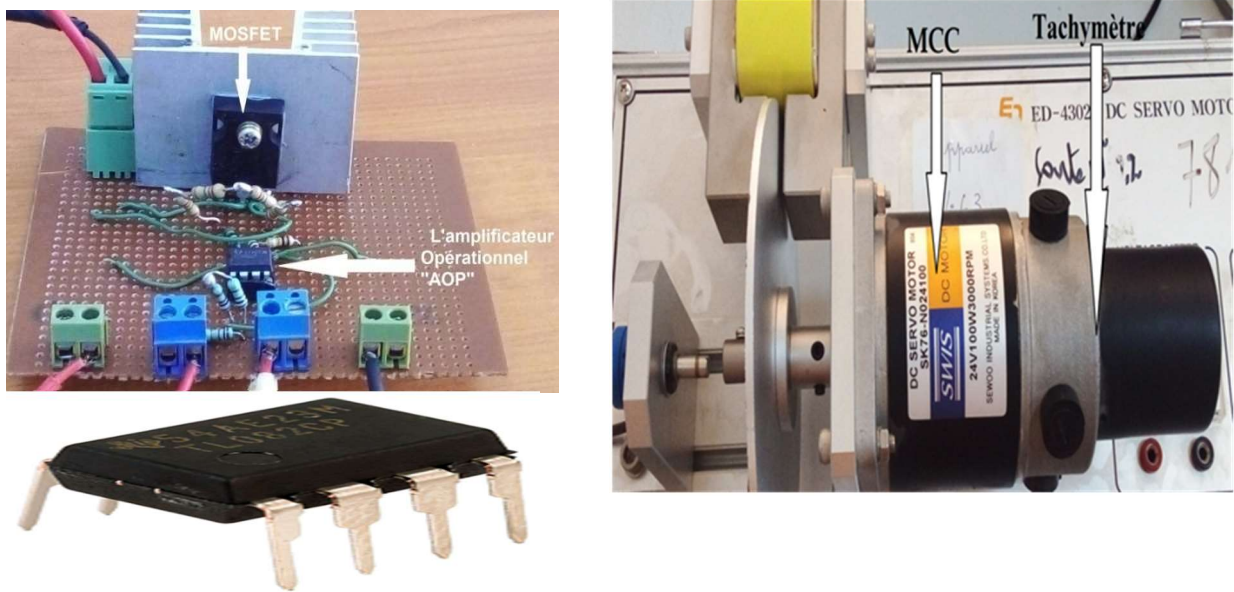


Figure IV.2 : Le moteur et le circuit d'amplification

### IV.3 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 réalisée de diverses manières : séparée, shunt, série, compound et par aimants permanents. Ce dernier est le plus utilisé pour les petits moteurs

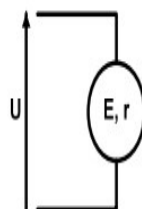


Figure IV.3: Moteur CC à aimant permanents

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

$U$  est la tension d'excitation.

$$U = r \cdot I + E$$

Le flux étant constant, on peut écrire que  $E = K \cdot W$

$\Omega$  : étant la vitesse de rotation.

### Tachymétrie

La tachymétrie ou la génératrice bathymétrique délivre une tension proportionnelle à sa vitesse de rotation. Pour la génératrice à courant continu l'excitation est assurée par des aimants permanents.

Le schéma de principe est le suivant :

$R_c$  est la résistance de la charge.

$$E = r \cdot I + U \text{ avec } E = K \cdot W \text{ et } U = R_c \cdot I$$

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

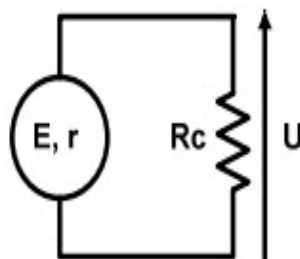


Figure IV.4 : Schéma de fonction d'une génératrice bathymétrique

$K_e$  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.4 Description du matériel

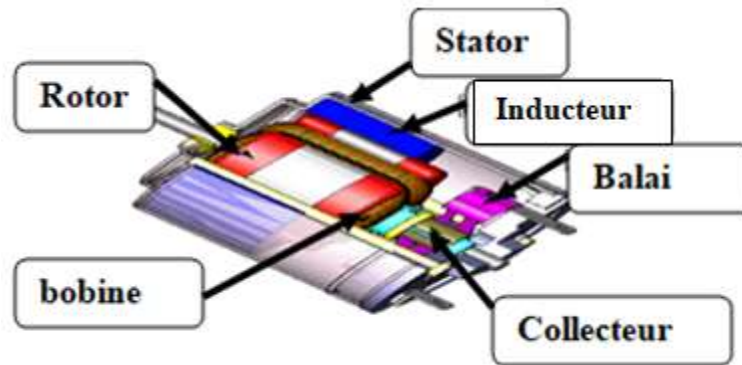


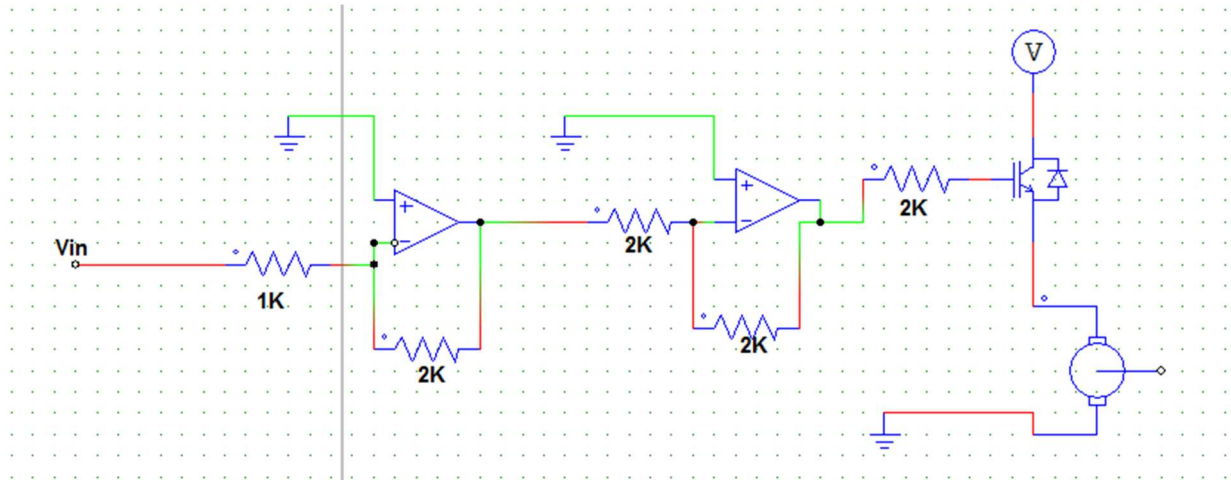
Figure IV.5 : Moteur à courant continu

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 ( $K_e = 1.8 \cdot 10^{-3}$ ).

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.

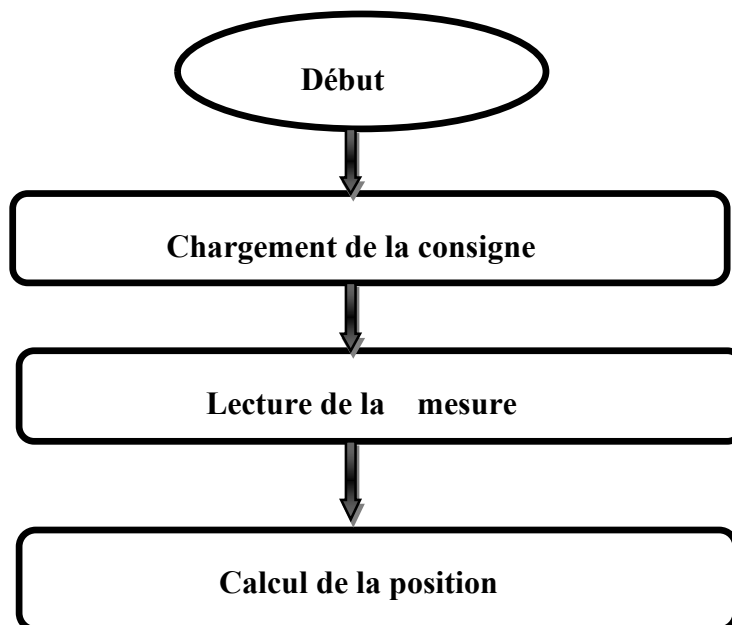
Le schéma électrique du circuit d'amplification est présenté dans la figure ci dessous:



**Figure IV.6 :** Schéma électrique du circuit d'amplification

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.5 Description du programme :



**Figure IV.7 :** Organigramme de la commande en boucle ouverte

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 Chargement de la consigne

La consigne peut être chargée de deux façons :

1. Manuel : la consigne est introduite dans le programme (mode hors-ligne)
2. 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 :

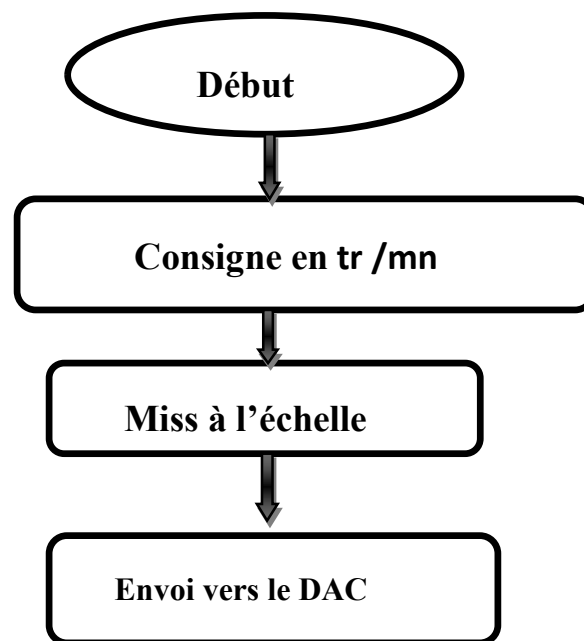


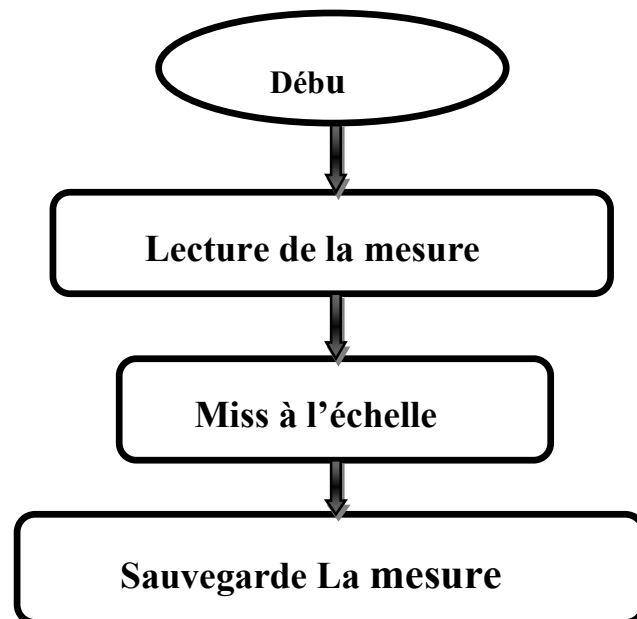
Figure IV.8 : Chargement de la consigne

#### IV.7 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 bathymétrique délivre une tension entre 0 et 10V pour une vitesse angulaire

entre 0 et 1400Tr/mn. Cette tension sera connecté à l'entrée analogique l'automate.  
Comme pour le chargement de la consigne une mise à l'échelle doit être mise en place.



**Figure IV.9:** 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

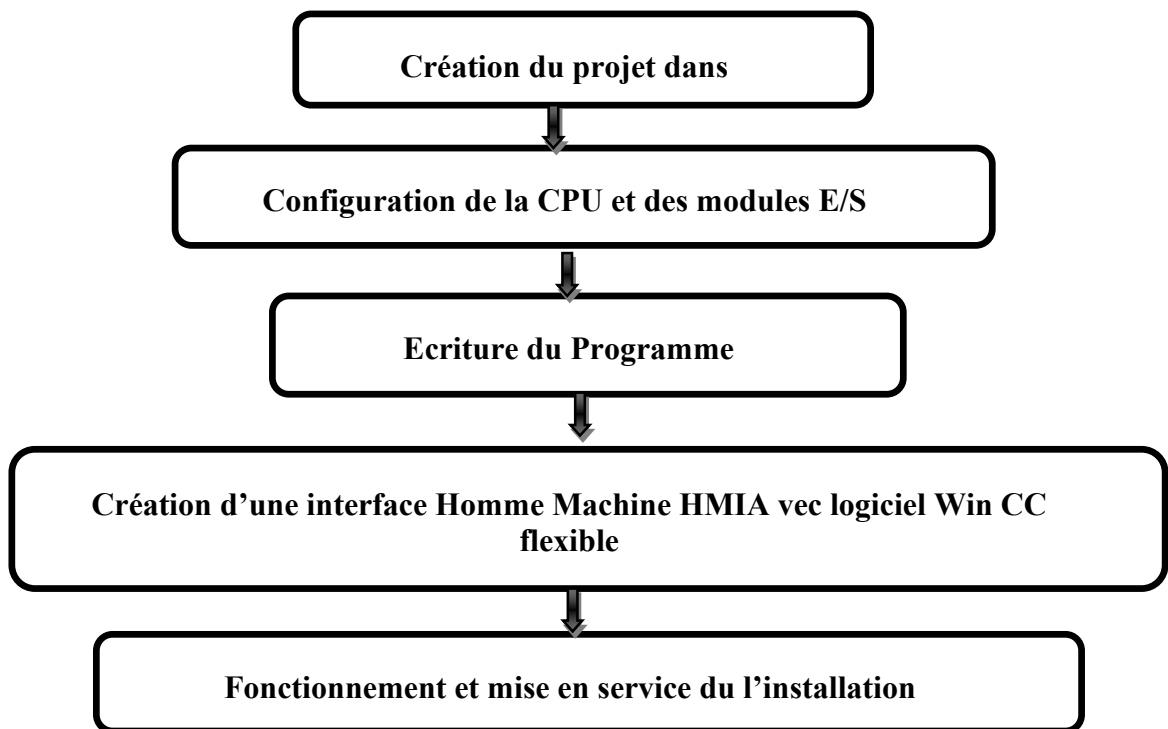


Figure IV.11 : Organigramme de l'application finale

## IV.9Création du projet S7 pour la commande d'un MCC

### IV.9.1 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 :

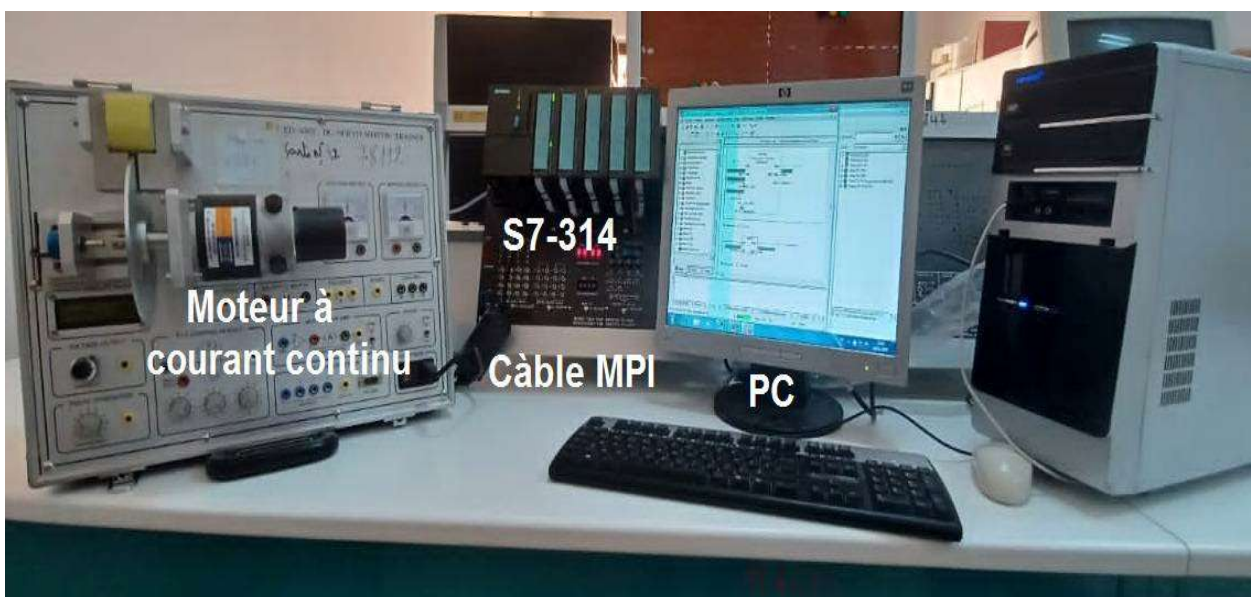


Figure IV.12 : Synoptique de la commande par API

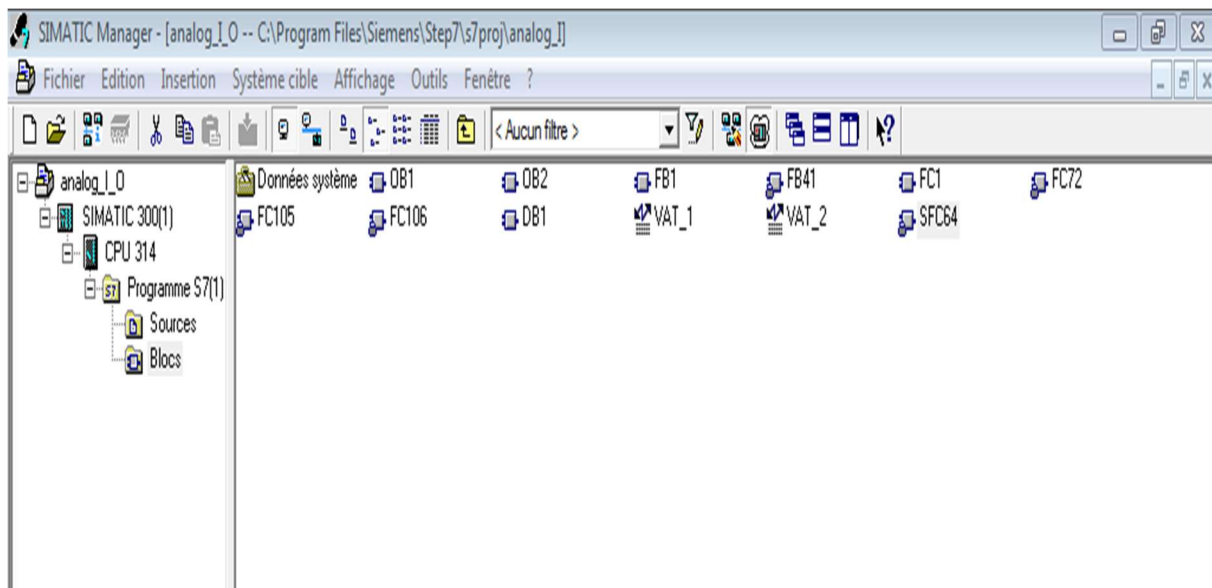


Figure IV.13 : Création du projet

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

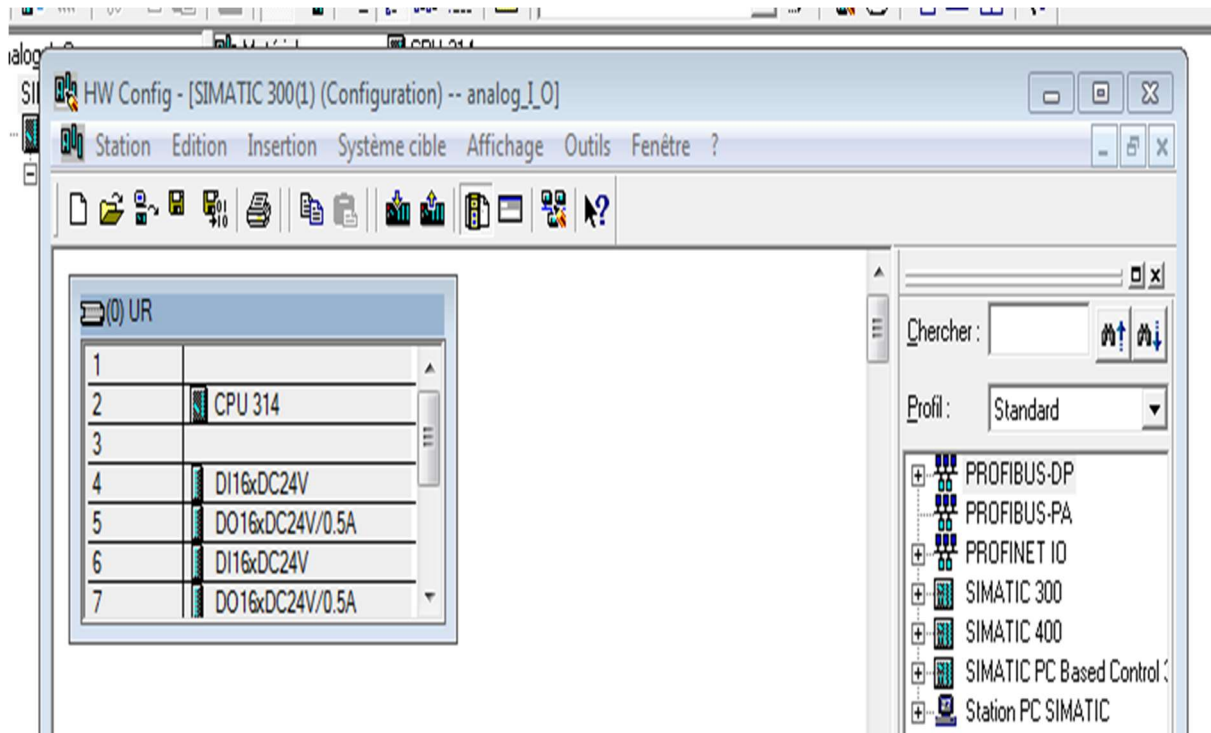


Figure IV.14 : Configuration matériel

**IV.10 Application et visualisation de la commande**

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

Le signal de commande est représenté dans la figure suivant :



**Figure IV.15 :** Vue générale crée pour la HMI



**Figure IV.16 :** La tension et la vitesse du système( Entrée/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.17 :**La réponse de la vitesse et tension correspondante

On voit bien que le signal se stabilise autour de la tension 2.73 V, 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.11 Programme de la commande du moteur :

- Programme dans FC2 Lecture de la mesure :

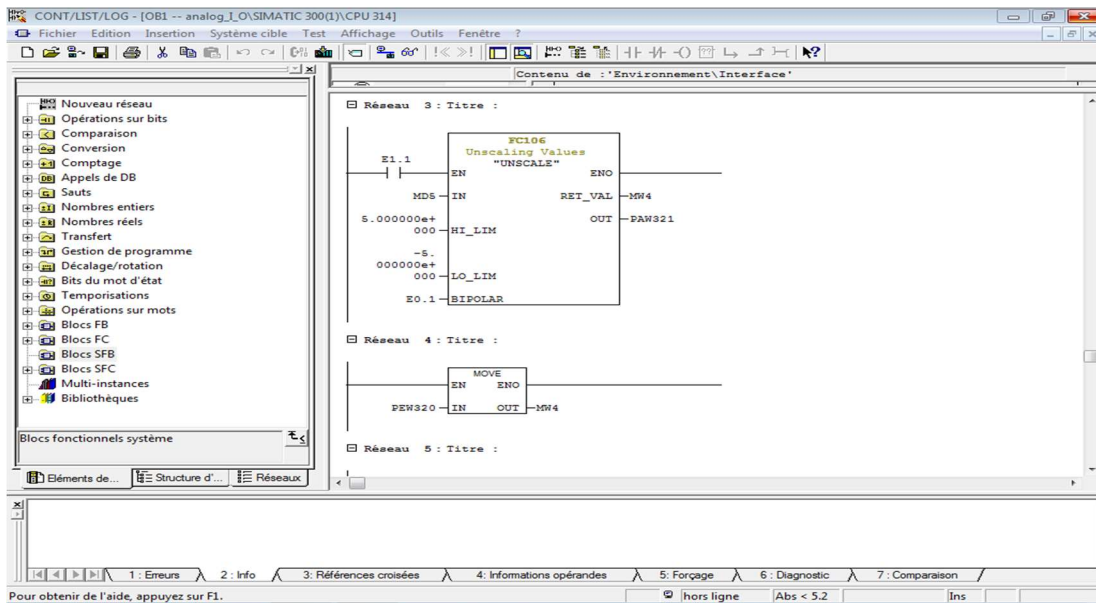


Figure IV.18 : bloc UNSCALE

- Programme dans FC3 Chargement consigne :

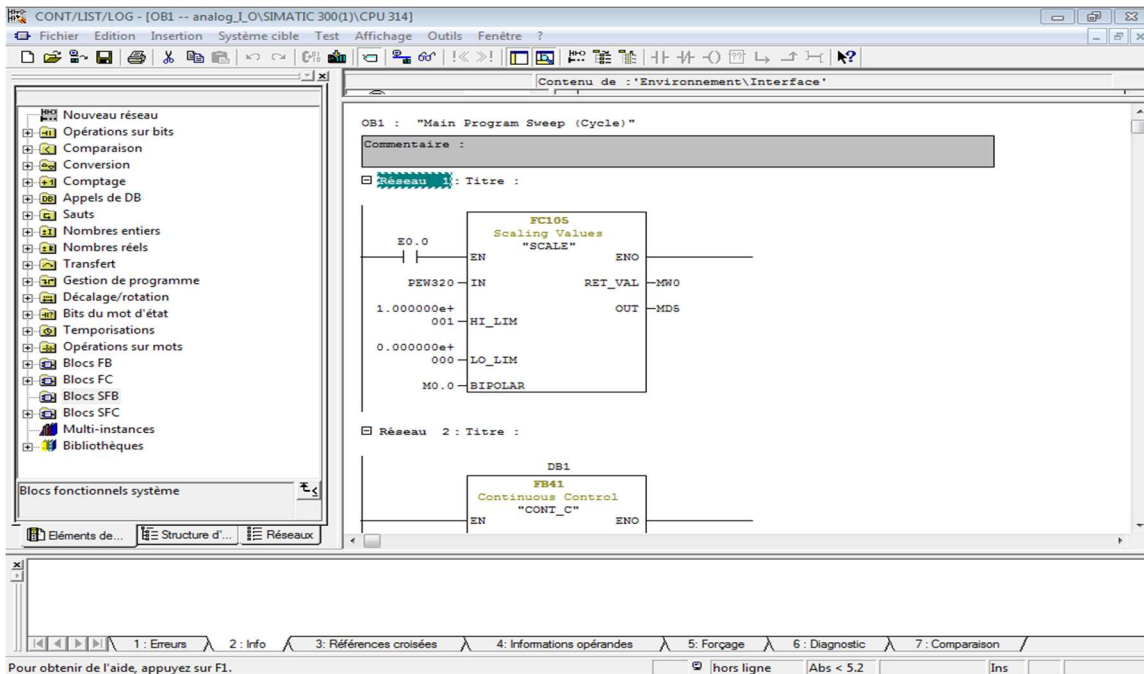


Figure IV.19 : Bloc SCALE

**IV.12 Conclusion**

Dans ce chapitre nous avons étudié la commande en boucle ouverte de moteur à courant continu. Dans le logiciel STEP7 la première étape est le chargement de la consigne, la lecture de la vitesse angulaire moyenne et le calcul de la position angulaire finale et du nombre de tours effectués. Le programme est créé dans SIMATIC MANAGER. La création d'un HMI avec le logiciel WinCC flexible pour la visualisation de la commande.

# ***CONCLUSION GENERALE***

### Conclusion générale

Ce mémoire étudie les différentes étapes de l'élaboration d'un projet software et l'établissement d'une commande à base d'automate programmable SIEMENS grâce au logiciel de programmation STEP7 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.

## الملخص :

يهدف هذا العمل إلى إظهار مختلف الخيارات المتاحة في الستاب 7 للتحكم التسلسلي والتحكم المستمر في الوقت الحقيقي، وذلك لتنفيذ حلول ومهام الآلية والتحكم على المبرمج الآلي من السلسلة 7 التابعة لشركة سيمنس الألماني S7-314.

يتميز الستاب 7 بواجهة رسمية سهلة الاستخدام، حيث البرامج مهيكلة في وحدات مما يسهل تحديد المشكلات وتحديثها.

تم أنحاز التطبيقات المختلفة في المخبر بقسم الآلية في جامعة المسيلة، تبين أن المبرمج الآلي S7-314 يوفر إمكانية واسعة في التحكم و الآلية.

## الكلمات المفتاحية :

المبرمج WinCC، المبرمج الآلي S7-314، ستاب سات، التحكم في المحرك ذو التيار المستمر،

## Résumé :

Ce travail permet de montrer les différentes possibilités qu'offre le STEP7, pour la commande séquentielle et la commande continue 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.

## *Références bibliographiques*

- [1] A. Gharbi, « Etude De Synthèse Sur Un Système De Production Automatisé De Tindal / Complexe Algérienne Des Textiles Industrielles / Unité M'sila », Université De M'sila ,2022
- [2] S. Naguez, « Association Variateur De Vitesse Et Moteur Asynchrone Et Pc Avec Communication Profinet Par Automate Programmable Siemens S7\_1200 », Université De M'sila ,2022
- [3] O. Alioui, « Mise En Œuvre D'une Application A Base Du Profinet Avec Automate Programmable Industriel Siemens Et Variateur De Vitesse », Université De M'sila ,2022
- [4] L. Bourgumoux, « Les Automates Programmables Industrielles », Polytech Marviel, 2005.
- [5] [www.Wikipedia](http://www.Wikipedia).
- [6] [www.Nicolasgled.Com](http://www.Nicolasgled.Com).
- [7] G. Michel, « Les A.P.I Architecture Et Application Des Automates Programmables Industriels », Edition Dunod, 1987.
- [8] Siemens, « Appareils De Terrain Pour L'automatisation Des Processus », Sitrans, 2005.
- [9] [www.Siemens.Com](http://www.Siemens.Com)
- [10] Siemens, « Step 7, Régulation Pid », Simatic, 2000.
- [11] Siemens, « Programmation Avec Step7 », Simatic, 2000.
- [12] D. Nanping, « Application Of S7-400 And Wincc In Dense Phase System », Université Beijing, China / China, Troisième Symposium International Sur Les Ateliers D'application Des Technologies De L'information Intelligentes.
- [13] Manuels Siemens, « Wincc Flexible », Simatic Hmi, 2008.