



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et Informatique Industriel

Réf. : Entrez la référence du document

Présenté et soutenu par :
Benhamza ilyes

Le : lundi 25 juin 2018

**L'automatisation d'un procédé 'pince dépileur ' dans l'usine de
briqueterie babahoum par l'automate s7-300 simens**

Jury :

Mr	ABDOU LATIFA	MCA	UNIV BISKRA	PRESIDENT
M	TOUBA MOSTEFA	MAA	UNIV BISKRA	ENCADREUR
Mr	TURKI NAJIBA	MCB	UNIV BISKRA	EXAMENATEUR

Année universitaire : 2017/2018

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie

Département de Génie Electrique

Filière : Automatique

Option : Génie du système industriel

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme:

MASTER

Thème

**L'automatisation d'un procédé 'pince dépileur ' dans l'usine de
briqueterie babahoum par l'automate s7-300 simens**

Présenté par :

Benhamaza Ilyes

Avis favorable de l'encadreur :

Touba Mohamed Mostefa

Avis favorable du Président du Jury

Abdou Latifa

Signature

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Génie des systèmes industriel

Thème :

L'automatisation d'un procédé 'pince dépilleur ' dans l'usine de briqueterie babahoum par l'automate s7-300 simens

Proposé par : Benhamza Ilyes

Dirigé par : Toubia Mohamed Mostefa

RESUMES (Français et Arabe)

Ce travail entre dans le cadre de la technologie de automatisation et étudier la façon de rendre le processus fonction d'une manière automatique pour que nous avons réalisé notre travail dans l'entreprise Babahoum pour produire la brique rouge et plus particulièrement dans la zone dépilleur que vider le brique pour l'envoyer au magasin pour le vendre, et ceci est fait par la une ensemble des machines que construire une système automatisé réduit dans un appareil s'appelle pince dépilleur que où nous avons utilisé dans ce processus l'automate SIMENS s7 300 de la production de la société allemande Siemens dans le but de programmer ce dernier pour contrôler le travail du pince fin d'accélérer son travail et son coût et gagner du temps et réduire l'effort des moteurs pour éviter les défaillances fréquentes

Nous sommes partis de la soumission et la présentation de l'entreprise, puis expliquer les caractéristiques d'automate S7-300 et puis nous avons fini d'expliquer la structure du pince et son fonctionnement et nous l'avons programé l'automate par logiciel STEP7 et nous avons testé nos simulations en utilisant le programme de simulation WIN-CC FLEXIBLE

ملخص

هذا العمل يندرج في اطار تكنولوجيا التحكم الالي ودراسة كيفية جعل العملية تسير بطريقة الية بحيث قمنا بعملنا في مؤسسة باباهم لانتاج الاجر الاحمر و تحديدا في الجزء المتعلق بافراغ الاجر لارساله الى المخزن لبيعه, وذلك يتم بواسطة مجموعة من المحركات تشكل نظام الي يتمثل في كمامشة للافراغ حيث استعملنا في هذه العملية المتحكم الالي SIMENS s7-300 من انتاج شركة سيمنس الالمانية لهدف برمجة هذا الاخير ليتحكم في عمل الكمامشة لاجل تسريع عمله ومردوديته وتوفير الوقت و التقليل من جهد المحركات لتفادي كثرة الاعطال .
انطلقنا من تقديم و عرض للمؤسسة ثم شرح لخصائص المتحكم الالي S7-300 وانتهينا بشرح لبنية الكمامشة وتركيبته وكيفية عمله وقمنا ببرمجة المتحكم الالي بواسطة STEP7 و اختبرنا عملنا باستعمال برنامج المحاكاة WIN-CC FLEXIBLE .

Dédicace

Ma mère la source et l'espoir !

Mon père le repère et l'exemplaire !

*A vous je dédis ce modeste mémoire pour
avoir m'ont encouragé et poussé à atteindre l'idéal,
m'ont soutenu*

tout au long de mes études.

Je dédie ce modeste travail à :

Mon frère daoude

Mes soeurs shahra, shaima.

Mon cousin ghazali.

Tous mes fidèles amis :

Brahim ,youcef.

*Tout les enseignants et étudiants de la brache
automatisme.*

*Je dédie enfin ce mémoire à toute personne ayant
contribué de près ou de loin à sa concrétisation.*

Remerciement

Louange à DIEU le très grand et miséricordieux, le seul et unique qui nous a donné la force et le courage pour terminer nos études et élaborer ce travail.

Avant de commencer la présentation de ce travail, Nous profitons de l'occasion pour remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études.

Nous tenons à exprimer nos vifs remerciements pour mon grand et respectueux, **Moustafa toub**, d'avoir accepté de nous encadrer pour mon projet de fin d'études, ainsi que pour ses précieux conseils et surtout pour nous avoir laissé une grande liberté dans la conception et la rédaction de ce travail, ses remarques pertinentes et son encouragement.

Nous tenons à exprimer nos profondes reconnaissance et toutes nos pensées de gratitude à **brahim zergeni**, qui nous accompagne de près durant tout ce travail, pour sa disponibilité, pour la confiance qu'il a su nous accorder et les conseils précieux qu'il nous prodigués tout au long de la réalisation de ce projet.

Nos remerciements vont aussi à tous professeurs, enseignants et toutes les personnes qui nous soutenus jusqu'au bout, et qui ne cessent de nous donner des conseils très importants en signe de reconnaissance.

Sommaire

Introduction général	1
-----------------------------------	---

Chapitre 01 : representation de l'entreprise

.....

Introduction	3
---------------------------	---

1-présentation de la briqueterie babahoum	3
--	---

1.1-Historique.....	3
---------------------	---

1.2-Atelier de préparation.....	3
---------------------------------	---

1.3-L'atelier de fabrication.....	4
-----------------------------------	---

2-Les différents types de brique que l'entreprise produire	4
---	---

3-principe de fabrication des briques	5
--	---

3.1-Extraction.....	6
---------------------	---

3.2-préparation d'argile.....	7
-------------------------------	---

3.3-séchage.....	10
------------------	----

3.4-cuisson.....	11
------------------	----

3.5-Dépilage.....	12
-------------------	----

4-Opérations de maintenance préventive	13
---	----

5-conclusion	14
---------------------------	----

Chapitre 02 :l'automate programmable industriel

.....

Introduction	15
---------------------------	----

1-l'automate programmable industriel	15
---	----

1.1-définition.....	15
---------------------	----

1.2-les marques d'automates.....	15
----------------------------------	----

2-Automate siemens	15
2.1- Contrôleurs modulaires SIMATIC (la gamme).....	15
2.2- Les logiciels utilisés pour programmer automate siemens.....	15
2.3- SIMATIC S7_300.....	16
3-SIMATIC S7-300 CPU315-2DP	17
3.1- Interfaces.....	18
3.2-Configuration et programmation.....	20
3.3 Eléments de commande et de signalisation.....	20
3.4- Vue d'ensemble.....	23
3.4.1-Constituants d'un S7-300.....	23
3.4.2-Configuration.....	24
3.4.3- Sous-réseaux.....	24
3.5-Adressage.....	25
3.6- Mise en service.....	26
4 .Communication avec le S7-300	26
4.1- Interfaces.....	27
4.1.2- Interface MPI/DP (X1).....	27
4.1.3- Interface PROFIBUS DP(DRIVE) (X3.....	28
5. Concept de mémoire	29
5.1- Zones de mémoire et rémanence.....	29
5.2- Plages d'opérandes de la mémoire système.....	29
5.3- Fonctions de mémoire.....	31
5.3.1- Charger le programme utilisateur.....	31
6-Critères de choix	31
7- Utilisation de Step 7	32

7.1-Créer son projet.....	32
7.2-Le matériel.....	33
7.3-Le programme.....	35
7.4-Le langage CONT.....	36
7.5-Le langage LOG.....	37
7.6-Le langage LIST.....	38
8-conclusion.....	41

Chapitre 03: le pince dépileur

.....

Introduction.....	43
1 .description de la pince.....	43
2. structure général de la pince DP.....	45
3- Cahier des charges.....	50
4-grafcet.....	52
4- exemple de programme de quelque parts de cycle du pince.....	54
4-1. sélection de mode auto ou mode manuel.....	54
4-2. Avance pince en mode auto.....	55
4-3. recule pince en mode auto.....	55
4-4.ouverture de pince en mode auto.....	56
4-5 monte pince	57
4-6 rotation tetes manuel 90 degré.....	58
4-7 rotation tetes manuel 0 degré.....	61
4-8 rotation tetes auto 90 degré.....	60
5-conclusion.....	60
Conclusion générale	61

Liste de figures

Figure 1.1 : Chaîne de préparation.....	03
Figure 1.2 : Chaîne de fabrication.....	04
Figure 1.3 : Brique ordinaire perforées de 12 trous.	05
Figure 1.4 : Brique ordinaire perforées de 8 trous.	05
Figure 1.5 : Brique ordinaire perforées de 16 trous.	05
Figure 1.5 : Schéma de la fabrication des briques.	05
Figure 1.6 : Tapis transporteur argile et sable (mélange).	06
Figure 1.7 : convoyeurs entre les distributeurs et desagregateur et broyeur pré finisseur.....	07
Figure 1.8 : Mouilleur de la matière première.	07
Figure 1.9 : Broyeur de la matière première.....	08
Figure 1.10 : BSF (broyeur super finisseur).	08
Figure 1.11 : Malaxeur.	09
Figure 1.12 : la mouleuse.	09
Figure 1.13 : Séchoir.	10
Figure 1.14 : empileur.	10
Figure 1.15 : l'entrée de briques préparées au four.....	11
Figure 1.16 : sorti de four.	11
Figure 1.17 :La zone dépileur.....	12
Figure 1.18 :La zone dépileur.....	12
Figure 2.1 : SIMATIC S7_300.	16
Figure 2.2 : tout l'élément général d'automate avec ordre.	16
Figure 2.3 :CPU315-2DP.	18

Liste de figures

Figure 2.4: Configuration typique avec CPU technologique.....	19
Figure 2.5 : Eléments de commande et d'affichage de la CPU technologique.....	20
Figure 2.6 : Entrées et sorties intégrées de la CPU technologique, visibles après ouverture de la porte en façade.	21
Figure 2-7 Constituants d'un S7-300.	22
Figure 2-8 Emplacements du S7-300 avec CPU technologique et adresses initiales des modules respectifs.....	26
Figure 3-9 : Interfaces de la CPU technologique.....	27
Figure 3-10 Zones de mémoire de la CPU technologique.	29
Figure 3-11 Phases de traitement au cours d'un cycle.	30
Figure 3-12 Mémoire de chargement et mémoire de travail de la CPU technologique.....	43
Figure 3.1 : la pince dp.	44
Figure 3.2 : fin de course de wagon.	44
Figure 3.3 : wagon pleine de brique.	44

Liste de figures

Figure 3.4 : cellule que détecte la présence de brique.	45
Figure 3.5 : moteur triphasé au dessous de la tête 2.	45
Figure 3.6 : deux vérins en parallèle entre les doigts.	45
Figure3. 7 : Moteur levage.	46
Figure3. 8 : Détecteur fin de course sécurité chaine.	46
Figure3.9 : détecteur 24v pour rotation de la tête.	47
Figure3. 10 : détecteur 24V pour l'ouverture.....	47
Figure3.10 :l'axe général de translation.	48
Figure 3. 11 :moteur triphase pour translation la pince.	48
Figure 3.12 : deux capteurs de sécurité pour limite de translation.....	49
Figure 3.13 : les variateurs de la pince.	49
Figure3.14 :l'automate de la pince dp (siemens S7 300 15-2dp).	50

Liste des tableaux

Liste des tableaux

Tableaux 1.1 les opérations de maintenance.....	14
Tableau 2-1 Eléments de commande et d'affichage de la CPU technol.....	21
Tableau 2-2 : Entrées et sorties intégrées de la CPU technologique.....	22
Tableau 2-3 : Positions du commutateur de mode de fonctionnement.....	22
Tableau 2-4 Affichages d'état et d'erreur de la CPU.....	23
Tableau2-5 Constituants d'un S7-300	24
Tableau 2-6 Partenaires sur le sous-réseau.....	25
Tableau 3-7 Modes de fonctionnements possibles des interfaces avec la CPU technologique.....	25
Tableau 3-8 Plages d'opérandes de la mémoire système.....	51

Introduction général

Introduction général

Les automates programmables sont apparus aux USA vers les années 1969, à l'époque ils étaient destinés essentiellement à automatiser les chaînes de montages automobiles. Ils sont apparus en France en 1971, ils sont de plus en plus employés dans l'industrie.

On sait maintenant que la deuxième partie du XXe siècle a passée à l'histoire comme étant l'ère de l'automatique.

Dans le sillon de l'automatique apparurent plusieurs autres « tiques » et entre autres, l'informatique et la robotique. Et c'est à travers l'automatique, d'abord en 1968-69 aux Etats Unis, que les premiers automates industriels ou « contrôleurs programmables » firent leur apparition.

Leurs premières applications furent d'abord le remplacement des horloges de contrôle du temps des employés ; par la suite, leurs multiples utilisations industrielles, en particulier sur les lignes de production des usines, deviennent indispensables non seulement au point de vue contrôle, mais aussi du côté économique pour l'espace et l'entretien. C'est alors que de nombreux systèmes à relais durent céder leur place.

Les premiers automates programmables n'effectuaient que la commutation ON/OFF (et vice-versa) avec la possibilité de temporisation, comme les relais. Leurs applications étaient limitées seulement aux procédés répétitifs ainsi qu'à certaines machines. Par contre, leurs avantages consistaient dans une installation plus facile, la visualisation des étapes; ils possédaient des indicateurs diagnostiques permettant la localisation des pannes. C'était déjà mieux que les relais, en plus de pouvoir être reprogrammé advenant un changement de fonction ou de procédé.

De 1970 à 1974, la technologie des microprocesseurs (du moins les premiers) ajoutèrent une plus grande flexibilité et une « intelligence » à l'automate programmable. Les capacités d'interface avec l'utilisateur s'améliorent. L'automate programmable peut maintenant exécuter les opérations arithmétiques en plus des opérations logiques; il manipule les données et les adresses ; il effectue la communication avec d'autres automates ou ordinateurs, donnant ainsi une nouvelle dimension aux applications de l'automate programmable.

La console de programmation s'allie avec un moniteur permettant la programmation avec des symboles familiers de relais ce qui facilite beaucoup la compréhension et le dépannage car la logique peut être vue dans la même forme que les dessins à relais.

Les automates programmables utilisent une mémoire non-volatile (RAM+Pile, EEPROM ou EAPROM par exemple) pour emmagasiner les instructions. Ces derniers accompliront des fonctions logiques, arithmétiques, de temporisation, de comptage et de manipulation des données. En plus, les fonctions de contrôle PID et d'autres fonctions complexes comme le contrôle numérique de processus sont présentes. Puisque les automates programmables ont été conçus pour accomplir des opérations semblables à celles des relais, la programmation est basée généralement sur la nomenclature des diagrammes en échelle (ou schéma à relais). Des langages de haut niveau ont été aussi implantés sur certains automates afin de produire une plus grande flexibilité de programmation.

Introduction général

Dans ce contexte, nous avons fait une étude sur le processus de fonctionnement d'un système automatisé et c'est au niveau de l'usine briqueterie babahoum dans le wilaya de Biskra, nous avons programmé un ensemble des machines que construire un système automatisé réduit dans un appareil que s'appelle pince dépileur et ce dernier est un appareil que vider les wagon de brique, et cette machine fonctionne en parallèle par automate programmable industriel siemens S7-300 (315-2DP).

Le présent mémoire est organisé comme suite :

-Le premier chapitre présente les différents ateliers existants à l'intérieur de l'usine de fabrication de brique d'une part et d'autre part de détailler chaque phase de l'atelier à sa propre tâche d'exécution.

-Le deuxième chapitre expose la structure générale de l'API et ensuite détaille l'API S7-300 que nous allons l'utiliser et comment la programmer, et puis on donne un aperçu sur le logiciel step 7.

-Le troisième chapitre se compose de deux parties, la première partie représente en détail la pince et la deuxième partie représente le programme de l'automate pour commander dans le processus de fonctionnement de la pince.

Chapitre 01 : représentation de l'entreprise

Introduction

L'objectif de ce chapitre est de présenter un bref sur les étapes de la fabrication des briques dans l'usine Babahoum, et ces premières sont des produits céramiques dont les matières premières sont des argiles, avec additifs, la forme des briques est généralement rectangulaire. Elles sont utilisées dans la construction des bâtiments et des travaux publics.

On présentera la briqueterie Babahoum et on détaillera le principe de fabrication de brique et mentionnera les machines utilisées pour ce processus et on terminera par le processus de dépilage et les opérations de maintenance préventive.

1- présentation de la briqueterie babahoum

1.2- Historique

A titre d'exemple l'unité de Briqueterie Babahoum Branis-dar Aarous a une capacité de production de 124416 briques par jour, a été mise en service depuis mars 2001.

1.3- Atelier de préparation

L'atelier de préparation des argiles se constitue des équipements suivants :

3 distributeurs : 1 pour l'argile
1 pour sable
1 pour réserve

1 Désagregateur

1 Broyeur dégrossisse

1 Mouilleur

1 Broyeur pré-finisieur

1 Tapis navette

1 Convoyeurs (tapis)

1 Stock

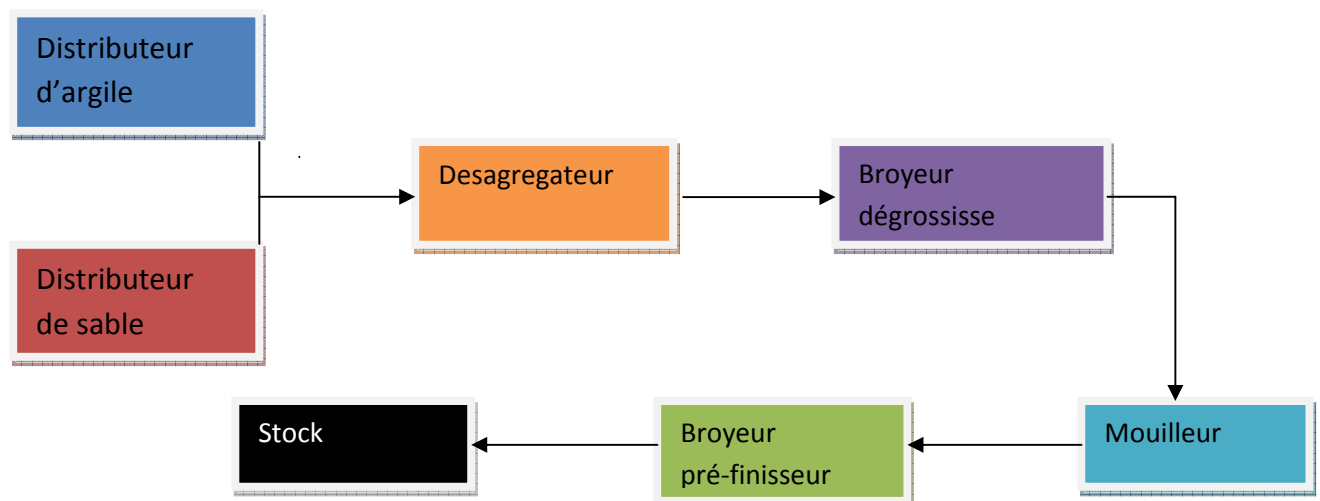


Figure 1.1 : Chaîne de préparation.

1.4- Atelier de fabrication

L'atelier de fabrication est constitué des équipements suivants :

- (01) BSF
- (01) Malaxeur
- (01) Mouleuse
- (01) Pré coupeur
- (01) Coupeur multi fils
- (01) Séchoir
- (01) empileur
- (02) Four (temps de séchage 4 heures)
- Un dépileur automatique

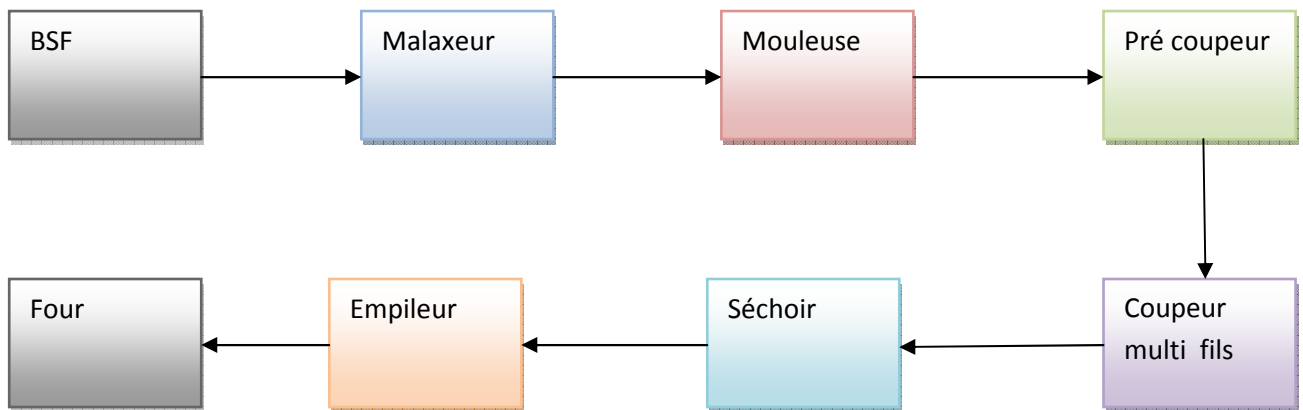


Figure 1.2 : Chaîne de fabrication.

2- Les différents types brique que l'entreprise produit

La brique est un matériau de construction qui est fabriqué en portant une petite quantité d'argile, préalablement mise en forme, à une température appropriée la température de frittage. Les particules d'argile commencent alors à fondre et s'agglomèrent pour former une masse à caractère pierreux. Après la cuisson, la brique conserve une certaine porosité, qui lui confère d'ailleurs des propriétés spécifiques et la distingue des autres matériaux de construction.

Les briques se divisent en 3 groupes principaux sont : brique ordinaire, brique poreuse, brique d'argiles creuse à perforation.

L'entreprise BABAHOUM produit seulement du brique ordinaire et précisément les modèles 8 trous (10 cm) et 12 trous (15cm) et 16 trous (20cm).



Figure 1.3 : Brique ordinaire perforées de 12 trous.



Figure 1.4 : Brique ordinaire perforées de 8 trous.



Figure 1.5 : Brique ordinaire perforées de 16 trous.

3- Principe de fabrication des briques

Les briques sont une argile qu'obtient leur état final par être cuises d'environ 1000 C°. Par des combinaisons chimiques à partir de 700 C°, l'argile en perdant son eau, se transforme en terre cuite. L'oxyde de fer des argiles donne la coloration aux briques.

En général la fabrication des briques se compose des cinq opérations principales comme indiquées sur le schéma suivant :

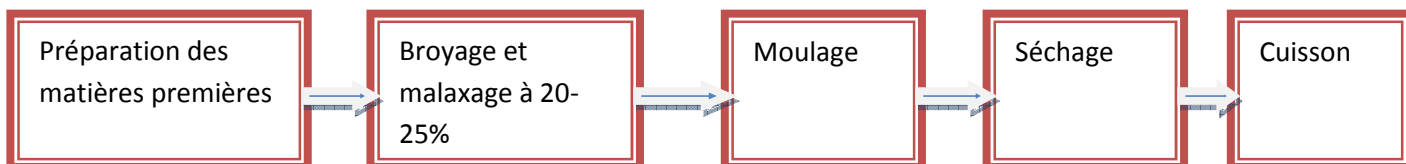


Figure 1.5 : Schéma de la fabrication des briques.

3.1- Extraction

La matière de la brique de terre cuite est l'argile. il faut entendre 'argile' au sens large et considérer également les terres limoneuse et schisteuses, matières premières de même composition minéralogique que l'argile.

L'argile est extraite dans des argilières situées en zone d'extraction. L'argile est omniprésente dans le sous-sol et ses propriétés différentes en fonction de l'origine géologique [1].

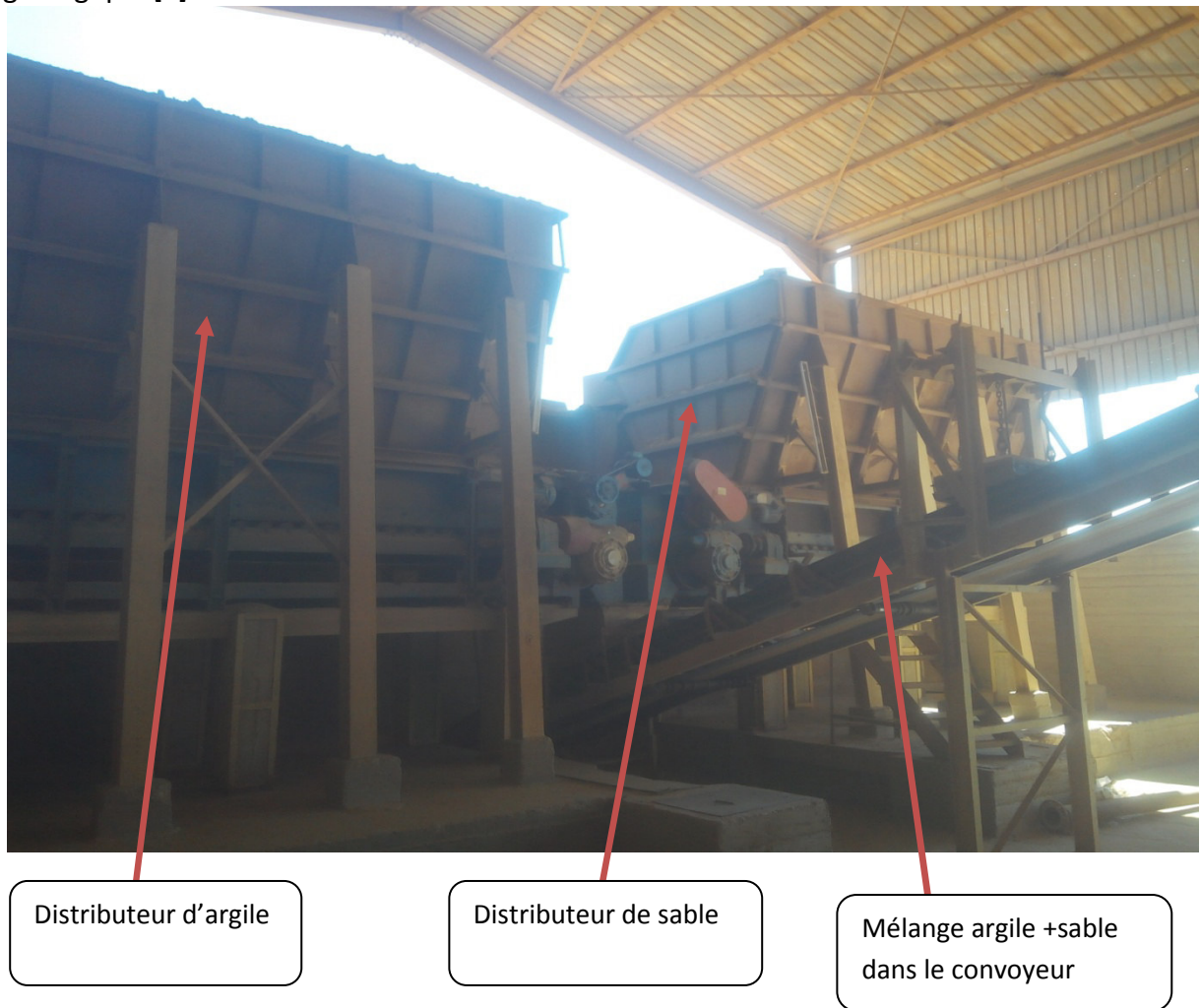


Figure 1.6 : Tapis transporteur argile et sable (mélange).

3.2- Préparation d'argile

Il ya deux opérations principales pour la préparation sont le broyage et le malaxage premièrement et le dosage et le mélange des matières premières et deuxièmes.

Le mélange (argile +sable) passe par le désagrégateur et puis par broyeur dégrossisse pour être défiguré et passe par le mouilleur et finalement par broyeur pré finisse et avec ce dernière étant placé dans le stock afin qu'il soit prêt pour la phase de fabrication.



Figure 1.7 : convoyeurs entre les distributeurs et desagregateur et broyeur pré finisseur.

Remarque : Broyeur dégrossisse a été ajouté dans l'usine BABAHOUM pour améliorer la qualité du produit et fait des briques de BABAHOUM des meilleures briques dans l'Algérie.



Mouilleur

Figure 1.8 : Mouilleur de la matière première.

Le but est d'obtenir une masse argileuse bien homogène qui sera facilement transformée en produit fini.



Figure 1.9 : Broyeur de la matière première.

Après le broyage l'argile est mise à stock prêt pour la phase de fabrication, alors le distributeur commencer par envoyer la matière première vers BSF (broyeur super finisseur).



Figure 1.10 : BSF (broyeur super finisseur).

Le BSF broie la matière première pour la dernière fois et la convoyer au malaxeur et puis la matière sortira par la mouleuse en forme de briques directement vers le séchoir.



Figure 11 : Malaxeur.



Figure 1.12 : la mouleuse.

3.3- Séchage

Avant d'être cuites, les briques crues doivent encore perdre une grande partie de leur teneur en eau – du moins en est-il ainsi pour la plupart des argiles. Le séchage se poursuit jusqu'à ce que les briques ne contiennent plus qu'environ 2% d'eau. Le risque serait en effet de les voir se fendre ou éclater sous la dilatation de la vapeur dans la masse. D'autre part, la stabilité dimensionnelle du produit n'est obtenue qu'au terme du retrait consécutif à la dessiccation.

Le séchage s'opère dans des chambres ou des tunnels où il se poursuit de manière régulière et rapide (généralement de 2 à 4 jours). On utilise l'air chaud de la zone de refroidissement du four pour le séchage des briques. La température et le taux d'humidité sont contrôlés tout au long du processus de séchage, au moyen d'un système informatique réglé de façon très précise. [1]



Séchoir

Figure 1.13 : Séchoir.

Les briques entrent et sortent du séchoir au niveau d'empileur.



Sorti de séchoir

Figure 1.14 : empileur.

3.4- Cuisson

C'est la dernière étape que doit subir la brique d'argile façonnée et séchée, avant de pouvoir devenir une brique de terre cuite à proprement parler. C'est là une phase d'une grande importance qui doit se dérouler très progressivement. On augmente graduellement la température jusqu'à l'obtention de la température de cuisson (comprise entre 850 et 1200°C, en fonction du type d'argile) ; on diminue ensuite progressivement la température jusqu'au refroidissement complet. Chaque mélange d'argile se caractérise par sa propre «courbe de cuisson». [1]



Figure 1.15 : l'entrée de briques préparées au four.

3.5- Dépilage

Après la cuisson, les briques sont prêtes à être transportées et livrées sur chantier. Pour des raisons de facilité et de sécurité, elles sont préalablement empilées sur des palettes et emballées de façon à minimiser la quantité d'emballage utilisé.

Le wagon sort de four et sois transporté par transbordeur vers la zone dépileur.



Figure 1.16 : sorti de four.

La zone dépileur est démontré sur Figure 17 et 18



Figure 1.17



Figure 1.18

4 -Operations de maintenance préventive

L'entretien des différents équipements de la réalisation de l'unité de la briqueterie s'effectué selon le programme cité ci-dessous (tableau 1) :

Désignation	Entretien	Fréquence
Motoréducteur	Vérification niveau d'huile vidange	1 fois par semestre
électricité	-Vérification d'ompirage -Vérification des fins de course -Serrage borniers -Dépoussiérage des cellules photo électrique	-1 par jour -1 fois par mos -1 par jour
Rouleaux	-Nettoyage et graissage -Vérifier l'état des rouleaux	1 fois par mois
Palier	Graissage	1 fois par mois
Circuit pneumatique	-Vérification des flexibles -Nettoyage des distributeurs -Nettoyage des silencieux et filtres	1 fois par mois

Tableaux 1.1 les opérations de maintenance.

5 –Conclusion

Dans ce chapitre nous avons représenté la briqueterie BABAHOUM en parallèle avec explication des différentes étapes de fabrication de briques et le rôle, et montré en bref le cycle d'usine et comment il fonctionne et d'autre part les opérations de maintenance préventive de l'entretien des différents équipements de l'unité de briqueterie.

Chapitre 02 :

**l'automate programmable
industriel**

Introduction

L'industrie X.0 s'agit d'une révolution industrielle ,auparavant l'industrie était à la main et à la vapeur est appelé l'industrie 1.0 et aussi correspond à la création de la première usine, et après l'introduction de l'électricité la vapeur et la main a été remplacées dans le fonctionnement des machines et depuis on l'appelé l'industrie 2.0, puis il est apparu l'industrie automatisé 3.0 ,et finalement est apparu l'industrie 4.0 et c'est la dernière technologie dans l'industrie et elle à été établi en l'Allemagne .

1- L'automate programmable industrielle

1.1- Définition

L'automate est un PC industriel, au contraire de PC ordinaire qu'a les entrées différentes comme clavier, Fiche USB ...etc, l'automate a des entrée 24V viennent des capteurs, et pour ce qui concerne les sorties ,le PC ordinaire a des sorties comme l'écran et l'imprimante et les icône...ect, l'automate a des sorties selon son type et elle fourni 24V OU 220V pour actionner les actionneurs et ce sera à travers des pré actionneur ,et la chose qui est en commun d'eux c'est l'unité centrale .

1.2- Les marques d'automates

Il ya plusieurs des marques par exemple : GE, Siemens, Allen Bradley, ABB, Schneider, Delta, Mitsubishi, omron ...etc.

Les plus utilisé sont Schneider et siemens et Allen Bradley et le meilleur est siemens selon l'opinion publique.

2-automate Siemens

2.1- Contrôleurs *modulaires* SIMATIC (la gamme)

"S7_1200", "ET 200 avec CPU", "S7_300", "S7_400", "S7_1500").

2.2- Les logiciels utilisés pour programmer automate siemens

Les logiciels de programmation de l'automate siemens sont : step7, TIA portal, micro Win.

-Micro Win est utilisé pour programmer les automates siemens S7200.

- Step7 est utilisé pour S7300 et S7400 et ET200.

-TIA portal est utilisé pour S7300 et S400 et S71200 et S71500.

2.3- SIMATIC S7_300

En général on y trouve S7_300 dans la majorité d'utilisation car elle est très puissante, elle contient" alimentation, UC, les entrées, les sorties ".

Pour l'alimentation on le donne 220V ou 340 V et elle nous donne 24V.

UC contient la RAM et ROM.

L'ancien S7_300 contient seulement la RAM, et il ya d'autres nouveaux types qui contiennent un logement de ee proxy, et une batterie.



Figure 2.1 : SIMATIC S7_300. [6]

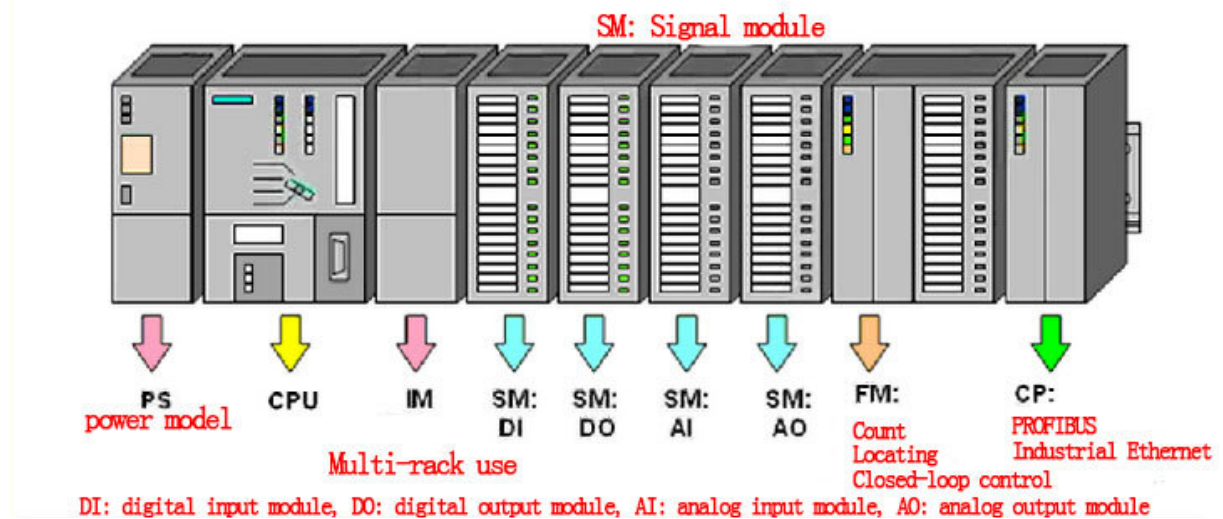


Figure 2.2 : tout l'élément général d'automate avec ordre. [7]

Fonctionnalités : Le S7-300 est une plate-forme d'automatisation universelle pour des applications avec des architectures centralisées et décentralisées, orientée sécurité, motion control ou avec interface Ethernet/PROFINet intégrée.

Le S7-300 peut également s'intégrer dans des solutions compactes avec HMI ou dans des têtes de station pour traitement intelligent décentralisé.

Caractéristiques techniques : 24 CPU standards: avec interface Ethernet/PROFINet intégrée · CPUS de sécurité · CPU compactes avec fonctions technologiques et périphéries intégrées, · CPU technologiques pour la gestion des fonctions motion control.

Modules d'E/S TOR et analogiques pour la quasi-totalité des signaux avec possibilité de traitement des interruptions et du diagnostic · Modules pour emploi dans des zones à atmosphère explosive · Modules de fonction technologique :

- ex : régulation et came électronique et des modules de communication point à point ou par bus ASi, Profibus ou Industriel Ethernet.

- Simplicité de montage et sa grande densité d'implantation avec des modules au modulo 32 permettent un gain de place appréciable dans les armoires. [2]

2.4 -Caractéristiques de l'automate S7-300

-Possède 24 sortes de CPU standards : avec interface Ethernet/PROFINET intégrée.

-CPU compactes avec fonctions technologiques et périphérie intégrées.

-Modules d'E/S TOR et analogiques pour la quasi-totalité des signaux avec possibilité de traitement des interruptions et du diagnostic [8].

3-SIMATIC S7-300 CPU315-2DP

S7-300 CPU315-2DP contient deux port (profibus) l'un d'eux peut être utilisé come profibus ou MPI.



Figure 2.3 :CPU315-2DP. [9]

3.1- Interfaces

La CPU technologique possède deux interfaces :

- _ Une interface MPI/DP paramétrable comme interface MPI ou comme interface DP (Interface maître ou interface esclave)
- _ Une interface DP(DRIVE) pour le raccordement des systèmes d'entraînement

Interface MPI/DP

L'interface MPI/DP est prévue pour le raccordement d'autres composants SIMATIC, par exemple : PG, OP, automates S7 et stations périphériques décentralisées. Son utilisation en tant qu'interface DP autorise la configuration de réseaux sur de grandes étendues géographiques.

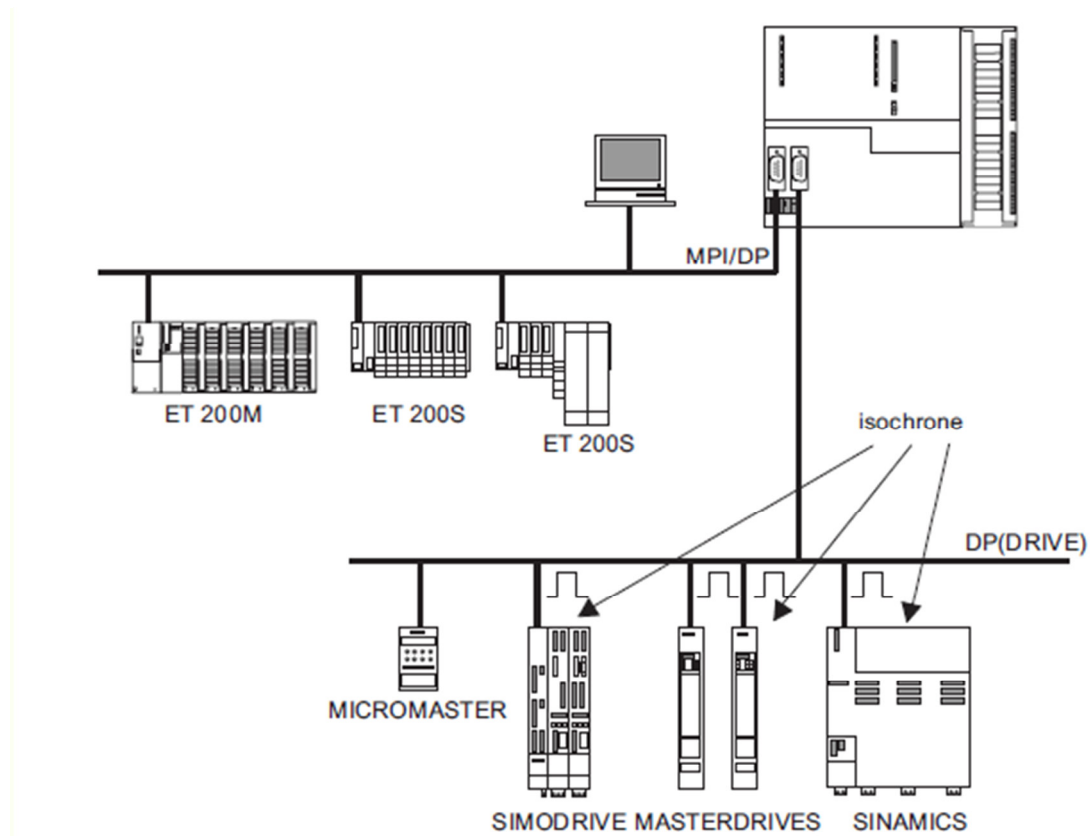


Figure 2.4: Configuration typique avec la CPU technologique.

Interface DP(DRIVE)

L'interface DP(DRIVE) est optimisée pour le raccordement d'entraînements. Elle supporte les entraînements SIEMENS les plus importants comme : MICROMASTER 420/430/440 et COMBIMASTER 411 ,SIMODRIVE 611 universal, SIMODRIVE POSMO CD/SI/CA, MASTERDRIVES MC/VC, ET 200M avec IM 153-2 (isochrone !) et SM 322 pour une sortie de came additionnelle, ET 200S avec IM 151-1 High feature, SINAMICS S120 (en option avec TM15 ou TM17 High feature pour des cames rapides), Interface d'entraînement analogique ADI4, Capteur PROFIBUS isochrone "SIMODRIVE sensor isochron" .

Les constituants configurables dans HW Config sont indiqués dans la fenêtre "Hardware Katalog" de HW Config. Sélectionnez le profil "SIMATIC Technologie-CPU" dans HW Config.

Pour que la liste de sélection soit complète dans le profil, vous devez avoir installé la dernière version de la technologie S7.

Les Entrées et sorties

La CPU technologique dispose de 4 entrées TOR et de 8 sorties TOR. Ces entrées et sorties sont utilisées pour les fonctions technologiques, par ex. référencement (cames de référence) ou signaux de commutation de came rapides. Les entrées et sorties peuvent aussi être exploitées par les fonctions technologiques dans le programme utilisateur STEP 7.

Remarque : il ya les entrées et sorties TOR (tout ou rien) et il ya les entrées et sorties analogique (elle utilise de -10V à 10V ou 0 à 20 mA ou 4 à 20 ma et cette dernière est la plus utilisé).

3.2-Configuration et programmation

La configuration et la programmation de la CPU technologique s'effectuent intégralement dans STEP 7 (à partir de la version 5.3 SP3) et dans le pack optionnel S7-Technology V3.0 (le pack optionnel S7-Technology est intégré dans STEP 7 après l'installation).

La configuration complète du matériel (par ex. la création des sous-réseaux pour les deux interfaces MPI/DP et DP(DRIVE) ainsi que la sélection des constituants d'entraînement, s'effectue avec STEP 7 dans HW Config.

Le pack optionnel S7-Technology vous est nécessaire pour paramétrer les objets technologiques, par ex. les axes, les disques-cames, les cames et les palpeurs.

Le paramétrage est réalisé dans des masques spécialement conçus à cet effet. Les données des objets technologiques sont rangées dans des blocs de données et accessibles au programme utilisateur STEP 7.

Par ailleurs, le pack optionnel S7-Technology est doté d'une bibliothèque contenant des blocs fonctionnels standards selon PLCopen pour la programmation des tâches de Motion Control. Vous appelez ces FB standard dans votre programme utilisateur STEP 7.

Pour créer le programme utilisateur STEP 7 (avec les tâches de Motion Control) vous avez à votre disposition les langages STEP 7 que sont KOP, FUP, AWL et tous les outils d'ingénierie, par ex. S7-SCL ou S7-GRAPH.

La CPU technologique supporte uniquement l'installation et la configuration à une rangée.

3.3 Eléments de commande et de signalisation

Eléments de commande et d'affichage de la CPU

La figure suivante montre les éléments de commande et d'affichage de la CPU technologique.

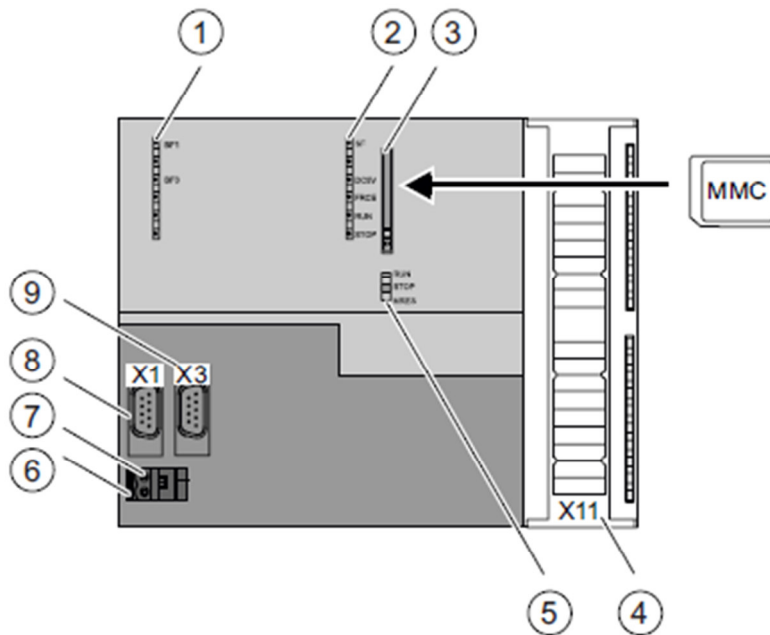


Figure 2.5 : Eléments de commande et d'affichage de la CPU technologique.

Tableau 2-1 Eléments de commande et d'affichage de la CPU technologique

La figure vous montre, sous le numéro	L'élément suivant de la CPU technologique
1	Affichage d'erreur de bus
2	Affichages d'état et de défaut
3	Logement de la micro-carte mémoire (MMC) avec éjecteur
4	Raccordement de la périphérie intégrée
5	Commutateur de mode de fonctionnement
6	Raccordement de la tension d'alimentation
7	Curseur de mise à la terre
8	Interface X1 MPI/DP
9	Interface X3 DP(DRIVE)

Pour le Logement de la micro-carte mémoire SIMATIC (MMC) On utilise une micro-carte mémoire (MMC) SIMATIC comme cartouche mémoire. Les MMC sont utilisables comme mémoires de chargement et comme supports de données amovibles.

Entrées et sorties intégrées pour technologie intégrée

Vous pouvez utiliser les entrées et sorties de la technologie intégrée pour des fonctions technologiques que vous configurez avec S7T Config (partie intégrante du pack optionnel S7-Technology).

Les sorties TOR sont prévues pour des fonctions de commutation rapides des cames et sont exploitables avec des fonctions technologiques dans le programme utilisateur STEP 7. Les entrées TOR peuvent également être exploitées dans le

programmes utilisateur STEP 7 avec des fonctions technologiques, comme par ex. la prise de référence (came de référence).

Vous utilisez les entrées et sorties intégrées pour les applications nécessitant un traitement technologique rapide. Vous câblez les autres entrées et sorties que vous désirez exploiter dans le programme utilisateur STEP 7 de la manière habituelle, avec des modules d'entrées / sorties additionnels.

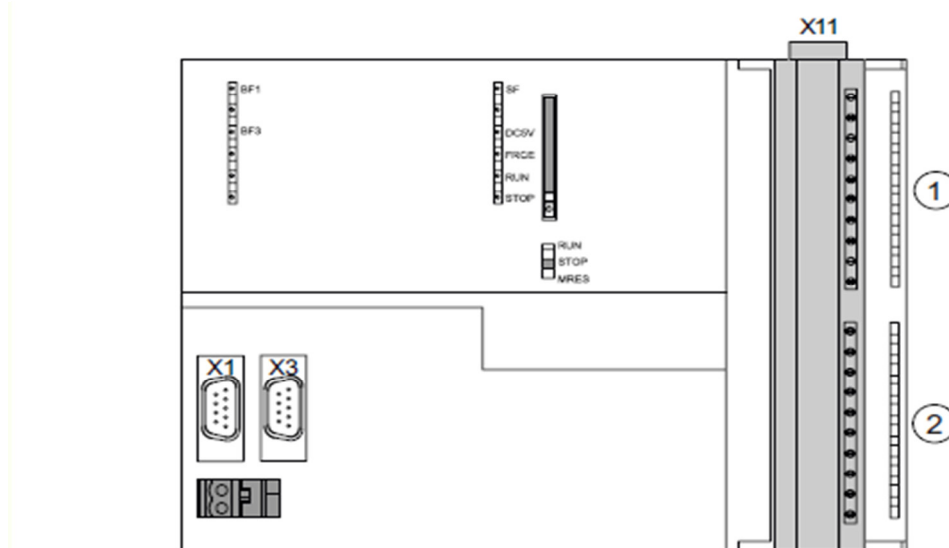


Figure 2.6 : Entrées et sorties intégrées de la CPU technologique, visibles après ouverture de la porte en façade.

Tableau 2-2 : Entrées et sorties intégrées de la CPU technologique

La figure vous montre, sous le numéro	la périphérie intégrée suivante
1	4 entrées TOR
2	8 sorties TOR

Commutateur de mode de fonctionnement

Le mode de fonctionnement actuel de la CPU est réglé avec le commutateur de mode de fonctionnement.

Tableau 2-3 : Positions du commutateur de mode de fonctionnement

Position	Signification	Explication
RUN	Mode de fonctionnement RUN	La CPU traite le programme utilisateur
STOP	Mode de fonctionnement STOP	La CPU ne traite aucun programme utilisateur.
MRES	Effacement général	Position du commutateur de mode de fonctionnement pour l'effacement général de la CPU. L'effacement général à l'aide du commutateur de mode de fonctionnement nécessite une séquence d'actions particulière.

Remarque : Chaque CPU dispose d'une prise à 2 points pour le raccordement de la tension d'alimentation. A l'état de livraison, le connecteur avec raccords filetés est déjà enfiché sur cette prise.

Indications d'état et de défauts

La CPU est dotée des LED de signalisations suivantes :

Tableau 2-4 Affichages d'état et d'erreur de la CPU

LED	Couleur	Signification
SF	rouge	Erreur matérielle ou de logiciel
BF1	rouge	Erreur de bus (MPI/DP)
BF3	rouge	Erreur de bus sur DP (DRIVE)
DC5V	vert	Alimentation 5V pour CPU et bus S7-300
FRCE	aune	Contrat de forçage actif
RUN	vert	CPU en mode RUN. La LED clignote à 2 Hz au démarrage, à 0,5 Hz en attente.
STOP	jaune	CPU en mode STOP ou HALT ou démarrage. La LED clignote • à 0,5 Hz lors de la demande d'un effacement général • avec 2 Hz lors de l'effacement général • avec 2 Hz lors de la mise à l'arrêt (la LED RUN est allumée).

Arrêter

1. Dans la phase de "mise à l'arrêt", la commande de la CPU technologique est déjà à l'état

STOP. Les sorties des périphéries centralisée et décentralisée sont désactivées au niveau des interfaces MPI/DP. La LED "STOP" clignote à 2 Hz. La LED "RUN" s'allume.

2. Les entrées/sorties intégrées pour la technologie intégrée ainsi que la périphérie décentralisée sur DP(DRIVE) sont encore actifs lors de la phase de mise à l'arrêt.

3. La technologie intégrée de la CPU technologique arrête les entraînements sur le PROFIBUS DP(DRIVE) de manière contrôlée.

4. La technologie intégrée passe ensuite aussi à l'état STOP. Les entrées/sorties intégrées pour la technologie intégrée ainsi que la périphérie décentralisée sur DP(DRIVE) sont désactivés. La LED "STOP" s'allume. La durée maximale de la phase de mise à l'arrêt dépend de votre configuration dans S7T Config.

3.4- Vue d'ensemble

3.4.1-Constituants d'un S7-300

les constituants que vous pouvez utiliser pour installer un S7-300 avec une CPU technologique dans La figure suivante que vous montre une installation en exemple :

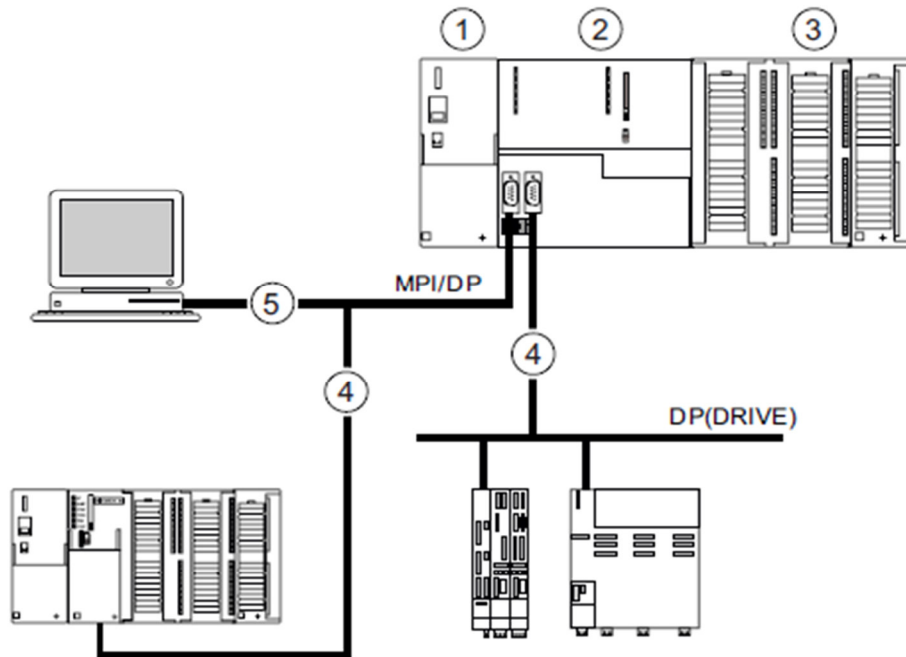


Figure 2-7 Constituants d'un S7-300.

Tableau2-5 Constituants d'un S7-300 :

La figure vous montre, sous le numéro	les constituants suivants d'un S7-300
(1)	Alimentation (PS)
(2)	Unité centrale (CPU)
(3)	Module de signaux (SM)
(4)	Câble bus PROFIBUS
(5)	Câble de raccordement d'une console de programmation (PG) ou de mise en réseau avec d'autres commandes SIMATIC.

Une console de programmation (PG) est utilisée pour la programmation du S7-300. La PG est à connecter à la CPU par un câble PG. Avec le câble de bus PROFIBUS, vous raccordez la CPU - à d'autres commandes SIMATIC via l'interface MPI/DP - à d'autres entraînements via l'interface DP(DRIVE).

3.4.2-Configuration

La CPU technologique admet uniquement une installation à une rangée.

3.4.3- Sous-réseaux

3.3.2.1- Extension et mise en réseau de sous-réseaux

Vue d'ensemble : sous-réseaux de la CPU technologique

La CPU technologique propose les sous-réseaux suivants :

- Multi Point Interface (MPI) ou PROFIBUS DP
- DP(DRIVE) : optimisé pour les entraînements

Vitesse de transmission

Les vitesses de transmission maximum mentionnées ci-dessous sont possibles :

- MPI / PROFIBUS DP : 12 MBauds

Nous vous recommandons de régler 12 MBauds pour la CPU technologique.

- DP(DRIVE) : 12 MBauds

Nombre de partenaires

Le nombre maximum de partenaires par sous-réseau indiqué ci-dessous est possible.

Tableau 2-6 Partenaires sur le sous-réseau

Paramètre	MPI	PROFIBUS DP	PROFIBUS DP(DRIVE)
Nombre	127	126	33
Adresses	0 à 126	0 à 125	1 à 125
Remarque	Par défaut : 32 adresses Sont réservées : • Adresse 0 pour PG • Adresse 1 pour OP	dont : 1 maître (réservé) 1 raccordement PG (adresse 0 réservée) 124 esclaves ou autres maîtres	dont : • 1 maître (réservé) et 32 esclaves ou entraînements

3.4.4- Interfaces

Interface MPI/DP

Vous pouvez modifier la configuration de cette interface dans STEP 7 et en faire une interface PROFIBUS DP.

L'interface multipoint (MPI) est l'interface de la CPU avec un PG/OP ou pour la communication dans un sous-réseau MPI.

L'interface PROFIBUS DP sert principalement à raccorder la périphérie décentralisée. Le

PROFIBUS DP vous permet, par exemple, de monter de vastes sous-réseaux.

Interfaces

Tableau 3-7 Modes de fonctionnements possibles des interfaces avec la CPU technologique

Interface MPI/DP (X1)	Interface DP(DRIVE) (X3)
<ul style="list-style-type: none"> • MPI • Maître DP • Esclave DP 	<ul style="list-style-type: none"> • Maître DP pour DP(DRIVE)

3.5-Adressage

Emplacements du S7-300 et adresses initiales des modules correspondants

La CPU technologique occupe 2 emplacements : numérotés 2 et 3.

Dans le cas des modules d'entrées / sorties, les adresses d'entrée et les adresses de sortie commencent ?partir de la même adresse initiale des modules.

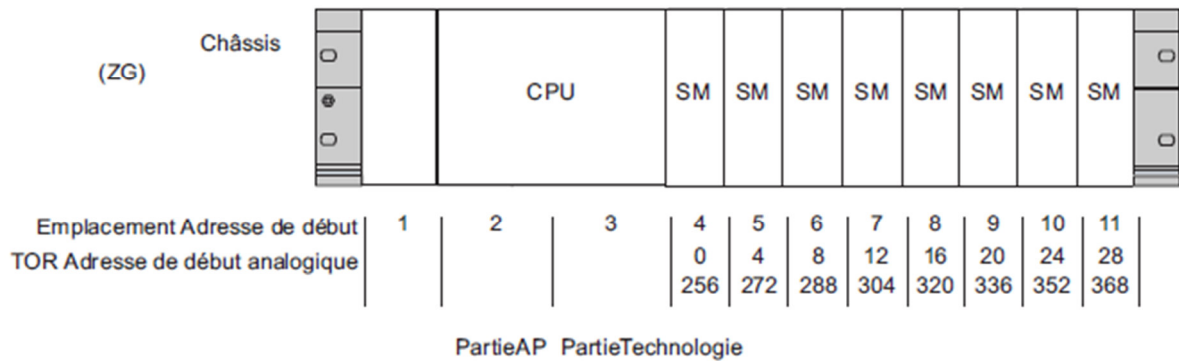


Figure 2-8 Emplacements du S7-300 avec CPU technologique et adresses initiales des modules respectifs

Entrées et sorties intégrées pour technologie intégrée

La CPU technologique comporte 4 entrées TOR intégrées et 8 sorties TOR intégrées pour la technologie intégrée. Ces entrées et sorties intégrées sont utilisées pour les fonctions technologiques, par ex. la prise de référence (cames de référence) ou les signaux de commutation de came rapides.

Les entrées et sorties intégrées peuvent aussi être exploitées avec des fonctions technologiques dans le programme utilisateur STEP 7.

Vous utilisez les entrées et sorties intégrées pour les applications nécessitant un traitement technologique rapide.

Zones d'adressage DP

Die CPU 315T-2 DP a les zones d'adressage suivantes :

- pour les entrées et les sorties respectivement : 2048 octets
- dont en mémoire image du processus, pour les entrées et les sorties respectivement : octets 0 à 127

Zones d'adressage DP (entraînement)

La CPU 315T-2 DP a les zones d'adressage suivantes :

- pour les entrées et les sorties respectivement : 1024 octets
- dont en mémoire image de la périphérie, pour les entrées et les sorties respectivement : octets 0 à 63

3.6- Mise en service

Conditions

Vous sont nécessaires pour exploiter intégralement les fonctionnalités de la CPU :

- STEP 7 à partir de V 5.3 + SP 3 et pack optionnel S7-Technology V3.0
- Le S7-300 est monté
- Le S7-300 est câblé
- Lorsque le S7-300 est mis en réseau :
Les adresses MPI/PROFIBUS sont réglées.
Les résistances de terminaison sont mises en circuit aux limites des segments. [3]

4 .Communication avec le S7-300

4.1- Interfaces

Vue d'ensemble

La CPU technologique possède deux interfaces :

- Interface MPI/DP (X1)
- Interface PROFIBUS DP(DRIVE) (X3)

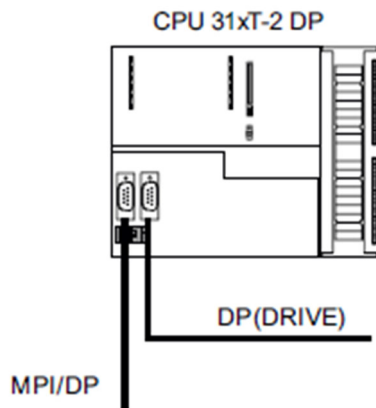


Figure 3-9 : Interfaces de la CPU technologique

4.1.2- Interface MPI/DP (X1)

Disponibilité

La CPU technologique est dotée d'une interface MPI/DP (X1). Une interface MPI/DP est toujours configurée comme interface MPI à la livraison de la CPU. Pour pouvoir l'utiliser comme interface DP, vous devez modifier sa configuration dans STEP 7 et en faire une interface DP.

Propriétés MPI

L'interface multipoint (MPI) est l'interface de la CPU avec un PG/OP ou pour la communication dans un sous-réseau MPI.

La vitesse de transmission habituelle (préréglée) pour toutes les CPU est de 187,5 kBauds.

Pour la communication avec un S7-200, vous pouvez également régler 19,2 kBauds. Vous pouvez régler la vitesse de transmission jusqu'à 12 Mbauds.

La CPU envoie automatiquement à l'interface MPI ses paramètres de bus réglés (p. ex. la vitesse de transmission). Ainsi, une console de programmation peut, par exemple, avoir les bons paramètres et se connecter automatiquement à un sous-réseau MPI.

Modes de fonctionnement de l'interface MPI/DP (X1) :

- MPI
- Maître DP
- Esclave DP
- I-Slave

Appareils raccordable via MPI

- PG/PC
- OP/TD
- S7-300/S7-400 avec interface MPI
- S7-200 (uniquement avec 19,2 kBauds)

Propriétés du PROFIBUS DP

L'interface PROFIBUS DP sert principalement à raccorder la périphérie décentralisée. Le PROFIBUS DP vous permet, par exemple, de configurer des sous-réseaux sur de grandes étendues géographiques.

L'interface PROFIBUS DP est configurable comme maître ou comme esclave et admet une vitesse de transmission pouvant atteindre 12 MBauds.

En mode maître, la CPU envoie ses paramètres de bus configurés (p. ex. la vitesse de transmission) à l'interface PROFIBUS-DP. Ainsi, une console de programmation peut, par exemple, avoir les bons paramètres et se connecter automatiquement à un sous-réseau PROFIBUS. L'envoi des paramètres de bus peut être désactivé pendant la configuration.

Appareils raccordables via PROFIBUS DP

- PG/PC
- OP/TD
- Esclaves DP
- Maître DP
- Actionneurs/Capteurs
- S7-300/S7-400 avec interface PROFIBUS DP

4.1.3- Interface PROFIBUS DP(DRIVE) (X3)

Propriétés

L'interface PROFIBUS DP(DRIVE) sert à raccorder des systèmes d'entraînement. Vous pouvez raccorder des systèmes d'entraînement selon PROFIdrive V3.0.

L'interface PROFIBUS DP(DRIVE) est configurable comme maître et admet une vitesse de transmission pouvant atteindre 12 MBauds.

L'interface PROFIBUS DP(DRIVE) supporte le synchronisme d'horloge (isochronisme).

La CPU envoie à l'interface PROFIBUS DP(DRIVE) ses paramètres de bus réglés (p. ex. la vitesse de transmission). L'envoi des paramètres de bus peut être désactivé pendant la configuration.

Avec la fonction "Routing" vous avez accès aux paramètres d'entraînement des esclaves sur la ligne DP(DRIVE) pour effectuer la mise en service et le diagnostic. Il n'est pas possible de faire le diagnostic du PROFIBUS DP(DRIVE) à partir du programme utilisateur STEP 7.

Appareils raccordables

Vous pouvez raccorder des entraînements au PROFIBUS DP(DRIVE), par ex. :

- MICROMASTER 420/430/440 et COMBIMASTER 411
- SIMODRIVE 611 universal
- SIMODRIVE POSMO CD/SI/CA
- MASTERDRIVES MC/VC
- ET 200M avec IM 153-2 (isochrone !) et SM 322 pour une sortie de came additionnelle
- ET 200S avec IM 151-1 high feature
- SINAMICS S120 (en option avec TM15 ou TM17 high feature pour des comes rapides)
- ADI4 (interface d'entraînement analogique)

- Capteur PROFIBUS isochrone "SIMODRIVE sensor isochron"

Les constituants configurables dans HW Config sont indiqués dans la fenêtre "Hardware Katalog" de HW Config. Sélectionnez le profil "SIMATIC Technologie-CPU" dans HW Config.

Pour que la liste de sélection soit complète dans le profil, vous devez avoir installé la dernière version de la technologie S7.

Appareils non raccordables

Ne pas utiliser de partenaires PROFIBUS actifs (PG, PC, OP, TD etc.) Au PROFIBUS DP(DRIVE). Si vous utilisez ces partenaires PROFIBUS sur le DP(DRIVE), vous pénaliserez le cycle DP par des temps d'accès additionnels. Le traitement isochrone des entraînements risque alors de ne pas pouvoir être assuré. [3]

5. Concept de mémoire

5.1- Zones de mémoire et rémanence

La mémoire de la CPU technologique se divise en trois zones :

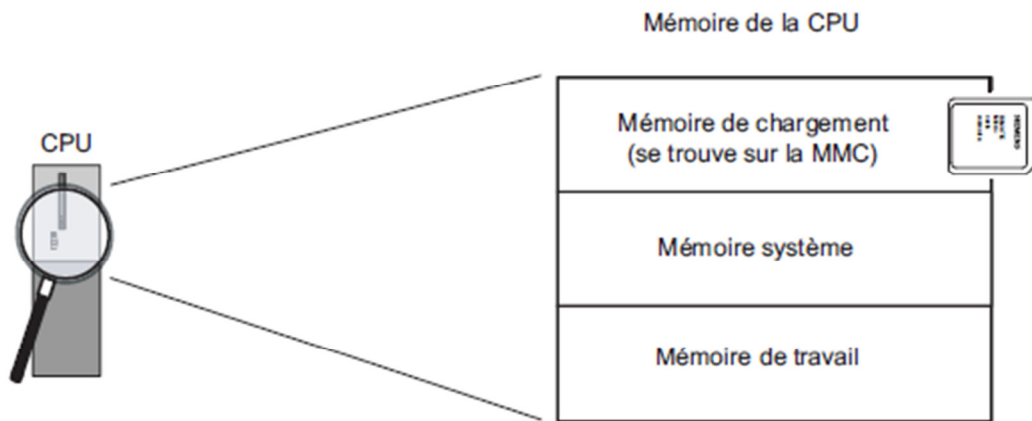


Figure 3-10 Zones de mémoire de la CPU technologique.

5.2- Plages d'opérandes de la mémoire système

Vue d'ensemble

La mémoire système des CPU S7 est répartie en plages d'opérandes (voir tableau ci-dessous). A l'aide des opérations correspondantes, vous adressez dans votre programme les données directement dans la plage d'opérandes correspondante.

Tableau 3-8 Plages d'opérandes de la mémoire système

Plages d'opérandes	Description
Mémoire image des entrées	La CPU lit au début de chaque cycle de l'OB 1 les entrées depuis les modules d'entrées et enregistre les valeurs dans la mémoire image des entrées.
Mémoire image de sorties	Le programme calcule les valeurs pour les sorties pendant le cycle et les archive dans la mémoire image des sorties. A la fin du cycle OB 1, la CPU écrit les valeurs de sortie calculées dans les modules de sorties.
Mémento	Cette plage met à disposition l'espace mémoire pour les résultats intermédiaires calculés dans le programme

Temporisations	Les temporisations sont disponibles dans cette plage.
Compteurs	Les compteurs sont disponibles dans cette plage.
Données locales	Cette plage de mémoire est réservée aux données temporaires d'un bloc de code (OB, FB, FC) pour la durée du traitement de ce bloc.
Blocs de données	Voir Recettes, Archive des valeurs de mesure et Blocs de données technologiques.

Mémoire image des entrées et des sorties

Si les plages d'opérandes Entrées (E) et Sorties (A) sont adressées dans le programme utilisateur, les états de signaux ne sont pas interrogés sur les modules de signaux TOR, mais il y a accès à une zone de mémoire dans la mémoire système de la CPU. On désigne cette zone de mémoire par mémoire image.

La mémoire image de processus est divisée en deux parties : la mémoire image des entrées et la mémoire image des sorties.

Actualisation de la mémoire image

La mémoire image est actualisée de façon cyclique par le système d'exploitation. La figure suivante présente les phases de traitement au cours d'un cycle.

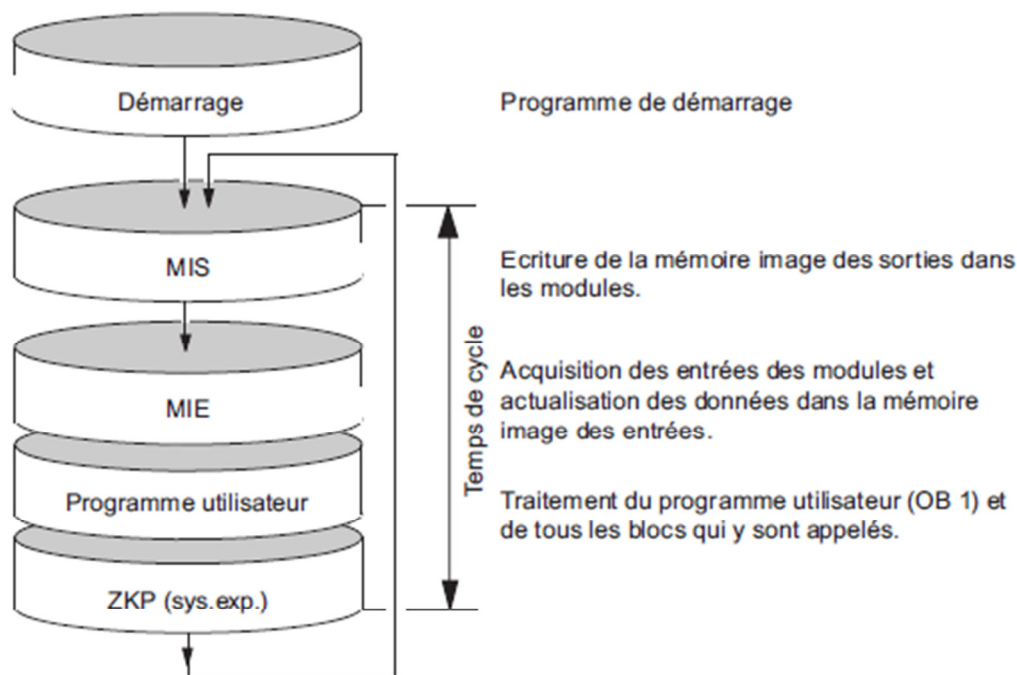


Figure 3-11 Phases de traitement au cours d'un cycle.

5.3- Fonctions de mémoire

5.3.1- Charger le programme utilisateur

Les fonctions de mémoire vous permettent de créer, de modifier ou d'effacer des programmes utilisateur entiers ou uniquement des blocs isolés. Vous pouvez en outre assurer la rémanence de vos données en archivant vos propres données de projet.

Généralités : chargement d'un programme utilisateur via PG/PC

Le programme utilisateur est chargé entièrement sur la CPU via le PG/ PC avec la MMC.

Dans la mémoire de chargement, des blocs occupent de la place, comme cela est indiqué sous "Propriétés - Bausteincontainer offline" dans le registre "Blocs" sous "Taille dans la mémoire de chargement".

Vous ne pouvez démarrer le programme que lorsque tous les blocs sont chargés. [3]

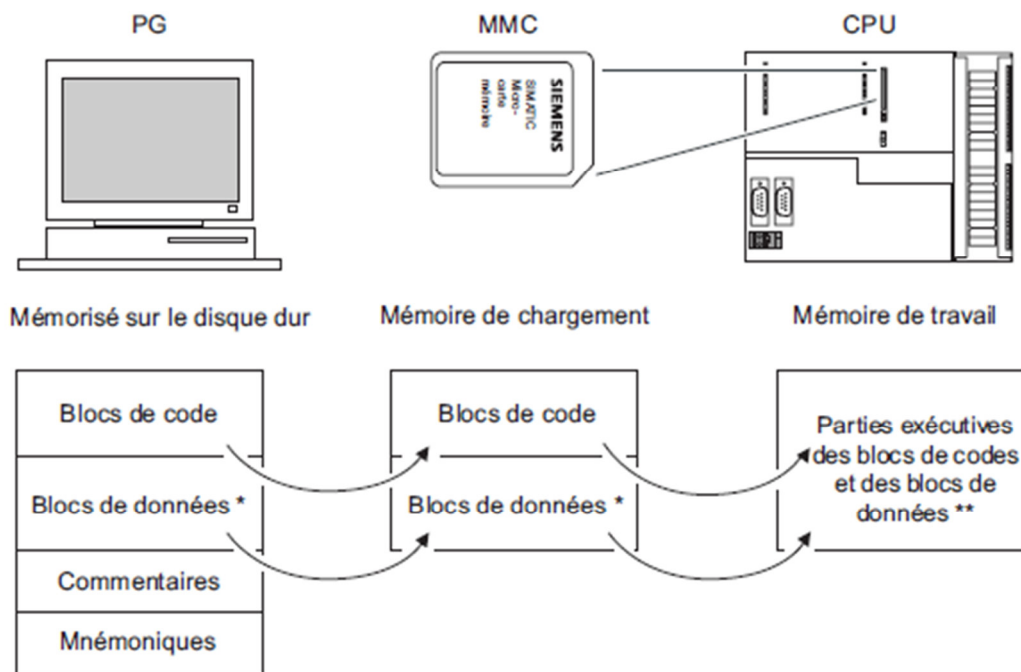


Figure 3-12 Mémoire de chargement et mémoire de travail de la CPU technologique. [3]

6-Critères de choix

Descriptif technique

Automates modulaires pour solutions système dans l'automatisation manufacturière pour l'entrée et le milieu de gamme.

Gamme des produits

- 7 CPU standard
- 7 CPU compacte
- 5 CPU de sécurité
- 2 CPU technologiques
- 1 CPU technologique de sécurité

Garantie de disponibilité des pièces de rechange

10 ans

Plage de température

0 ... 60 °C (2)

Temps d'exécution d'opérations sur bits, min.

0,004 μ s (CPU 319)

Mémoire de travail, max.

2 Mo (CPU 319),

2,5 Mo (CPU 319F)

Mémoire de chargement/de masse, max.

Micro-carte mémoire

8 Mo

Sauvegarde, max.

Programme et données par micro-carte mémoire (sans entretien) [4]

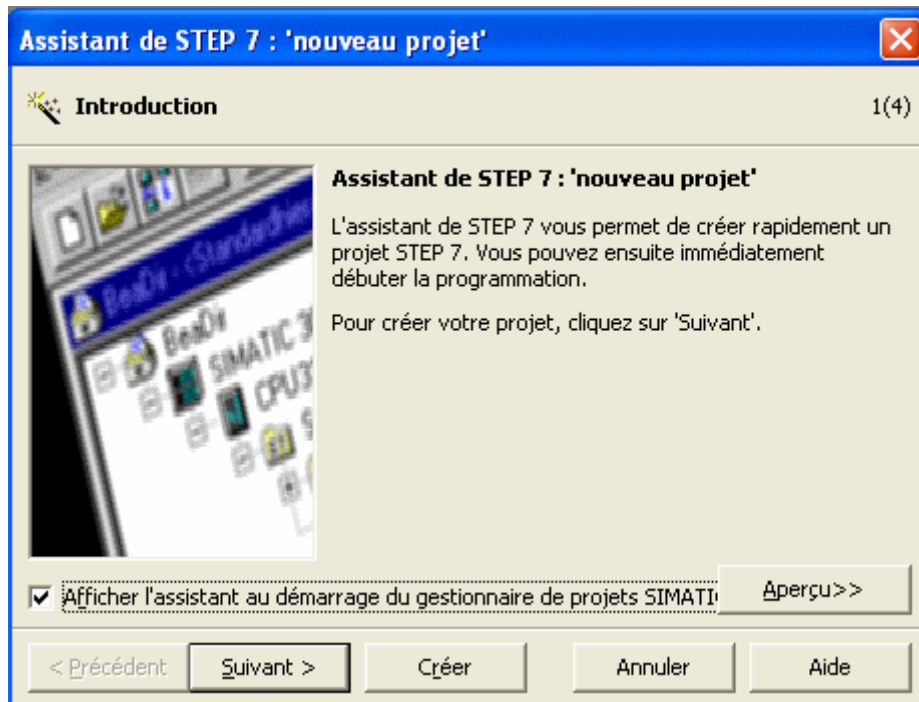
7- Utilisation de Step 7

Step 7 permet l'accès "de base" aux automates Siemens. Il permet de programmer individuellement un automate (en différents langages). Il prend également en compte le réseau des automates, ce qui permet d'accéder à tout automate du réseau (pour le programmer), et éventuellement aux automates de s'envoyer des messages entre eux. Il ne permet pas d'incorporer les ordinateurs dans le réseau (durant le fonctionnement, il n'y a pas de dialogue entre les PC et les automates, donc pas de supervision du processus par un logiciel centralisé, comme ce serait possible sous PCS7).

7.1-Créer son projet

Un projet contient la description complète de votre automatisme. Il comporte donc deux grandes parties : la description du matériel, et la description du fonctionnement (le programme).

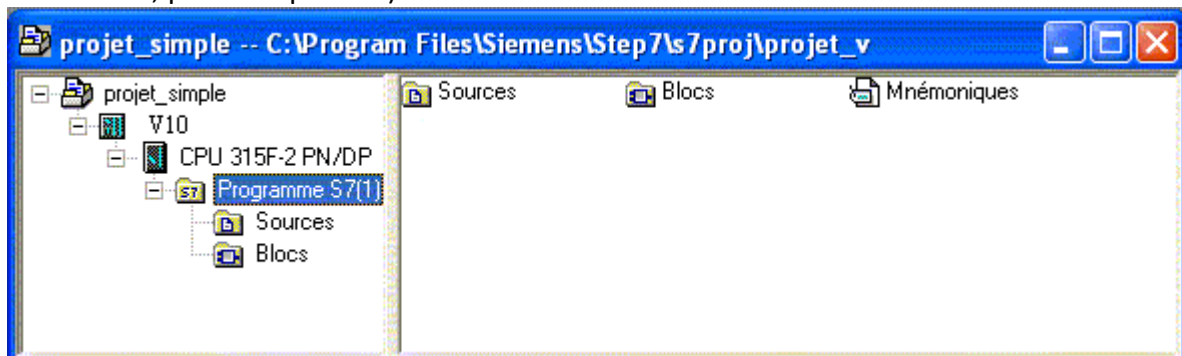
En entrant dans Step7, il peut y avoir un assistant qui vous propose de créer un nouveau projet, il vaut mieux l'annuler car par défaut il configure mal la liaison avec l'automate. On choisira donc plutôt « fichier -> nouveau » ou « fichier -> ouvrir ». N'utilisez pas un projet existant, suivant les filières les projets peuvent être incompatibles !



7.2-Le matériel

La première chose à faire est de décrire le matériel. Nous disposons de 10 PC appelés ESx (x entre 0 et 9), ES0 et ES1 pouvant servir de serveurs; et de 10 valises (comportant un automate, ses modules d'entrées/sorties, ainsi qu'une console de simulation) nommées Vx (V10 à V19). Ils sont tous reliés par un réseau industriel (PROFIBUS, câble violet) et par Ethernet (redondant, inutilisé ici).

Repérez votre PC et votre valise, leurs numéros sont indiqués sur une étiquette rouge. Pour commencer, insérez dans le projet une station SIMATIC 300 (renommez la du nom de votre valise, par exemple V10).



On définit ensuite le matériel: un rail support (à trouver dans la liste du matériel pour la gamme Simatic 300, dans les racks), puis (dans l'ordre de leur implantation physique, de gauche à droite) l'alimentation (repérez sur le matériel, en haut son type : PS 307 5A, en bas son numéro de référence : 307-1EA-0AA0). Insérez l'automate (son numéro IP 192.168.0.1xx est noté sur la valise), puis le module 32 sorties ToR, le module 32 entrées ToR, le module 8E/8S (sauf sur ES2), et enfin le module analogique. On pourrait aussi décrire son PC (mais il n'interviendra pas dans le fonctionnement final, ce sera pour PCS7). La solution la plus simple est de préparer un projet contenant la description de l'automate, sans y mettre de

programme, et l'ouvrir à chaque nouveau programme (enregistrer sous... pour garder le projet initial). Je parle évidemment de ceux qui auraient plusieurs TP successifs à effectuer ici.

Emplacement	Module	Référence	Firmware	Adresse MPI	Adresse d'entrée	Adresse de sortie
1	PS 307 5A	6ES7 307-1EA00-0AA0				
2	CPU 315-2 PN/DP	6ES7 315-2EG10-0AB0	V2.3			
X1	MF/DP				2047*	
X2	V10				2046*	
3						
4	DI32xDC24V	6ES7 321-1BL00-0AA0			0...3	
5	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0				0...3
6	DI8/DO8xDC24V/0.5A	6ES7 323-1BH01-0AA0			4	4
7	AI2x12Bit	6ES7 331-7KB02-0AB0			6...9	

Vérifiez bien les différentes adresses. Pour le CPU, en premier (X1) vous définissez la connexion PROFIBUS (à connecter en 500kbits, en format standard) : les 10 valises Vxx (V10 à V19), ont xx comme numéro PROFIBUS. En X2 (seconde interface, ici Ethernet), définissez son numéro IP (192.168.0.1xx) et donnez le nom de votre valise (par exemple V10), il ne doit plus s'appeler PN-IO (il vaut mieux ne pas activer cette liaison, Profibus suffit). Utilisez également l'adressage du tableau ci-dessus pour les modules d'E/S.

Quand nous ferons des projets PCS7 il faudra également définir le PC, nommé ESx (x entre 0 et 9), de numéro PROFIBUS x et IP 192.168.0.00x, voire plusieurs PC et plusieurs valises. Nous avons en plus un gros automate (série AS400) nommé AS400, d'adresse PROFIBUS 20, et 192.168.0.120 comme IP. Il est prévu d'y adjoindre une périphérie décentralisée mais ce n'est pas encore fait. Ci dessous sa description actuelle (à définir uniquement si vous l'utilisez) :

Emplacement	Module	Référence	Firmware	Adresse MPI	Adresse d'entrée	Adresse de sortie
1	PS 407 10A	6ES7 407-0KA01-0AA0				
3	CPU 414-3 DP	6ES7 414-3XJ04-0AB0	V4.0			
X2	DP				8191*	
X1	MF/DP				8190*	
IF1						
5	CP 443-1	6GK7 443-1EX11-0XE0	V2.5		8189	

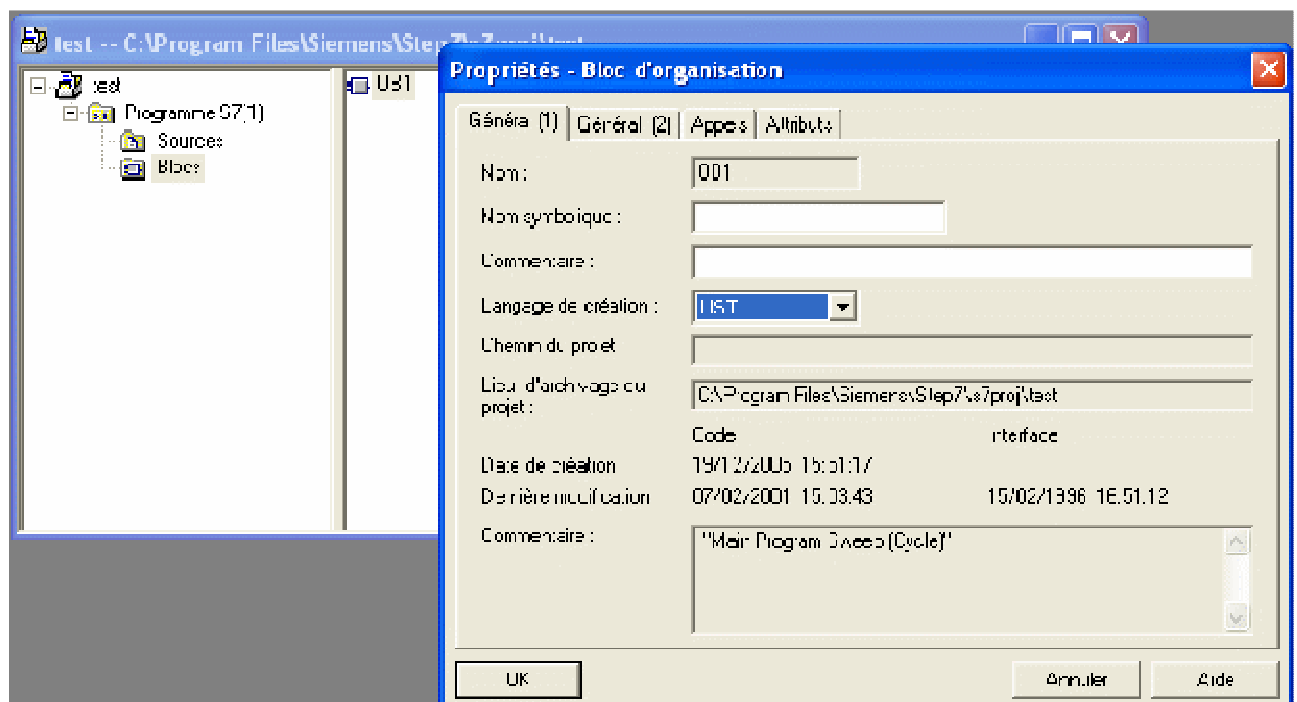
On peut (devrait, sauf dans les problèmes suffisamment simples comme le premier exercice) donner des noms explicites aux différentes E/S, en choisissant "mnémoniques" dans le dossier "programme" de l'automate. De base, les entrées ToR se notent E a.b (E=ein) avec a l'adresse du module (ou la partie d'adresse, on regroupe par octet, donc dans un module 32 E/S il y a 4 adresses a), b étant le numéro du bit dans l'octet (entre 0 et 7). Exemple : E0.4 est la cinquième entrée du premier bloc d'entrées) les sorties se notent A a.b (A=Aus). Les entrées et les sorties peuvent utiliser les mêmes adresses, les 32 entrées du premier bloc s'appellent E0.0 à E3.7, les 32 sorties du second bloc s'appellent A0.0 à A3.7. On peut également accéder directement à un octet complet (B), un mot (W) de deux octets, un double mot (D) de 4 octets. Pour stocker des résultats intermédiaires, on dispose de mémoires internes (mémento) nommés en ToR M0.0 à M65535.7 (si on a assez de mémoire), ou MB, MW, MD. Pour l'arithmétique on dispose aussi des types int, dint, real, char, date, time...

Les mnémoniques sont des variables globales (pour tous les blocs ou sous-programmes). Mais elles sont définies pour un matériel donné uniquement. Voici un exemple de table de mnémoniques :

	Etat	Mnémonique	Opéran	Type de d	Commen
15		led15	A 1.6	BOOL	
16		led16	A 1.7	BOOL	
17		octet s1	AB 0	BYTE	
18		octet s2	AB 1	BYTE	
19		octet s3	AB 2	BYTE	
20		octet s4	AB 3	BYTE	
21		double s1	AD 0	DWORD	
22		afficheur	AD 2	DWORD	
23		mot s1	AW 0	WORD	
24		mot s2	AW 2	WORD	
25		bouton1	E 0.0	BOOL	
26		bouton2	E 0.1	BOOL	
27		bouton3	E 0.2	BOOL	

7.3-Le programme

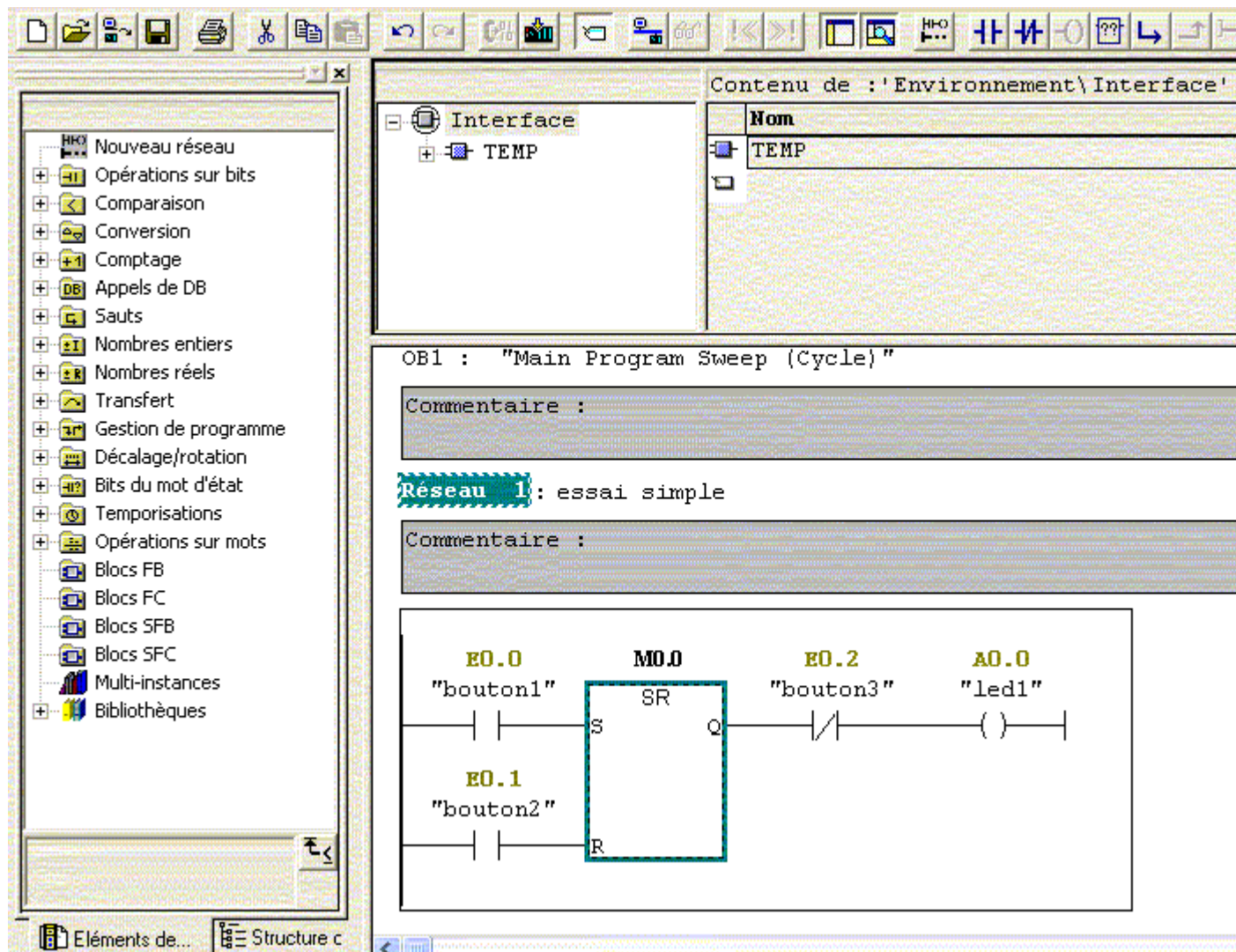
Le programme sera placé dans l'automate (->programme->blocs). Le "programme principal" s'appelle obligatoirement OB1 (OB= Bloc d'Organisation, contient un bout de programme, on pourrait aussi appeler cela un sous-programme). On double clique sur OB1 pour entrer le programme. Il faut avant tout choisir son langage préféré (dans "affichage" s'il ne le propose pas automatiquement): CONT (langage à contacts), LIST (langage textuel), ou LOG (portes logiques). D'autres langages (optionnels) existent, les trois qui me semblent les plus intéressants sont SCL (langage proche du Pascal, permettant des algorithmes et calculs complexes), GRAPH (proche du Grafcet), HiGRAPH (proche des réseaux de Petri).



7.4-Le langage CONT

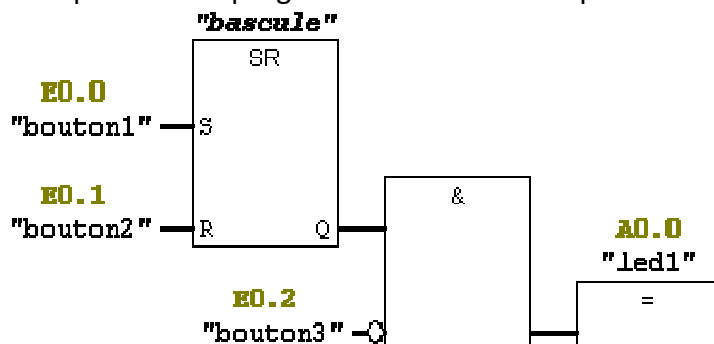
C'est une suite de réseaux qui seront parcourus séquentiellement. Les entrées sont représentées par des interrupteurs -| |- (ou -|/|- si entrée inversée), les sorties par des bobines -()- ou des bascules -(S)- -(R)-. Il y a également des opérations unaires (une entrée une sortie) : l'inverseur -|NOT|-, l'attente d'un front montant -(P)- ou descendant -(N)-. Les sorties sont obligatoirement à droite du réseau. On doit évidemment définir nos E/S, soit directement par leur code (E a.b / A a.b), ou avec leur nom en clair défini dans la table des mnémoniques (entrez le nom entre guillemets). On relie les éléments en série pour la fonction ET, en parallèle pour le OU. On peut utiliser des bits internes (peuvent servir en bobines et interrupteurs), comme on utilise dans une calculatrice une mémoire pour stocker un résultat intermédiaire (M a.b). On peut aussi introduire des éléments plus complexes, en particulier les opérations sur bits comme par exemple une bascule SR (priorité déclenchement), RS (priorité enclenchement), POS et NEG pour la détection de fronts... Dans le document en ligne « CONT pour S7 » on trouvera d'autres fonctions utiles, les compteurs, les tempos, à la rigueur le registre à décalage qui permettrait de gérer du séquentiel sans Grafset. On peut également utiliser des fonctions plus complexes (calculs sur mots par exemple), mais là je pense qu'il vaut mieux travailler en langage LIST. On trouve normalement dans la fenêtre « éléments de programme » l'ensemble des opérations existantes (si on ne l'a pas fermée, bien sûr, sinon choisir « affichage -> vues d'ensemble »).

Le programme est en général décomposé en plusieurs réseaux, par exemple un réseau par sortie (2 parties du schéma non reliées entre elles doivent être dans deux réseaux différents). Les réseaux sont exécutés séquentiellement.



7.5-Le langage LOG

C'est un langage graphique, utilisant les symboles de l'électronique numérique (portes logiques). Il n'y a rien de spécial à dire, c'est très intuitif. On peut utiliser plusieurs entrées pour une même porte, placer des inverseurs sur les entrées.... Ici, on découpe son programme en plusieurs réseaux (en général quand un ensemble de blocs n'est pas relié au reste, ou un réseau par sortie...) Voici l'exemple correspondant au programme CONT montré plus haut :



7.6-Le langage LIST

C'est un langage textuel, qui est le plus proche du comportement interne de l'automate (correspond à peu près à l'assembleur dans un ordinateur). Le système sait toujours traduire du CONT ou du LOG en LIST, mais pas l'inverse (ou alors je ne sais pas faire). Le programme se compose d'une suite de lignes, chacune spécifiant un code opération suivi d'un opérande (et un seul). L'opérande peut être une adresse absolue (E0.0) ou un mnémonique entre guillemets (si les mnémoniques ont été définis, bien sûr). Comme on ne peut pas utiliser deux opérandes dans une même ligne, pour faire « x=a et b » on écrit :

```
U "a"  
U "b"  
= "x"
```

On utilise U pour ET (und), O pour Ou (oder), X pour Ou Exclusif), UN, ON et même XN pour les entrées inversées, = pour stocker le résultat. L'opération (O ou U) pour le premier opérande n'a pas grande importance. Pour une bascule on utilisera S et R :

```
U "a"  
S "x"  
U "b"  
R "x"
```

NOT inverse le résultat précédent, FP indique si le résultat précédent vient de passer de 0 à 1 (front montant), FN pour le front descendant. Ici aussi, on peut décomposer le programme en plusieurs réseaux, mais peut-être est-ce moins courant qu'en CONT ?

On peut, comme en programmation classique, faire des sauts (goto) : SPA label (inconditionnel), SPB (si dernier calcul = 1), SPBN (si 0). On saute à une ligne précédée de « label : »

En LIST, on peut aussi traiter des mots (peut-être plus facilement que dans les autres langages) : octets (B), entiers de 16 bits (W) ou 32 bits (D), réels 32 bits). On utilise un accumulateur sous forme d'une pile (à 2 niveaux). On empile une valeur par L (load). Les calculs se font entre les deux accumulateurs, se notent pour les entiers +I, -I, /I, *I (idem +D,... sur 32 bits, +R réels). On peut comparer les deux derniers niveaux de la pile par <I, >I, <=I, >=I, =I, <>I (idem <D pour les doubles ou <R pour les réels).

Attention, l'écriture est plus proche de la notation polonaise que d'une calculatrice : on charge les deux opérandes puis seulement on dit l'opération à effectuer.

Exemple (sur des entiers 16 bits) :

```
L   MW0           //charger le mot double  
L   130           //charger la constante  
>R                //tester  
=   A0.0          //s'allume si MW0 était supérieur à  
130
```

On sauve le sommet de la pile par T (T MW0 par ex). Sur les réels on peut faire de nombreux calculs : ABS, SQR, SQRT, LN, EXP, SIN, COS, TAN, ASIN, ACOS, ATAN.

Les conversions de formats de mots sont possibles : ITD ou DTI (16 à 32 bits), DTR ou RND, TRUNC (entier à réel). On note B les nombres en BCD, toujours sur 32 bits. Ils sont entre -999 et plus 999, les 3 quartets de droite pour les 3 chiffres, celui de

gauche pour le signe (0 = positif, F = négatif). On traduit par ITB ou BTI, DTB ou BTD. C'est avec cela que vous gérerez l'afficheur numérique de la valise.

```
L   EW6           //acquérir l'entrée numérique
L   W#16#FFF      //masque pour ne garder que 12bits,
la capacité du convertisseur
UW          //masquage
ITB         //traduire en BCD
T   AW2         //transférer sur l'afficheur
```

Les constantes sont par défaut en décimal. On peut insérer des «_» non significatifs mais pas d'espaces (1234 ou 1_234 mais pas 1 234). On peut utiliser une autre base X en commençant par X# (2#0110_1100 16#1ABA6). Les réels contiennent un «.», et peuvent être suivis d'un exposant (puissance de 10) : 1.2E-5. Les caractères sont entre quotes (simple apostrophe), le «\$» est un caractère d'échappement, c'est à dire qu'il permet d'entrer un caractère spécial qu'un ne pourrait pas entrer autrement (\$', \$\$, \$N pour nouvelle ligne...) ou directement son code en hexa (\$4E pour 'N').

Le système gère automatiquement le type de la constante, si on veut imposer son type il suffit de la précéder par TYPE#: BOOL (bool#true, Bool#FALSE, BOOL#1), BYTE ou CHAR (BYTE#16#FF, BYTE#'a', BYTE#127), INT ou WORD ou W (W#16#FFFF), DINT ou DWORD ou DW voire REAL (car les réels sont stockés sur 32 bits).

Les constantes de dates débutent par DATE# (ou D#), TIME# (T#) DATE_AND_TIME# (DT#) ou TIME-OF_DAY# (TOD#). La date est sous la forme année-mois-jour (D#2006-06-28), l'heure h:mn:s.ms (les derniers sont optionnels) (DT#1995-02-02-11:11:11) Pour les durées, on utilise TIME dans le format définit plus haut, ou on précise les unités (on peut alors prendre des nombres réels pour le dernier) : D (jour)H (heure) M S MS (T#1d3h2m, T#2.5s, T#150ms). Il y a aussi des formats hérités de S5.

Le Graph (langage proche du Grafcet mais moins puissant) n'est pas un langage de base dans S7. Mais il suffit de créer un bloc fonctionnel (FB) en Graph: on se met au niveau des blocs et on insère (clic droit) un FB (bloc fonctionnel). Il nous ouvre une fenêtre de propriétés, où l'on choisit le nom (FB1 est très bien) mais surtout le langage (choisir GRAPH). Le système crée automatiquement un DB (les données associées, DB1), un FC72 et un SFC64 (fonctions système nécessaires). Il n'y a plus qu'à rentrer le programme, le sauver. Vous aviez également le droit de préciser des mnémoniques. Le système me semble nécessiter des convergences en ET symétriques aux divergences (mais accepte les étapes initiales multiples). Par contre pour les OU il m'a l'air un peu plus souple. Il ne respecte pas la norme, en particulier les OU sont exclusifs (si deux voies sont possibles, seule la plus à gauche est empruntée), la règle 5 est bafouée (si une étape doit être activée et désactivée en même temps, il la désactive !), les simultanités sont farfelues.

Une étape est définie par un identificateur Sx (x numéro unique), un nom (par défaut StepX) qui servira pour les synchronisations, et une extension (commentaire) noté à droite. On clique avec le bouton droit sur cette extension pour demander l'insertion d'un objet (une action). Celle-ci est définie par un code sur une lettre (N normal : sortie allumée au début de l'activation et éteinte à la désactivation, S set : sortie mise à 1, R Reset : sortie éteinte, D délai : allumage au

bout d'un certain délai après l'activation, extinction à la désactivation, il y a d'autres options comme compteurs...).

Pour les transitions, on peut choisir le langage CONT ou LOG (j'utilise CONT). On peut insérer très facilement une tempo (Step12.T<10s), vérifier l'activation d'une étape (Step43.X). Par contre les fronts ne sont pas proposés de base. Pour y remédier, on peut par exemple rajouter une étape, on attend d'abord l'état 0 puis l'état 1.

Avant de terminer, il faut définir l'OB1 (programme principal), qui doit appeler notre FB. Je conseille d'entrer dans l'OB1 en langage CONT, d'insérer (voir fenêtre à gauche) le bloc FB en double cliquant sur FB1, donnez un nom de DB associée sur la boîte (DB1), et marquer M0.0 devant l'entrée INIT_SQ du bloc.

Il ne reste plus qu'à sauver, charger l'automate (voir ci-dessous) et tester.

Transfert vers l'automate

Après avoir enregistré votre projet (vous aviez le droit de le faire plus tôt, si vous êtes prudent), il faut transférer le projet dans l'automate (il vaut mieux que l'automate soit en mode STOP, mais en RUN-P l'automate peut être arrêté temporairement à distance). Il suffit de choisir « système cible -> charger ». On peut regarder le programme actuellement dans l'automate (s'il est en mode RUN ou RUN-P) par « affichage -> en ligne » (hors ligne correspond au projet que l'on est en train de créer sur le PC). On peut même directement modifier un programme dans la fenêtre « en ligne » (si l'automate est au repos), voire faire du copier-coller ou glisser entre la fenêtre en ligne et hors ligne.

Dans la fenêtre « en ligne », en entrant dans le programme (OB1 ou autres blocs), on peut directement visualiser l'état des variables dans le programme. On choisit pour cela « test -> visualiser ». En CONT, les schémas deviennent en pointillés aux endroits où « le courant n'arrive pas ». En LIST, un tableau est affiché à côté du programme, spécifiant la valeur (0 ou 1) des opérands, en LOG des 0 ou 1 sont écrits sur les liaisons. En Grafcet, les étapes actives sont en vert, les transitions validées sont montrées comme dans le langage correspondant, les valeurs des tempos, compteurs... sont notées à côté du schéma.

On peut également lister l'état de toutes les variables, voire les modifier. Pour cela, se placer sur les blocs (fenêtre gauche du projet), puis dans la fenêtre droite (il s'y trouve au moins OB1) cliquer avec le bouton droit et insérer une table des variables (VAT).

PLCSIM est un logiciel de simulation d'automates livré avec STEP7. On peut donc tester un programme sur un PC non relié aux automates (mais avec STEP7 installé, évidemment). Pour tester un programme, il n'est pas nécessaire d'avoir défini de matériel. Sinon, enregistrez-sous, puis supprimez la description du matériel (cliquez dessus avec le bouton de droite par exemple), et répondez NON quand il vous demande s'il faut également supprimer le programme. Démarrez le simulateur (outils -> simulation de modules ou l'icône représentant l'automate virtuel dans un nuage), affichez les E/S (insertion ->entrées ou sorties). Transférez le programme (par exemple par « système cible ->partenaires accessibles » et un copier-coller). Vous pouvez désormais tester (en mode RUN).

7.7-Transfert vers l'automate

Après avoir enregistré votre projet (vous aviez le droit de le faire plus tôt, si vous êtes prudent), il faut transférer le projet dans l'automate (il vaut mieux que

l'automate soit en mode STOP, mais en RUN-P l'automate peut être arrêté temporairement à distance). Il suffit de choisir « système cible -> charger ». On peut regarder le programme actuellement dans l'automate (s'il est en mode RUN ou RUN-P) par « affichage -> en ligne » (hors ligne correspond au projet que l'on est en train de créer sur le PC). On peut même directement modifier un programme dans la fenêtre « en ligne » (si l'automate est au repos), voire faire du copier-coller ou glisser entre la fenêtre en ligne et hors ligne.

Dans la fenêtre « en ligne », en entrant dans le programme (OB1 ou autres blocs), on peut directement visualiser l'état des variables dans le programme. On choisit pour cela « test -> visualiser ». En CONT, les schémas deviennent en pointillés aux endroits où « le courant n'arrive pas ». En LIST, un tableau est affiché à côté du programme, spécifiant les valeurs (0 ou 1) des opérandes, en LOG des 0 ou 1 sont écrits sur les liaisons. En Grafcet, les étapes actives sont en vert, les transitions validées sont montrées comme dans le langage correspondant, les valeurs des tempos, compteurs... sont notées à côté du schéma.

On peut également lister l'état de toutes les variables, voire les modifier. Pour cela, se placer sur les blocs (fenêtre gauche du projet), puis dans la fenêtre droite (il s'y trouve au moins OB1) cliquer avec le bouton droit et insérer une table des variables (VAT).

PLCSIM est un logiciel de simulation d'automates livré avec STEP7. On peut donc tester un programme sur un PC non relié aux automates (mais avec STEP7 installé, évidemment). Pour tester un programme, il n'est pas nécessaire d'avoir défini de matériel. Sinon, enregistrez-sous, puis supprimez la description du matériel (cliquez dessus avec le bouton de droite par exemple), et répondez NON quand il vous demande s'il faut également supprimer le programme. Démarrez le simulateur (outils -> simulation de modules ou l'icône représentant l'automate virtuel dans un nuage), affichez les E/S (insertion ->entrées ou sorties). Transférez le programme (par exemple par « système cible ->partenaires accessibles » et un copier-coller). Vous pouvez désormais tester (en mode RUN). [5]

8-conclusion

Dans ce chapitre, nous fournissons une brève définition de l'automate en général, puis nous nous sommes déplacés spécifiquement vers la famille siemens s7-300 et plus spécifiquement le model 315-2dp

Tenant compte du fait que tout ce qui concerne 315-2dp appartient à la famille s7-300 aussi et nous avons expliqué comment elle fonctionne, et Ensuite, nous sommes passés au logiciel step 7 qui est utilisé pour programmer l'automate siemens s7-300.

Chapitre 03:

la pince dépilleur

Chapitre 3 : la pince dépileur

Introduction

Dans ce chapitre on va faire la programmation du premier élément de la zone dépileur après la représenter, et ça sera par simens s7-300 et on va construire le grafcet et puis faire le programme par step7.

1 .description de la pince

La pince DP c'est la première machine de la zone de dépilage, qui est responsable du vidage des wagons.

La figure ci-dessus montre la structure générale de la pince où les flaches représentent les trois têtes :



Figure 3.1 : la pince dp.

Le départ cycle de la première pince dépend de la présence wagon est la présence matière fini.

La pince DP commence à fonctionner quand le wagon arrive et actionner la fin de course qui positionner au dessus de ce dernier là et il faut que le wagon est plein de brique qui sera détecté par une cellule comme il est présenté dans les trois figure ci-dessus :



Figure 3.2 : fin de course de wagon.



Figure 3.3 : wagon pleine de brique.



Figure 3.4 : cellule que détecte la présence de brique.

2. structure général de la pince DP

La pince DP a deux degré de liberté x y, elle est construite premièrement de trois têtes qui ferme / ouvre et tourner et monter et descende par une moteur triphasé sur chaque tête pour la rotation et deux vérin qui relier les doigts de chaque tête pour la fermeture et l'ouverture et une moteur triphasé pour levage come les figure suivant sont démontre.



Figure 3.5 : moteur triphasé au dessous de la tête 2.



Figure 3.6 : deux vérins en parallèle entre les doigts.



Figure3. 7 : Moteur levage.

Aux niveaux de ces composants et ses fonctionnements il ya fin de course et des détecteurs pour sécurité et fonctionnement.

Il ya une chaîne pour levage que attacher à la moteur et quand la chaîne a été cassée ou relaxant elle touche la fin de course et le sicle s'arrêter.

Et pour les têtes il ya deux capteur des deux cotés de chaque tête que détecter la position de rotation, et pour les doigts il ya a chaque tête un capteur que s'allume quant elle ouvert.



Figure3. 8 : Détecteur fin de course sécurité chaîne.

Remarque : Il ya deux axes pour levage.



Figure3.9 : détecteur 24v pour rotation de la tête.



Figure3. 10 : détecteur 24V pour l'ouverture.

Aux niveaux de translation il ya un moteur triphasé au dessous de la pince que faire la translation avant et arrière ou niveaux de l'axe général que la pince est pendant come le figure suivant représenter.



Figure3.10 :l'axe général de translation.



Figure 3. 11 :moteur triphase pour translation la pince.

Il ya sur l'axe deux capteurs a en parallèle de sécurité pour limiter le domaine de translation possible et utiliser pour faire les références.



Figure 3.12 : deux capteurs de sécurité pour limite de translation.

La fonction et le cycle du pince et associer al les variateurs et tous les moteurs de cette pince fonction avec codeur, le codeur donne la position a l'automate, précisément le codeur est relié à le variateur et le variateur à l'automate.



Figure 3.13: les variateurs de la pince.



Figure3.14 :l'automate de la pince dp (siemens S7 300 15-2dp).

3- Cahier des charges

on suppose que le pince n' est pas en fonction et n'est pas automatisé

Wagon arrive à sa place chargée par briques 3 rangé et 4 colonne chaque couche composé de 2 rams.(voir figure 3).

Le pince est dans sa position inustial ,la capteur 0° de rotaion est active ,le codeur de translation est moins de 200 et celle de monte /descente est 100 ,les 3 tetes sont ouvert .

Le pince avance à les couches des briques Dans l'ordre spécifié dans le tableau suivant :

Chapitre 3 : la pince dépileur

Ranger colonne	1	2	3	4
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

Tableau 3-1 l'roder des couches.

Pince avance a la couche

Faire stop et ensuite descend à la couche.

Faire stop et ensuite ferme les 3 têtes.

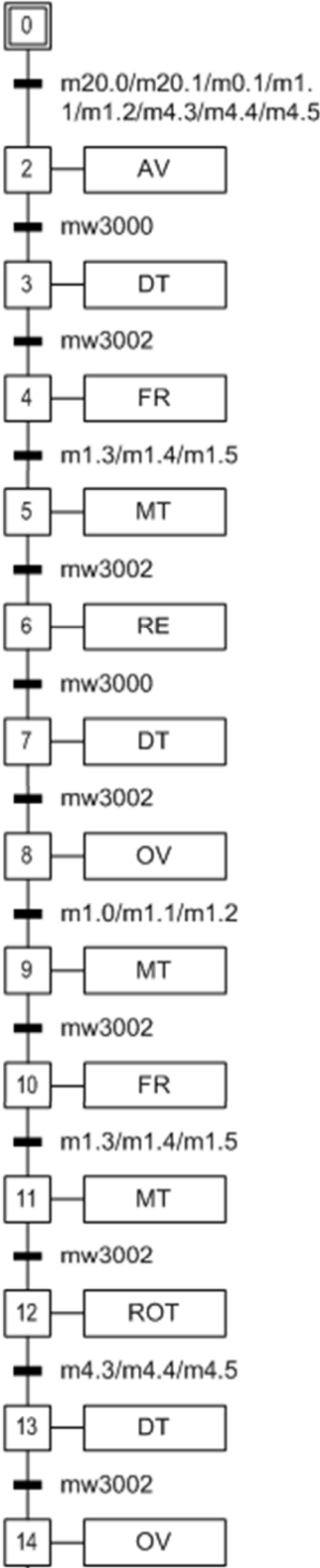
La pince monte a position 100 et ensuite recule vers les rouleaux.

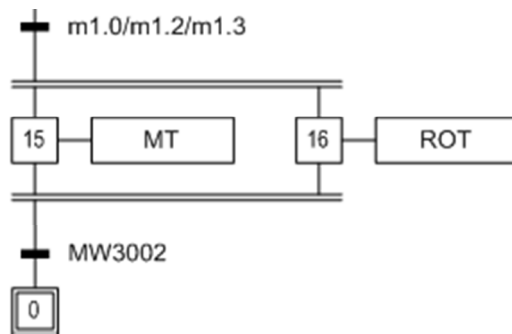
Ensuite la pince descend à 180 et ouvre, puis monte à 170 et ferme les têtes, puis monte à 150 et faire la rotation à 90° et puis il descend à 180 et ouvre les têtes.

Ensuite faire la rotation vers 0° et en le même temps il monte à 100.

Et répète le même cycle vers les couches suivants Chacun dans sa position.

4-grafcet





Mnémonique des etaps et transtions :

M20.0 =presence brique /m20.1= presence wagon/m1.1 =tete 1 ouvert /m1.2= tete 2 ouvert /m1.3= tete 3 ouvert /m4.3= tete 1 est 0 degré/m4.4=tete 2 est 0 degré/m4.5 =tete 3 est 0 degré ./mw3002 =valeur codeur de monte et descent /mw3000=valeur codeur de translation.

AV=avance pince ou avance mouteur

DT=descent pince ou mouteur

FR=fermer les tetes

MT= monte pince ou mouteur

RE=recul pince ou mouteur

OV= overir les tetes

ROT=rotation des tetes

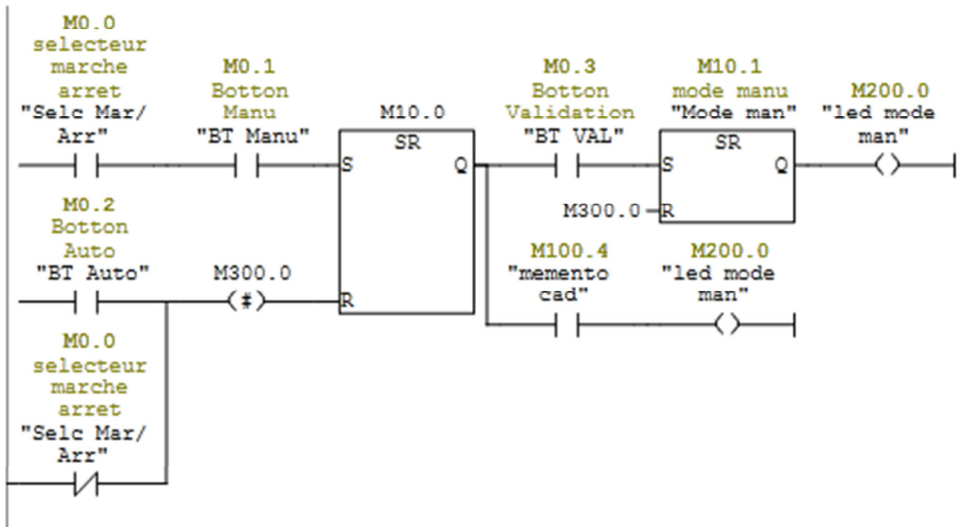
Chapitre 3 : la pince dépileur

4- exemple de programme de quelque parts de cycle du pince

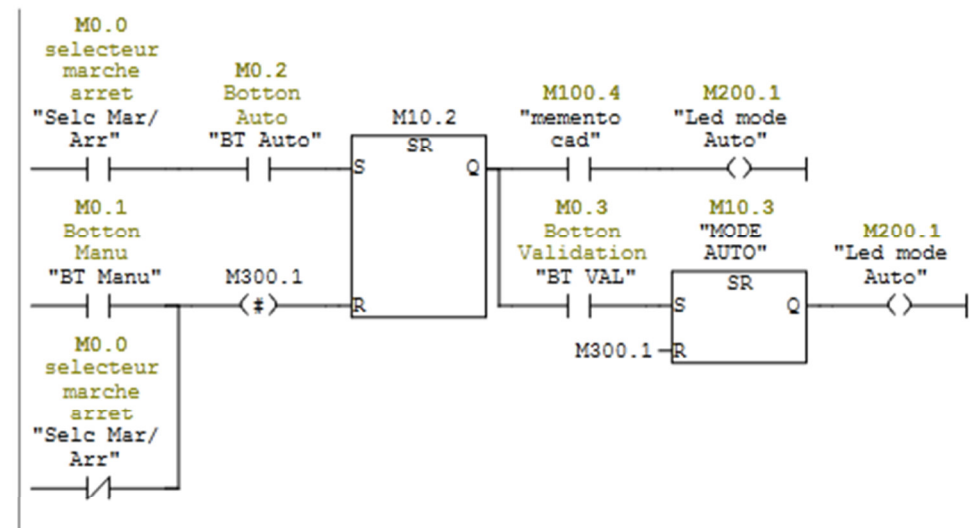
4-1. sélection de mode auto ou mode manuel

Bloc : FC1 Selection du mode
Pour selectioner le mode Manu ou le mode Auto

Réseau : 1 Mode manu

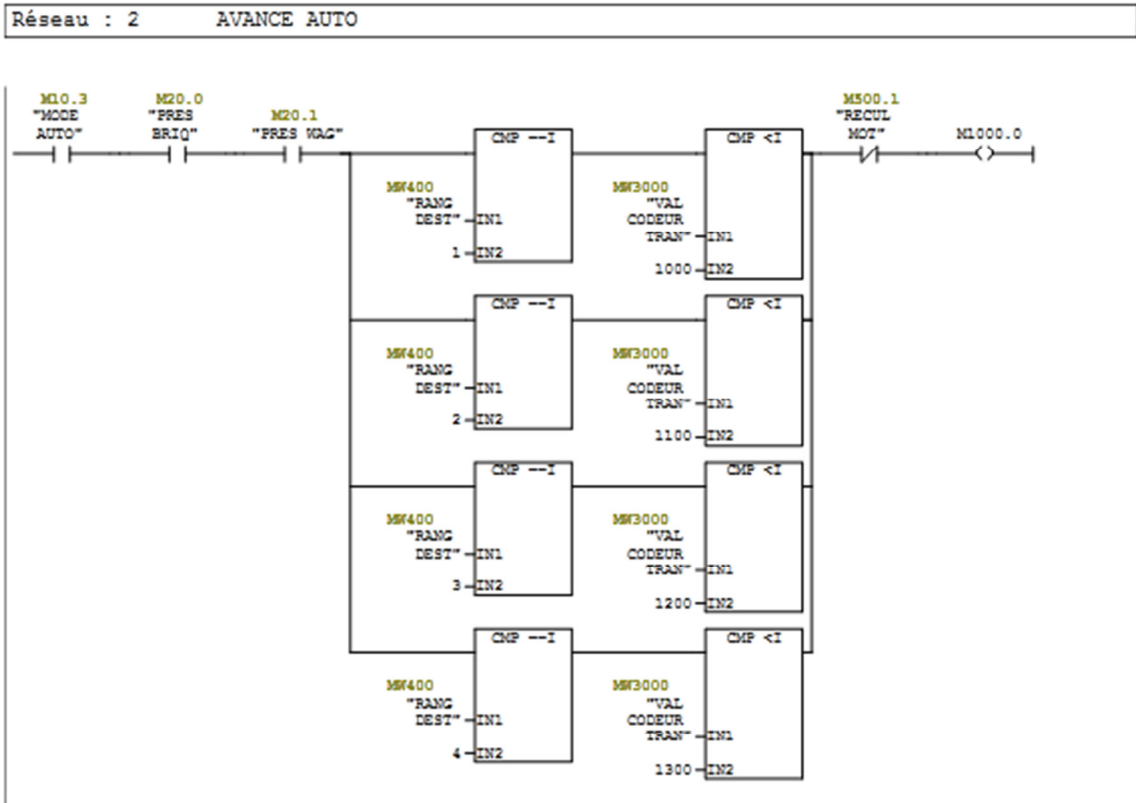


Réseau : 2 MODE AUTO

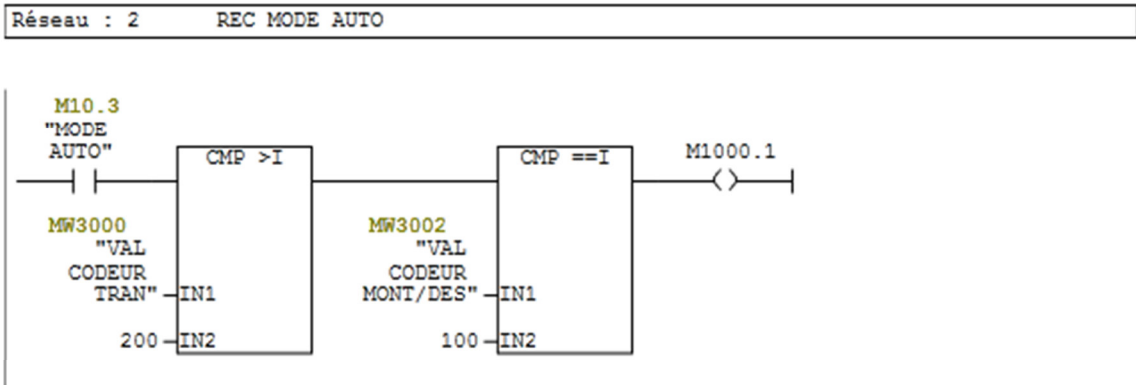


Chapitre 3 : la pince dépileur

4-2. Avance pince en mode auto



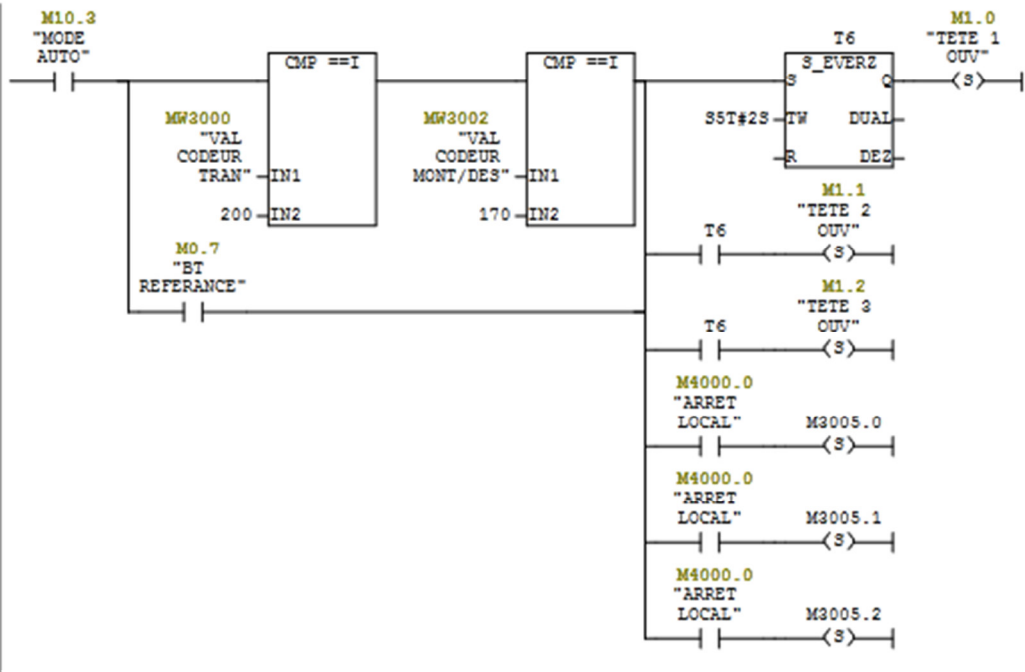
4-3. recule pince en mode auto



Chapitre 3 : la pince dépilleur

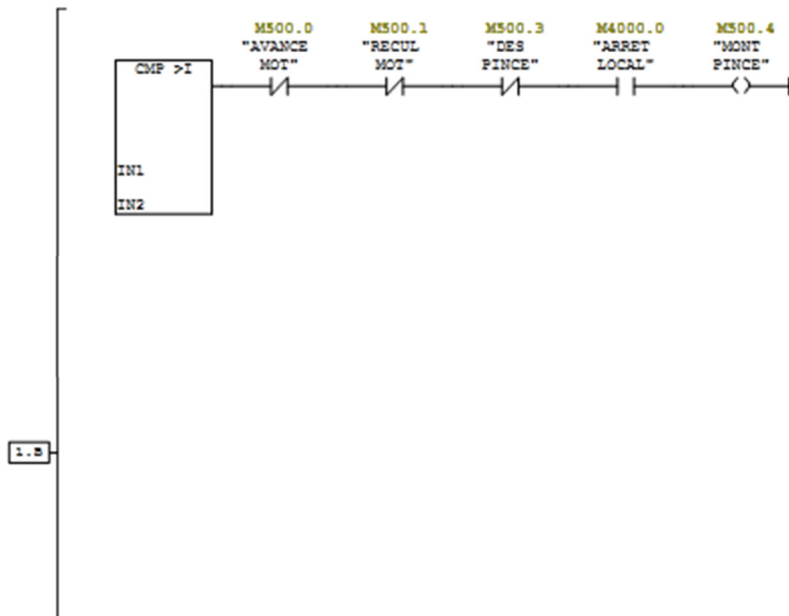
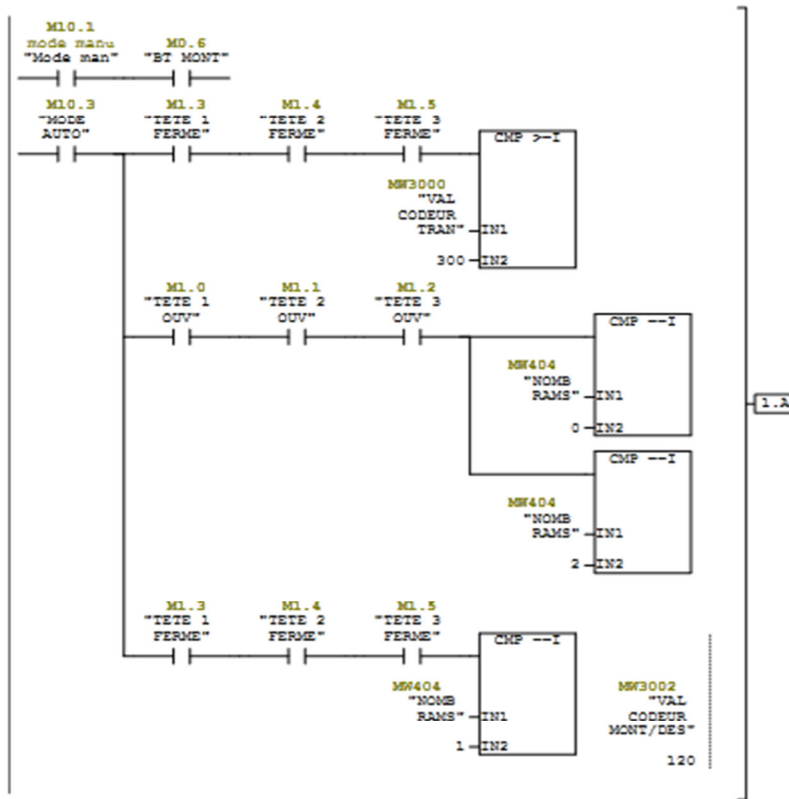
4-4.ouverture de pince en mode auto

Réseau : 3 OUVERTURE MODE AUTO



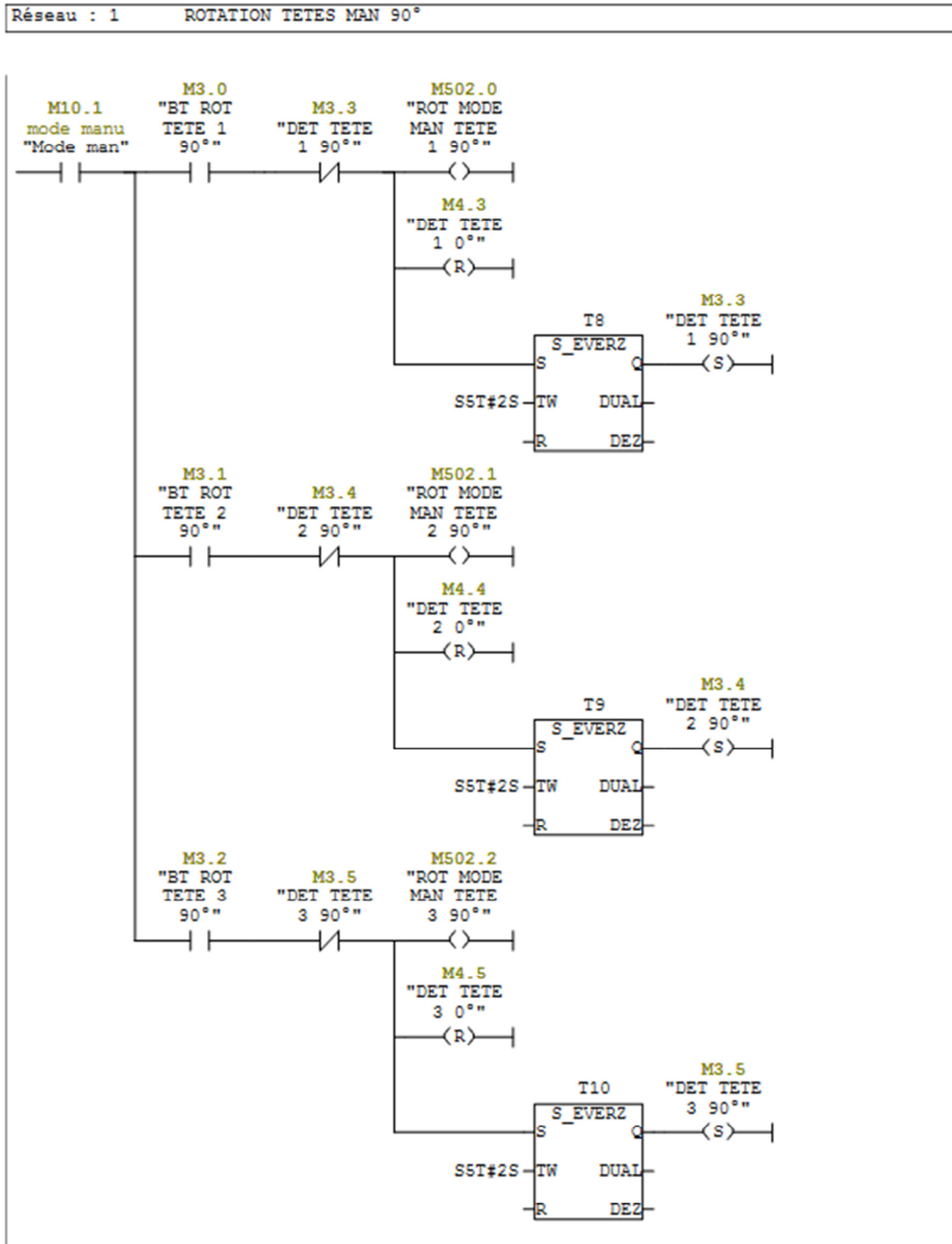
Chapitre 3 : la pince dépileur

4-5 monte pince



Chapitre 3 : la pince dépilleur

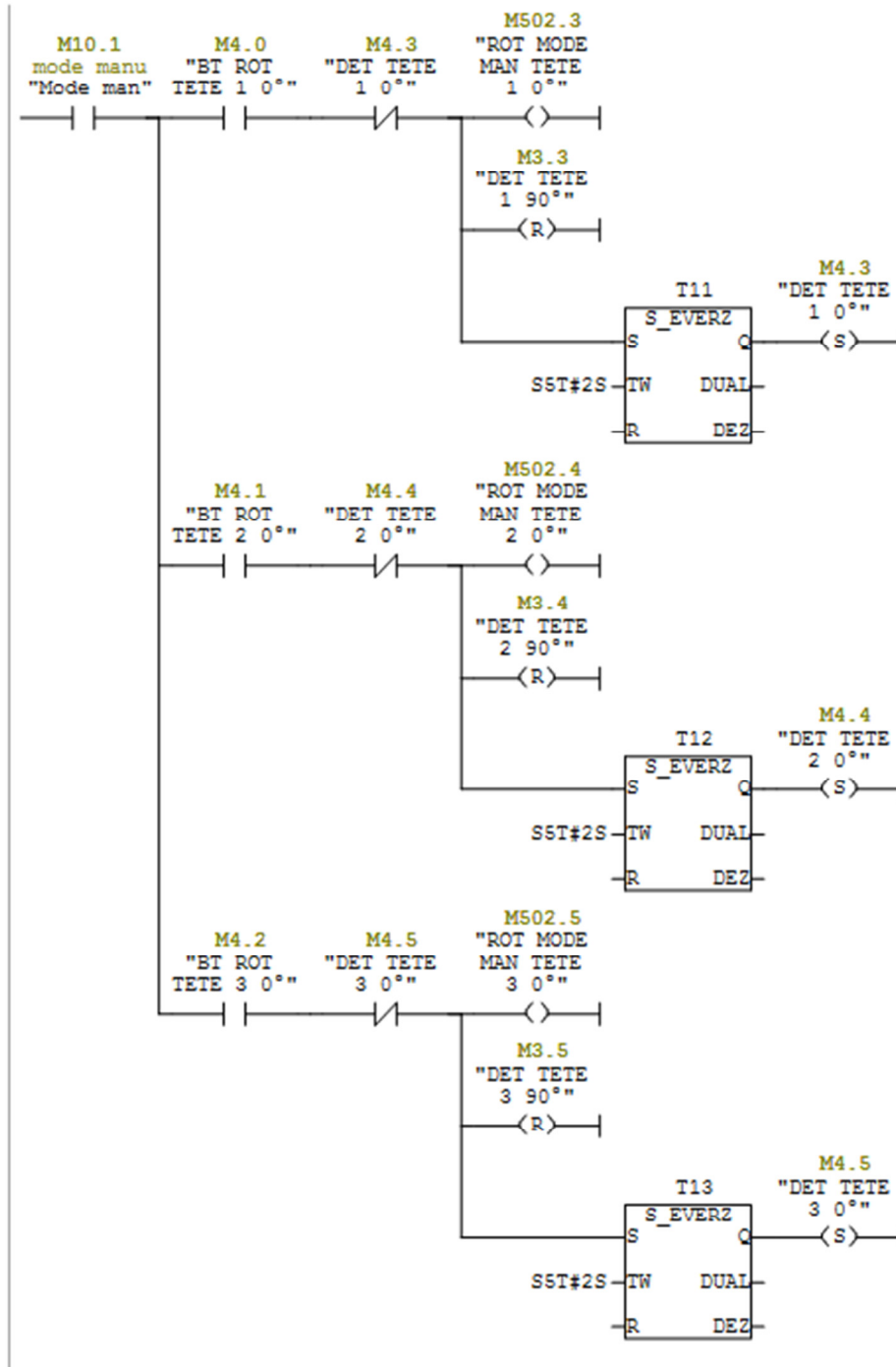
4-6 rotation tetes manuel 90 degré



Chapitre 3 : la pince dépilleur

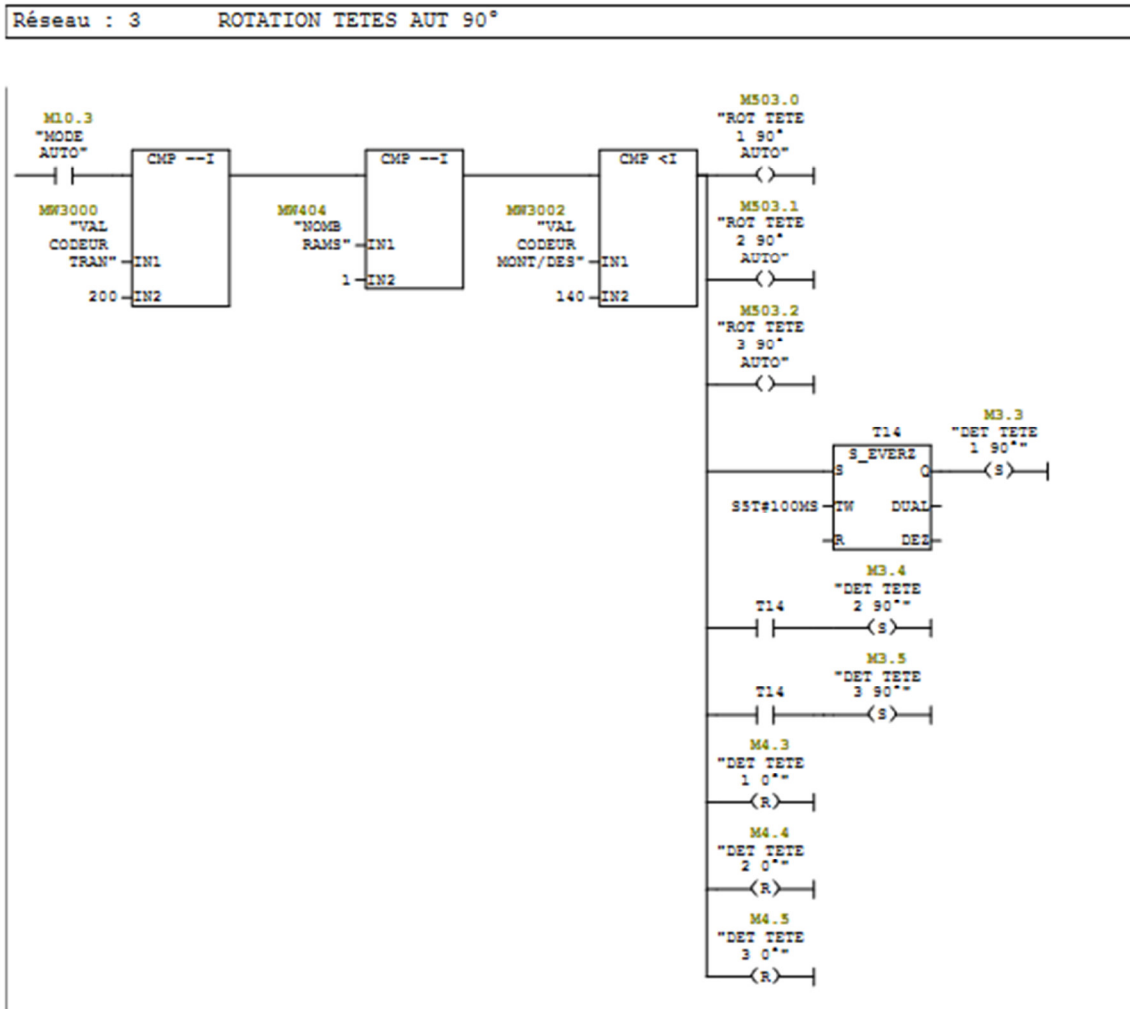
4-7 rotation tetes manuel 0 degré

Réseau : 2 ROTATION TETES MAN 0°



Chapitre 3 : la pince dépileur

4-8 rotation tetes auto 90 degré



5-conclusion

Nous avons présenté dans ce chapitre les différents détails sur la structure de pince dépileur et son fonctionnement avec l'API.

Nous avons commencé par donner la description général de la pince et démontré la structure mécanique par des nombreuse figures ,et on a passé ensuite à le cahier charge pour explique leur cycle automatisé et on a met le grafcet , pour finir à la fin par la programmation en forme de langage contact dans logiciel step7 qui on a fait pour l'automate programmable siemens s7-300 que commande le pince.

Conclusion général

Conclusion général

5-Conclusion général

L'objectif de ce travail est l'étude le processus d'un système automatisé, ainsi que la programmation d'un appareil que contiennent plusieurs machines (moteurs) fonctionnent en coordination les uns avec les autres par un automate programmable industriel simens S7-300.

Ce travail ne a aidée d'avoir une idée général sur le domaine de l'automatisation des systèmes industriel et de comprendre les étapes suivies pure ca.

Notre séjour dans la briqueterie de babahoum nous a donné l'occasion d'étudier les différentes étapes de fabrication de briques et de faire contacte avec le domaine industriel, ou on a tiré les conclusions suivants :

Afin de faire une bien automatisation pour le contrôle d'un procédé il faut :

- un matériel de contrôle approprié
- un langage pour communiquer avec le matériel de contrôle
- une connaissance du procédé à contrôler
- des méthodes de calcule de commande et une bonne connaissance du matériel et des logiciels et des algorithmes reliés à la commande.

On a fait et réalisé le programme en l'engage contact et le simuler par simulation simens.

En perspective de ce travail on propose une simulation de notre programme par logiciel Win-cc flexible.

REFERENCES BIBLIOGRAPHIE

[1] « fabrication de la brique » www.brique.be.

[2] « Automate programmable Siemens - Logiciel Siemens »
<file:///C:/Users/client/Desktop/Automate%20programmable%20Siemens%20-%20automatisme%20industriel,%20automates%20programmables%20neufs%20et%20occasions.html>

[3] « Manuel S7-300 Caractéristiques de la CPU du système S7-300 : CPU 315T-2 DP ».

[4] Contrôleurs SIMATIC « brochure_simatic-controller_overview_fr »

[5] « Utilisation de Step 7 » <http://ptrau.free.fr/autom/siemens/step7.htm>.

[6] : SIEMENS. [en ligne], www.siemens.com

[7] : Automate programmable Siemens – Logiciel Siemens. [en ligne],

<http://www.lcautomatisme.fr/15.html>

[8] : Manuel Système d'automatisation S7-300 Caractéristiques des modules 2013.

[9] :google images.