

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SIENTIFIQUE

UNIVERSITE MOULoud MAMMERI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'ELECTROTECHNIQUE



Projet de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état en électrotechnique
Option : Machines électriques

Thème

Réalisation d'un registre de coupe à l'aide
d'un Automate Programmable Industriel

Réalisé par :

- ❖ Mr : AIGOUN Omar
- ❖ Mr : HADDAD Makhoulf
- ❖ Mr : AKLI Makhoulf

Proposé par :

- ❖ Mr : HADDAD .S
- Dirigé par :
- ❖ Mr : R.MANSOUF
 - ❖ Mr : M.MAKEFEL

Promotion : 2010

Sommaire

| | |
|--|----|
| INTRODUCTION GÉNÉRALE..... | 1 |
| CHAPITRE I : Introduction aux Automates Programmables Industriels | |
| I.1 Introduction..... | 3 |
| I.2 Généralité sur les systèmes automatisés..... | 3 |
| I.2.1 Objectifs de l'automatisation..... | 3 |
| I.2.2 Les fonctions d'automatisation..... | 4 |
| I.2.3 Structure générale d'un système automatisé..... | 4 |
| I.2.4 Technologie des automates..... | 6 |
| I.3 Notions de la logique programmable..... | 7 |
| I.3.1 type de la logique..... | 7 |
| I.3.1.1 Logique combinatoire..... | 7 |
| I.3.1.2 Logique séquentiel..... | 7 |
| I.4. Pourquoi les API ?..... | 10 |
| I.4.1 Domaines d'utilisation des API..... | 10 |
| I.4.2 Les fonctions principales autour de l'automate programmable..... | 10 |
| Conclusion..... | 11 |
| CHAPITRE II : Présentation Matérielle l'API SIMATIC S7-200 | |
| II.1 Introduction..... | 12 |
| II.2 La famille SIMATIC S7..... | 12 |
| II.2.1 L'Automate programmable (API)..... | 12 |
| II.2.2 La console de programmation (CP)..... | 12 |
| II.2.3 Terminaux de dialogue(OP)..... | 13 |
| II.3 Étude de l'API S7 200..... | 13 |
| II.3.1 Module d'alimentation..... | 13 |
| II.3.2 Module unité centrale (CPU)..... | 14 |
| II.3.3 Les entrées et sorties..... | 15 |
| II.3.3.1 Les entrées et sorties locales..... | 15 |
| II.3.4 Les modules d'entrées/sorties d'extension..... | 16 |
| II.4 Connexion de la CPU S7-200..... | 17 |
| II.4.1 Raccordement de l'alimentation à la CPU S7-200..... | 17 |
| II.4.2 Connexion du câble USB/PPI multi-maitre..... | 17 |
| II.5 Caractéristiques du S7-200..... | 17 |
| II.5.1 Interruption du cycle du programme..... | 17 |
| II.5.2 Comptage rapide des E/S..... | 18 |
| II.6 Bus d'informations..... | 18 |
| II.7 Le cycle de fonctionnement d'un S7-200..... | 19 |

| | |
|---|----|
| II.8 L'adressage..... | 20 |
| II.8.1 Adressage absolu..... | 20 |
| II.8.2 Adressage symbolique..... | 21 |
| II.8.3 Adressage des E/S locales et des E/S d'extension..... | 22 |
| II.9 Les mémoires..... | 23 |
| II.9.1 les différentes zones de mémoire d'une CPU S7-200..... | 24 |
| II.10 Avantages des automates programmables..... | 25 |
| II.11 Inconvénients des automates programmables..... | 25 |
| II.12 La sûreté de fonctionnement..... | 26 |
| II.12.1 Les différents aspects du concept de sûreté..... | 26 |
| II.12.2 Les dispositifs généraux de sécurité..... | 26 |
| II.12.3 La sûreté du procédé..... | 27 |
| II.13 Classification des pannes..... | 27 |
| Conclusion..... | 28 |

CHAPITRE III : Description technique de la machine d'impression KOMORI

| | |
|---|----|
| III.1 Introduction..... | 29 |
| III.2 Description technique d'une machine rotative..... | 29 |
| III.2.1 Principe de fonctionnement..... | 29 |
| III.2.1.1 Cycle de découpage..... | 30 |
| III.2.1.2 Cycle de pliage..... | 30 |
| III.3 Problématique..... | 30 |
| III.4 Solution proposée..... | 30 |
| III.5 Présentation des éléments du nouveau dispositif..... | 31 |
| III.5.1 Automate programmable industriel S7 200..... | 31 |
| III.5.2 L'afficheur TD 200..... | 31 |
| III.5.2.1 Les fonctions du TD 200..... | 32 |
| III.5.2.2 Composants du TD 200..... | 32 |
| III.5.2.3 Les caractéristiques du clavier TD 200..... | 33 |
| III.5.2.4 Les caractéristiques du TD 200 version 3.0..... | 33 |
| III.5.2.5 Fonctionnement des touches de l'afficheur TD 200..... | 34 |
| III.5.3 Encodeur rotatif industriel..... | 34 |
| III.5.3.1 Les différents types d'encodeurs..... | 35 |
| III.5.4 Photocellule..... | 38 |
| III.5.5 Capteur inductif..... | 38 |
| III.5.6 Moteur de régulation..... | 39 |
| III.6 Principe de fonctionnement du dispositif..... | 40 |
| Conclusion..... | 42 |

| | |
|--|----|
| CHAPITRE IV : Programme de Commande de la Machine | |
| IV.1 Introduction..... | 43 |
| IV.2 Les éditeurs de programmation de STEP7-micro/WIN32..... | 43 |
| IV.2.1 Éditeur liste d'instruction(LIST)..... | 43 |
| IV.2.2 Éditeur logigramme (LOG/FBD)..... | 43 |
| IV.2.3 Éditeur schéma à contacts (CNT/LD)..... | 44 |
| IV.3 Organisation du programme utilisateur..... | 45 |
| IV.4 Édition d'un programme sous STEP7-micro/WIN32..... | 46 |
| IV.4.1 Création d'un projet..... | 46 |
| IV.4.2 Démarrage de STEP7-micro/WIN32..... | 47 |
| IV.4.3 Édition du programme en langage CONT..... | 49 |
| IV.4.4 Établissement de la communication avec le S7-200..... | 50 |
| IV.4.5 Chargement du programme dans la CPU..... | 50 |
| IV.4.6 Mise à l'état marche du S7-200..... | 50 |
| IV.4.7 Surveillance du programme à l'aide de la table de visualisation d'état..... | 50 |
| IV.5 Opérandes utilisés dans notre programme..... | 51 |
| IV.6 Structure du programme..... | 52 |
| | |
| CONCLUSION GENERALE..... | 77 |
| | |
| BIBLIOGRAPHIE | |
| ANNEXE | |

Remerciement

Au terme de notre travail, nous remercions en premier lieu le bon dieu de nous avoir donné la volonté, et le courage de mener dans de bonnes conditions ce présent travail.

Nous tenons a adresse nos vifs remerciements et gratitudes à M^r R. Mansouri qui a accepté de nous encadrer et à M^{rs} H.SAID et M.MAKEFAL nos deux co-promoteurs qui ont proposé le sujet dont les conseils les critiques, les encouragements, et pour leur continuelle disponibilité.

Nos vifs remerciements à tous les fonctionnaires de général maintenance pour leur très bon accueil, et un excellent climat de travail.

Nos remerciement a tout les travailleur de AURES emballage qui ont su garder toujours pour nous les portes du site ouvertes.

Nos remerciement s'adressent également, aux membres de jury, qui nous ferons l'honneur de juger notre modeste travail.

Et enfin nous avons l'agréable plaisir de remercier tout ceux qui contribué de près ou de loin a la réalisation de ce travail.

Dédicaces

Je dédier ce modeste travail à

Mes chers parents

Mes frères et sœurs et leurs enfants

Ma grand- mère

Mes cousins et oncles et toute ma famille

A tous mes amis (es)

Et à tous ceux que j'aime.

AKLI Makhlouf

Dédicaces

*Je dédie ce modeste travail en élevant
mon profond attachement a ceux dont je garde le
souvenir eternal des sacrifices et de la tendresse :
Mes très chers parents, que dieu les protèges.*

A

Mon frère et sa petite amie.

Ma sœur et son fiancé.

Ma grande mère.

*Toute ma proche famille, et en particulier saïd
nagh.*

A

Tous mes amis (es).

*Tous ceux qui m'ont hébergé le long de mon
cursus.*

*Tous les fonctionnaires de général
maintenance.*

*Tous ceux qui ont contribués à la réalisation de ce
travail.*

M.HADDAD

Dédicaces

Je tiens à dédier ce modeste travail en premier lieu à mes très chers parents pour leurs sacrifices et dévouements.

A mes frères et sœurs.

A ma sœur et son mari.

A tous l'équipage de générale maintenance (GM).

A tous mes amis du village Ait Irane.

A tous ceux que j'aime.

A tous ceux qui ont contribué de près et de loin à la réalisation de ce travail.

Omar.



***Introduction
générale***

Introduction générale :

L'histoire de l'imprimerie est étroitement liée à celle de l'homme, depuis que l'homme a développé des moyens de communiquer (représentation artistique sur la vie, théâtre,.....). Des égyptiens, qui gravaient la pierre passaient leurs journées à reproduire des œuvres en les recopiant à la main. L'être humain a toujours cherché à automatiser ces moyens de copie et à diffuser ces œuvres.

L'imprimerie a été révolutionnée dans les années 1880 par l'invention de la linotype (Otto Mergenthaler, 1884). Cette machine accélérât la composition en substituant à l'enregistrement manuel des caractères mobiles une saisie au clavier de chaque ligne de texte.

Dans les années 1940, on imagina de substituer aux lignes-blocs une plaque qui pourrait indifféremment comporter du texte ou une image. Cette plaque imprimante fixait l'encre aux endroits voulus par charge électrostatique (plaque dite « électro graphique ») ou par insolation (« cliché »). Cette technique donna naissance aux premières photocopieuses et ouvrait la voie à la conception des plaques offset.

Depuis plusieurs décennies, on assiste à une évolution technologique des matériels, notamment au niveau des équipements électromécaniques, des systèmes de télétransmission, de l'informatique et des capteurs, qui permet l'augmenter la fiabilité et la sécurité de la production.

L'événement de l'automatisation s'est généralisé à l'ensemble des activités de production, tant dans l'industrie, que dans les activités de service. Quelque soit son domaine d'application et les techniques aux quelles elle fait appel, l'automatisation s'est constamment développée dans l'unique but de réduire la pénibilité du travail humain et d'améliorer la productivité.

La compétitivité des entreprises exige une automatisation plus flexible et évolutive de la part des équipements de production. La technologie des automates programmables permet ce type d'automatisation. Ces systèmes de commande, à l'origine réservés à des applications exigeant des volumes de traitement importants sont maintenant largement utilisés dans les automatismes plus petits.

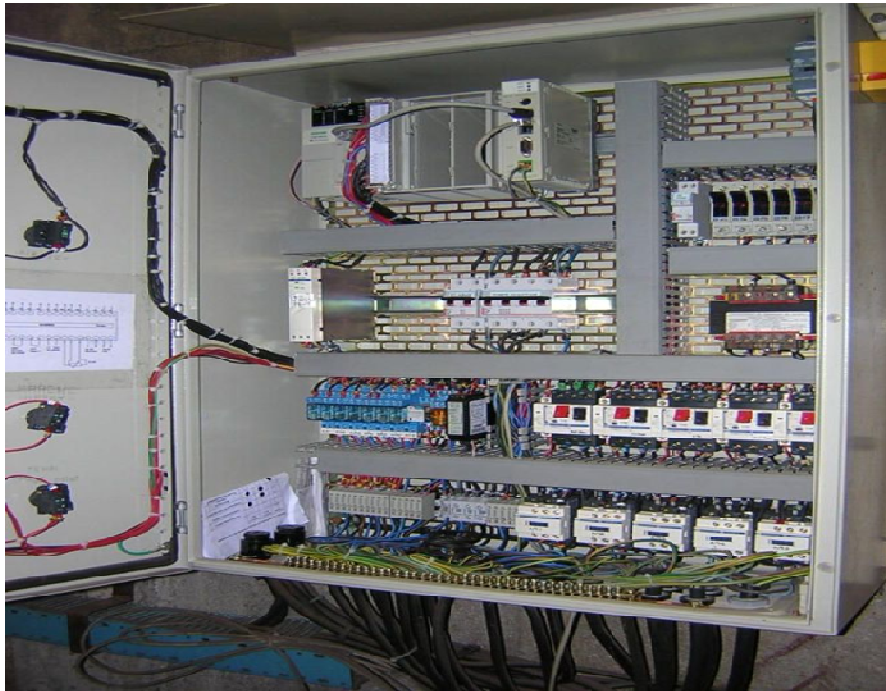
Les automates programmables constituent le volet le plus industriel et le plus simple à mettre en œuvre pour résoudre des problèmes de commande. L'automatisation des imprimeries devient impérative, en considérant les nouvelles conceptions basées sur la rigueur dans la gestion de la production.

Ce projet rentre dans cette perspective de rénovation de l'automatisme de la machine d'impression KOMORI et plus exactement du bloc de découpage BLAVA. Il nous a été demandé de concevoir une commande programmable à l'aide du langage de programmation dédié à la famille SIMATIC de SIEMENS et qui, pour notre cas, est le STEP7micro-WIN32 pour l'amélioration de la précision des coupes.

Le premier chapitre est porté sur la présentation de l'automate programmable en liaison avec un système automatisé. Le deuxième chapitre est consacré à la présentation de l'aspect de l'API utilisé (SIMATIC S7-200). Le troisième chapitre est consacré à la description de la machine d'impression KOMORI et ses différents composants. Le quatrième chapitre est porté sur la programmation du S7-200 appliqué à la BLAVA, et enfin on termine par une conclusion générale.

Chapitre I

*Introduction aux Automates
Programmables Industriels*



I.1 Introduction :

L'automate programmable industriel (API) est un dispositif électronique programmable adapté à l'environnement industriel. Sa fonction essentielle est le pilotage des systèmes de productions. Depuis leur apparition ils ont permis à l'homme de supprimer les travaux pénibles, dangereux, répétitifs, voir impossibles à réaliser par lui. Ils ont également permis de contourner les problèmes de matérialisation rencontrés au niveau des automatismes complexes. Les API prennent en charge les systèmes de commande combinatoires, séquentielles et à régulations numériques.

Aujourd'hui les API s'imposent pratiquement dans tous les domaines (domaine industriel, médical, agricole...). Ils répondent aux besoins d'adaptation et de flexibilité de nombreuses activités économiques. [7]

I.2 Généralités sur les systèmes automatisés :

Un système automatisé est un ensemble d'éléments en interaction, organisés dans le but de produire des produits de qualité, et ce pour un faible coût avec le minimum d'intervention humaine.

L'automatisation d'un procédé, consiste à en assurer la conduite par un dispositif technologique qui peut prendre en compte les situations pour lesquelles sa commande a été conçue. L'intervention d'un opérateur est souvent nécessaire pour assurer un pilotage global du procédé (spécification des consignes de fonctionnement) pour surveiller les installations et répondre en commande manuelle (dans certaines situations critiques) la commande du système. [7]

I.2.1 Objectifs de l'automatisation :

Pour qu'une entreprise puisse répondre aux exigences de la compétition économique, il est important de suivre le chemin de l'automatisation de ses systèmes de production, cette automatisation permet d'atteindre les objectifs suivants :

- Accroître la rentabilité, et la compétitivité :
 - produit à qualité constante.
 - produire les quantités nécessaires au bon moment.
- Amélioration de la flexibilité de production.
- Amélioration des conditions de travail :
 - Augmentation de la sécurité.
 - Adaptation à des tâches physiques ou industrielles pénibles pour l'homme. [2]

I.2.2 Les fonctions d'automatisation :

L'ensemble des fonctions (acquisition, traitement, action) forme un système. Son degré d'automatisation est extrêmement variable selon la nature du procédé et sa complexité, les connaissances qu'on a, et les objectifs assignés au projet. On distingue trois degrés de fonction d'automatisation : la surveillance, le guide opérateur et la commande.

- la surveillance des grandeurs répond à un objectif de connaissance technique et économique du procédé. Il s'agit d'une fonction passive vis-à-vis du procédé du moins dans le très court terme, l'organe de contrôle acquiert des informations, les analyses et produit des journaux de bord et des bilans.
- Le mode opératoire complète le précédent par des traitements plus élaborés, et propose aux responsables du site des actions pour conduire le procédé selon un critère donné.
- la commande correspond à l'automatisation complète de certaines fonctions, depuis l'acquisition des informations en passant par leur traitement pour aboutir à une action sur le procédé. [2]

I.2.3 Structure générale d'un système automatisé :

Tout système automatisé peut être décomposé en trois parties essentielles :

❖ **Partie opérative (PO):** appelée aussi processus ou partie puissance, elle réalise les opérations sur la matière d'œuvre afin d'élaborer la valeur ajoutée. Elle englobe :

- Les actionneurs qui réalisent les actions et les interventions physiques que le système de commande impose au processus industriel. Dans cette partie, on trouve des moteurs, des vannes, des vérins, des électro-aimants,....
- Des capteurs d'information qui mesurent instantanément toutes les grandeurs nécessaires à la surveillance, et à la commande du bon fonctionnement du système automatisé. Ils transforment des déplacements, des débits, des pressions, des températures et des niveaux en signaux électriques, qui sont en liaison permanente avec la partie commande, permettant ainsi la mise à jour des entrées de l'automate, et avec la partie contrôle, permettant la visualisation en temps réel l'évolution du processus. [5]

❖ **Partie commande (PC) :** que l'on appelle aussi automate, c'est à partir des informations en entrées acquises par les capteurs et les consignes du pupitre de l'opérateur, que l'automate élabore des ordres en sortie afin d'assurer le pilotage de la partie opérative. La partie commande englobe essentiellement :

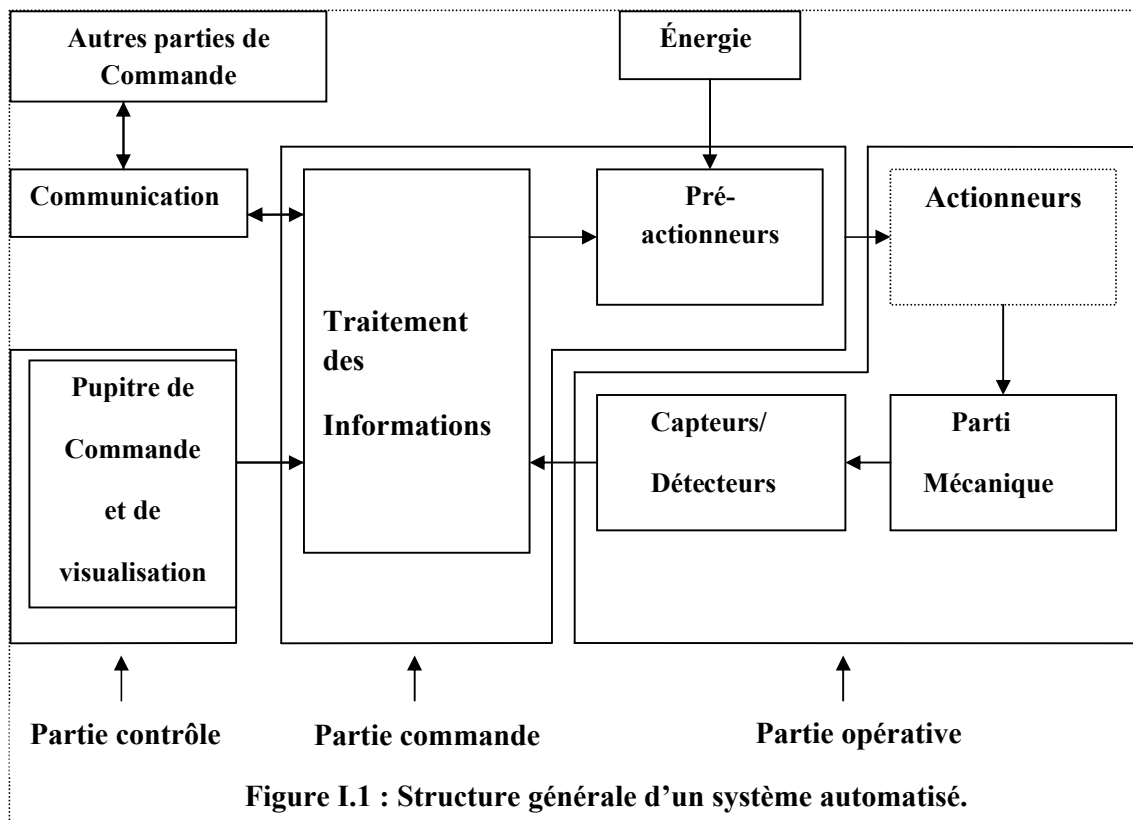
- Un circuit de traitement d'informations, qui est l'organe intelligent de l'automatisme. Suivant les informations récoltées par les capteurs, il élabore des signaux en sortie servant au pilotage correcte du processus. Ce circuit peut être réalisé en logique

programmée (l'automate programmable industriel) ou en logique câblée (circuit à base des bascules).

- Des pré-actionneurs qui permettent de commander les actionneurs en assurant le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique,...) et les actionneurs. Ces pré-actionneurs sont commandés à leur tour par le circuit de traitement d'information.

En fonction de ces consignes et de son programme de gestion des tâches implanté dans un automate programmable, ou réalisé par des relais (on parle de la logique câblée) elle va commander les pré-actionneurs, renvoyer des informations au pupitre de signalisation ou d'autre système de commande et de supervision en utilisant un réseau et un protocole de communication. [5]

- ❖ **Partie contrôle (PC)** : composée de pupitre de commande et de signalisation, elle permet à l'opérateur de commander le système (marche, arrêt, départ cycle,...), et visualiser également en temps réel les différents états de son évolution. [5]



I.2.4 Technologies des automates :

Pour la réalisation pratique de l'automate, l'automaticien dispose de plusieurs solutions que l'on regroupe habituellement en deux catégories fondamentales : solution câblée et solution programmée. [7]

❖ Solution câblée :

C'est une méthode de réalisation de la loi de contrôle en interconnectant judicieusement des opérateurs matériels (électrique ; pneumatique ; hydraulique,....) réalisant des fonctions logiques de base (AND.OR.NOT.NAND.NOR).

Avantages:

L'intérêt principal de cette solution réside dans :

- La facilité de maintenance de l'automate.
- La possibilité de reproduction fidèle du GRAFCET.
- Leur implantation facile pour des installations de taille modeste.

Inconvénients:

- Le volume de l'automate est proportionnel à la complexité du problème, Ce qui rend la tâche délicate dans la matérialisation des systèmes complexes.
- des modifications de la commande impliquent des modifications du câblage.

❖ Solution programmée :

L'idée de la logique programmée est de n'utiliser qu'un seul jeu d'opérateurs de base (unité logique) permettant de réaliser n'importe quelle fonction logique donnée par combinaison. Le pilier de cette logique est la mémoire, dans laquelle on écrit la suite des opérations est des actions à réaliser.

Avantage :

- banalisation du matériel : même matériel quelque soit la fonction logique à réaliser.
- facilite la modification de la loi de contrôle : il suffit de modifier le programme utilisateur.
- faible liaison entre le volume matériel et la complexité du problème.

Inconvénients :

- la vitesse d'exécution du programme est inversement proportionnelle à la complexité du problème.

I.3 Notions sur la logique programmable:

I.3.1 Type de logique:

I.3.1.1 Logique combinatoire:

Un circuit de logique combinatoire est constitué d'éléments logiques élémentaires appelés portes logiques. Ces portes logiques reçoivent des signaux appliqués en entrée, et produisent des signaux en sortie après application de fonctions logiques pour lesquelles elles sont destinées. Dans la logique combinatoire l'état des sorties ne dépend que de l'état des entrées. Elle est constituée de circuit sans mémoire c'est-à-dire en l'absence de variable d'entrée, les variables de sortie n'existent pas. Ainsi une information binaire transmise en entrée est transformée en une autre information disponible en sortie. La figure I.2 schématise un tel circuit combinatoire. [3]

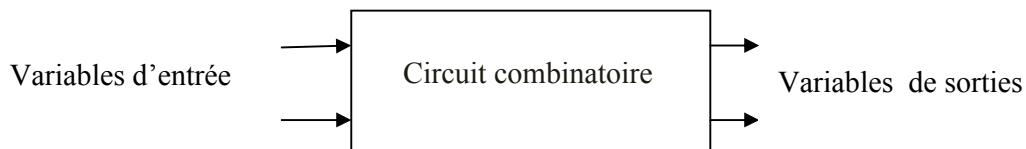


Figure I.2 : Schéma d'un circuit combinatoire.

I .3.1.2 Logique séquentielle:

La mémoire de l'ordinateur stocke les données et les instructions du programme en cours d'exécution. Pour réaliser cette mémoire, des circuits dits de mémorisation sont utilisés pour mémoriser un bit (0 ou 1). Il faut utiliser un circuit capable de se souvenir de la valeur qu'il a enregistrée. Ces circuits sont appelés circuits séquentiels. A l'inverse des circuits combinatoires, la valeur de la sortie d'un circuit séquentiel ne dépend pas seulement des valeurs logiques d'entrée mais aussi de la valeur de sortie antérieure. [3]

❖ Définition d'un circuit séquentiel:

Un circuit séquentiel est composé d'un circuit combinatoire et d'éléments mémoires appelés bascule, comme le montre la figure I.3.

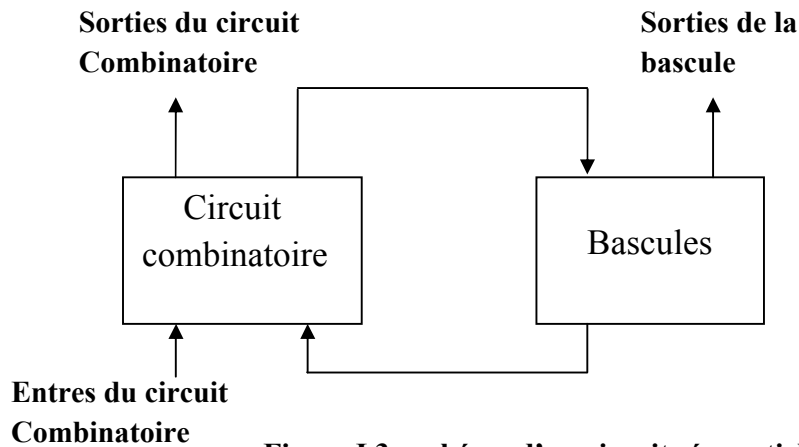


Figure I.3 : schéma d'un circuit séquentiel.

On distingue deux types de circuits séquentiels :

- **Circuit séquentiel synchrone :**

Dans un circuit séquentiel synchrone, l'état de la variable de sortie dépend des signaux appliqués en entrée synchronisé à l'aide d'un signal périodique. Ces circuits utilisent des horloges d'où leur appellation « circuits séquentiels à horloge ».

- **Circuit Séquentiel asynchrone :**

Contrairement aux circuits séquentiels synchrones, la variable de sortie ne dépend pas du signal d'horloge mais de l'ordre d'application des signaux en entrée.

- ❖ **Définition d'une bascule:**

Une bascule est un circuit capable de mémoriser un (0 ou 1), ce qui est équivalent à dire qu'un tel circuit peut se trouver dans l'un de ces deux états logiques. Elle possède deux sorties qui sont complémentaires l'une de l'autre (Q et \overline{Q}). Quand une bascule contient un 1 logique en sortie on dit qu'elle est mise à un ($Q=1$ et $\overline{Q}=0$), et quand elle contient un 0, on dit qu'elle est mise à zéro ($Q=0$ et $\overline{Q}=1$). Nous allons introduire dans cette section les quatre bascules les plus souvent utilisées.

- **La bascule RS (rest_set) :**

Connue aussi sous le nom de multivibrateur bistable, à deux entrées S et R : pour set (mise à un) et pour rest (remise à zéro). Les bascules RS sont facilement réalisées à base de portes logiques NOR (figure I.4), ou des portes NAND (figure I.5).

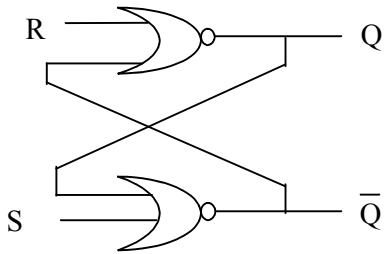


Figure I.4 : Réalisation d'une bascule RS à l'aide des portes NOR.

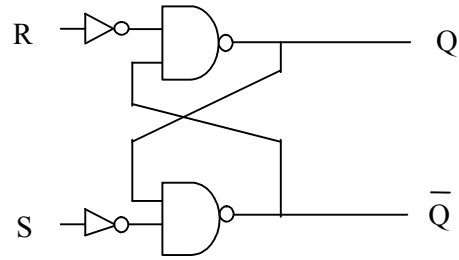


Figure I.5 : Réalisation d'une bascule RS à l'aide des portes NAND.

- **La bascule JK :**

Elle fonctionne comme la bascule RS avec un signal d'horloge en plus, elle est symbolisée comme suite :

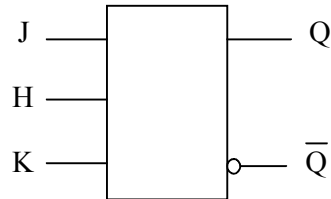


Figure I.6 : Bascule JK.

- **Bascule T :**

La bascule T change d'état chaque fois qu'une impulsion est reçue sur sa seule entrée T, et elle est symbolisée comme suite :

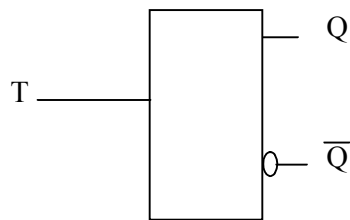


Figure I.7 Bascule T.

- **Bascule D :**

C'est la bascule la plus utilisée, car la sortie de cette bascule recopie sa seule entrée D au rythme des impulsions d'une horloge. Elle est symbolisée sur la figure I.8.

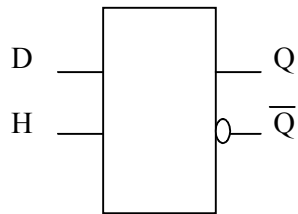


Figure I.8: Bascule D.

I.4 Pourquoi les API ?

Les automates programmables industriels sont apparus aux Etats-Unis vers 1969. ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

L'API s'est ainsi substitué aux armoires à relais en raison de sa souplesse (mise en œuvre, évolution), mais aussi parce que dans les automatismes de commande complexe le coût de câblage et de mise au point devenaient trop élevés. [2]

I.4.1 Domaines d'utilisation des API :

Les API s'adressent à des applications que l'on trouve dans la plupart des secteurs industriels comme en :

- Métallurgie et sidérurgie.
- Mécanique et automobile.
- Industries chimiques.
- Industrie pétrolière.
- Industrie agricole et alimentaire.
- Transport et manutention. [2]
-

I.4.2 Les fonctions principales autour de l'automate programmable :

Les fonctions susceptibles d'être amenées par un automate programmable sont :

❖ Détection :

Les capteurs « tout ou rien » tels que détecteurs de proximité, interrupteurs de position, manostats..., sont raccordés à l'automate programmable par des liaisons fil à fil aboutissant aux modules d'entrées.

Les capteurs analogiques ou numériques tels que lecteurs de codes, plots de lecture écriture..., sont raccordés à l'automate programmable par liaison « série ».

❖ **Action :**

A chaque actionneur sur la machine (moteur, vérin,...) est associé un pré-actionneur (contacteur, variateur de vitesse, distributeur,...).

Les actionneurs «Tout ou Rien» sont raccordés à l'automate programmable par des liaisons fil à fil depuis les modules de sortie.

Les pré-actionneurs continus tels que les variateurs de vitesse et positionneurs sont raccordés à l'automate programmable par des transmissions depuis les cartes de sorties analogiques.

❖ **Dialogue d'exploitation :**

En période d'exploitation, des dialogues homme machine sont nécessaires, pour la conduite de la machine, pour ses réglages ainsi que pour ses dépannages.

Les constituants prévus pour ces dialogues comportent, des constituants implantés tels que boutons, voyants, pupitres configurables..., et des terminaux d'exploitation amovibles, pour réglages et dépannages. [2]

I.5 Conclusion :

L'automate programmable industriel est un bon produit, facile à programmer, à connecter, à adapter aux conditions industrielles et à analyser les différents problèmes des processus d'applications par respect des règles d'installation, ce qui a permis à l'homme de l'adapter non seulement dans l'industrie, mais aussi dans différents secteurs.

L'automate programmable industriel ne se limite pas à l'échange d'informations avec son propre processeur, mais assure plusieurs relations avec d'autres processeurs dans le cadre d'un réseau, qui permet un dialogue permanent entre les différents acteurs d'une chaîne de production.

Chapitre II

Présentation matérielle de l'API SIMATIC S7-200



II.1 Introduction :

SIMATIC est un atelier de logiciel et de matériel, qui englobe une grande diversité de produits et de systèmes d'automatisations standardisés dans la firme SIEMENS. Offrant toutes les libertés de personnalisation, d'extensions spécifiques et permettant de mettre en place des solutions pour toute tâche d'automatisation. [13]

II.2 La famille SIMATIC S7 :

SIEMENS fabrique et développe des automates programmables industriels depuis plus de 30 ans, cette expérience a été capitalisée dans la conception de la famille SIMATIC S7 qui est une famille cohérente de matériels pour automatisations à logique programmée (automate programmable, console de programmation, terminaux de dialogue...). La compatibilité des appareils garantie au-delà des changements de générations, apporte une sécurité d'investissement sur des dizaines d'années. [13]

II.2.1 L'automate programmable (API) :

La gamme des automates programmables SIMATIC S7 va du micro-automate compact à l'automate le plus performant. Ils intègrent essentiellement un micro-processeur, une alimentation et des entrées/sorties, permettant de commander une large gamme d'appareils afin de répondre à nos besoins en matière d'automatisation.

Les différentes formes d'API (compact, modulaire), leur configuration souple et leur important jeu d'opérations en font une solution idéale pour la commande d'applications très variées. Le choix de l'automate est appuyé sur plusieurs critères (la capacité de traitement, vitesse de traitement, nombre d'entrées/sorties...). On peut dire qu'il est proportionnel à la complexité des systèmes à automatiser. [13]

II.2.2 Console de programmation (CP) :

La console de programmation (CP) d'un ensemble d'API est l'outil privilégié de communication homme-machine pour le développement à la mise au point et éventuellement l'exploitation des applications. Elle se présente comme un poste de travail composé d'un clavier, un écran, des dispositifs spécifiques associés, complétés éventuellement par des périphériques, et auxiliaires.

La console de programmation contient le logiciel de programmation, pour la CPUS7-200 on utilise step7-Micro/WIN32 et elle permet :

- d'écrire le programme utilisateur, de le compléter et de le transférer dans la CPU (cartouche mémoire).
- d'exécuter le programme pas à pas et de le visualiser.
- de forcer ou de modifier des données telles que les entrées et les sorties, les bits internes, les temporisations et les compteurs. [9]

II.2.3 Terminaux de dialogue pupitre opérateur (OP) :

L'offre d'unités de dialogue à eu pour but la visualisation en temps réel du déroulement du cycle du système automatisé, et de permettre aussi d'imposer certaines entrées sur l'API en cas de besoin (par exemple : changer les consignes de production, imposer un arrêt d'urgence du cycle...). Elles servent aussi à l'affichage des valeurs courantes des temporisations et des compteurs internes, de toute entrées/sorties, l'affichage manuel du numéro de programme à contrôler et la visualisation de l'instruction d'adressage. [7]

II.3 Etude de l'API S7-200

L'API S7-200 est un micro-automate programmable utilisable dans des applications d'automatisation variées. Sa forme compacte, son faible prix et son important jeu d'opérations en font une solution idéale pour la commande de petites applications.

La CPU S7-200 combine un microprocesseur, une alimentation intégrée, des circuits d'entrées, des circuits de sorties fixes dans un boîtier compact. Il existe cependant des CPU comme : CPU222, CPU224, CPU224XP, CPU226, qui comportent une large variété de module d'extension dont on peut se servir pour ajouter des fonctions à la CPU. On trouve des modules d'entrées/sorties Tout Ou Rien (TOR), analogique et des modules intelligents (en compte 7 modules au maximum pour une CPU S7-200 et qui peut y'avoir jusqu'à 250 entrées- sorties). Figure II.1. [9]

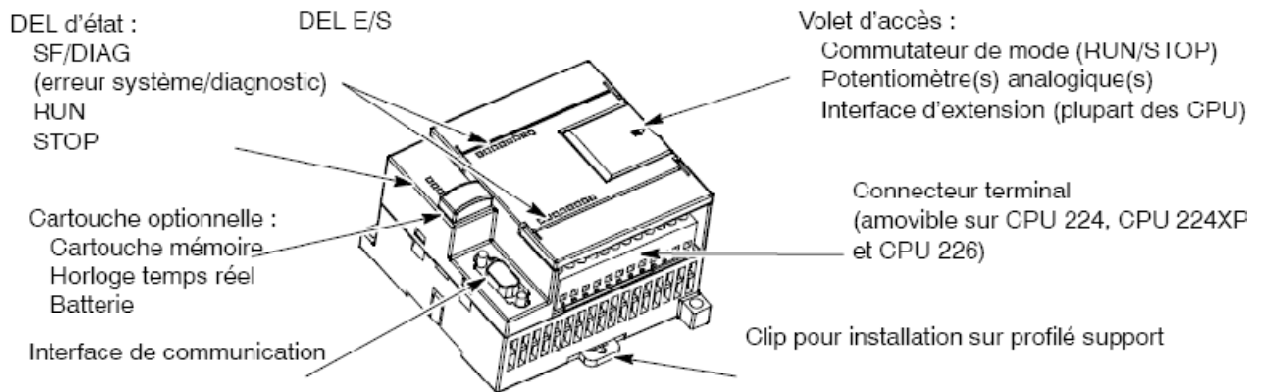


Figure II.1 : La CPU d'un S7-200.

II.3.1 Module d'alimentation:

La CPU S7-200 possède une alimentation intégrée, qui permet de fournir à l'automate l'énergie nécessaire pour son fonctionnement à partir d'une alimentation de 85 à 265V en alternatif ou d'une source à courant continu de 24V. Cette alimentation intégrée dérive les tensions 24V, 12V, 5V en continu pour alimenter la CPU, les modules d'extension et tous les équipements consommant les tensions continues précédentes. En règle générale un voyant positionné sur la façade indique la mise sous tension de l'automate. [5]

II.3.2 Module unité centrale (CPU) :

Chaque automate programmable doit comporter toujours un module unité centrale (CPU) qui constitue le cerveau de l'automate programmable. Elle prend en charge la responsabilité de traiter toutes les tâches propres au processeur telles que les fonctions logiques arithmétiques et le traitement numérique (transfert, comptage, temporisation,...).

Les CPU se distinguent en particulier par leurs microprocesseurs, leurs capacités mémoires et leurs temps d'exécution.

La CPU 224 qu'on a utilisée dans notre projet comporte :

- Une unité de commande (microprocesseur) avec une vitesse d'exécution de 0.22 microseconde /opération.
- Une mémoire de programme interne (RAM).
- Un logement pour une pile de sauvegarde du contenu de la mémoire RAM en cas de coupure de tension d'alimentation.
- Un logement pour une cartouche mémoire (EEPROM ou EPROM).
- Une interface de communication, qui permet la connexion de la CPU avec la console de programmation, terminal de dialogue.
- Commutateur de mode.
- Les DEL de signalisation d'état.
- Une alimentation interne (24 VCC, 85 à 265 VAC).
- Des entrées/sorties locale TOR (14 entrées, 10 sorties).
- Des interfaces d'extension.

Il existe en effet deux programmes qui s'exécutent à l'intérieur d'une CPU 224, le système d'exploitation et le programme utilisateur. [9]

➤ Système d'exploitation :

Chaque CPU contient un système d'exploitation. C'est avec ce système que sont livrés les automates à leurs sorties de l'usine. Son rôle est l'organisation de toutes les fonctions et procédures dans la CPU qui ne sont pas liées à une tâche d'automatisation spécifique. Il gère les fonctions suivantes :

- le déroulement du démarrage et du redémarrage de la CPU.
- l'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties.
- l'appel du programme utilisateur.
- la détection et le traitement d'erreurs.
- la gestion des zones de mémoire.
- la communication avec des consoles de programmation et d'autres partenaires de communication. [9]

➤ Programme utilisateur :

Après une étude approfondie de l'automatisme à concevoir (analyse du cahier des charges, étude des spécifications fonctionnelles, étude des spécifications techniques, détermination des équations logiques nécessaires à sa réalisation), on passe ensuite à l'étape de programmation et de réalisation. Avec un logiciel de programmation qui soit compatible avec la CPU employée, on crée le programme utilisateur qui contient toutes les fonctions nécessaires au traitement et à la gestion des problèmes d'automatisation spécifique, puis on le charge dans la CPU.

Le programme utilisateur permet en outre :

- la détermination des conditions pour le démarrage et le redémarrage de la CPU (par exemple : initialiser des signaux, des variables...).
- le traitement des données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, fixer des signaux binaires pour la sortie, écrire des valeurs analogiques).
- réagir aux alarmes.
- traiter les perturbations dans le déroulement normal du programme. [9]

II.3.3 Les entrées et sorties :

Les entrées/sorties d'un S7-200 sont portées soit sur la CPU, ou sur les modules d'extension. Ils constituent les interfaces vers les capteurs et pré-actionneurs d'une installation ou d'une machine.

II.3.3.1 Les entrées et sorties locales :

➤ Les entrées Tout Ou Rien « TOR » :

Elles véhiculent les signaux électriques correspondant aux grandeurs physiques saisies par les capteurs afin d'élaborer des informations binaires. On peut raccorder aux entrées TOR les différents capteurs logiques suivant :

- les boutons poussoirs.
- les détecteurs de fin de course.
- les capteurs de proximité (inductive ou capacitive).
- les capteurs photoélectriques.



Figure II.2 : Les entrées TOR.

Elles assurent l'adaptation, l'isolement, le filtrage et la mise en forme des signaux électriques. Le nombre d'entrées varie selon la CPU utilisée (6, 8, 14, 24,...) pour une CPU 224 il y'a 14 entrées TOR, les tensions d'entrées sont de 5VCC ou de 24, 48, 110, 240VAC. La figure II.2 illustre le raccordement des entrées TOR d'une CPU 224. [5]

➤ **Les entrées analogiques :**

Les entrées analogiques permettent de garder des grandeurs analogiques en faisant varier un code numérique au sein de l'entrée. Il existe trois types d'entrées analogiques :

- entrée haut niveau : elle accepte en tension 0/10V et en intensité 0/20mA ou 4/20mA.
- entrée pour thermocouple avec un signal d'entrée 0/20mV, 0/50mV, 0/100mV.
- entrée pour sonde avec un signal d'entrée 0/100mV, 0/250mV, 0/400mV. [5]

Remarque :

Dans notre cas on a utilisé la CPU224 qui ne contient pas d'entrées analogiques car toutes nos entrées sont de type TOR.

➤ **Les sorties Tout Ou Rien (TOR) :**

Les sorties TOR permettent de raccorder avec la CPU les différents pré-actionneurs tels que les électrovannes, les contacteurs, les voyants, les afficheurs, les relais de puissance,...

Les tensions de sorties usuelles sont de 5V en continu ou de 24, 48, 110, 220V en courant alternatif. Les courants vont de quelques milliampères à quelques ampères.

La figure II.3 illustre les sorties TOR d'une CPU 224.



Figure II.3 : Les sorties TOR.

Ces sorties possèdent des transistors et l'état de chaque sortie est visualisé par une diode électroluminescente.

➤ **Les sorties analogiques :**

Les sorties analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module. Il existe deux types de sorties analogiques :

- haut niveau avec résolution de 8bits (256 valeurs) en tension 0/10V ou en 0/20mA ou 4/20mA.
- haut niveau avec une résolution de 12bits (4096 valeurs) en tension 0/10V, 0/5V, +ou-5V, +ou-10V ou en intensité 0/20mA ou 4/20mA.

Ces sorties analogiques assurent la conversion numérique/analogique, la valeur de sortie en intensité ou en tension est proportionnelle à la valeur numérique. Les sorties analogiques peuvent posséder un convertisseur par voie, le nombre de voies sur ces cartes est de 2,4,..... [5]

II.3.4 Les modules d'entrées/sorties d'extension :

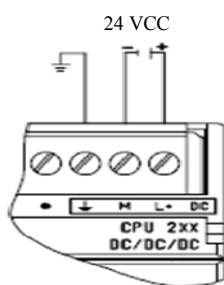
Les modules d'entrées/sortie d'extension constituent l'interface d'entrée/sortie. Ils sont équipés de connecteurs de raccordement permettant un montage rapide à la CPU. Ces entrées/sorties peuvent accepter suivant les modules, des informations en courant ou tension alternatif ou continu. On fait appel à ces modules lorsque les entrées/sorties locales ne suffisent pas pour raccorder tous les capteurs et les pré-actionneurs utilisés. [9]

II.4 Connexion de la CPU S7-200 :

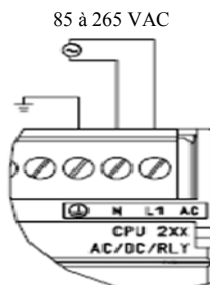
Pour connecter la CPU S7-200, il suffit de raccorder l'alimentation à la CPU, puis de connecter le câble de communication USB/PPI multi-maitre entre la console de programmation et l'interface de communication de la CPU. [9]

II.4.1 Raccordement de l'alimentation à la CPU S7-200 :

Il existe deux modèles de CPU S7-200: un modèle qui s'alimente en courant continu et un modèle qui s'alimente directement en courant alternatif. La figure II.3 montre le câblage de l'alimentation à une CPU S7-200. [9]



Installation en courant continu.



Installation en courant alternatif.

Figure II.4 : Raccordement de l'alimentation à la CPU.

II.4.2 Connexion du câble USB/PPI multi-maitre :

La figure II.5 montre un câble USB/PPI multi-maitre reliant le S7-200 à la console de programmation.

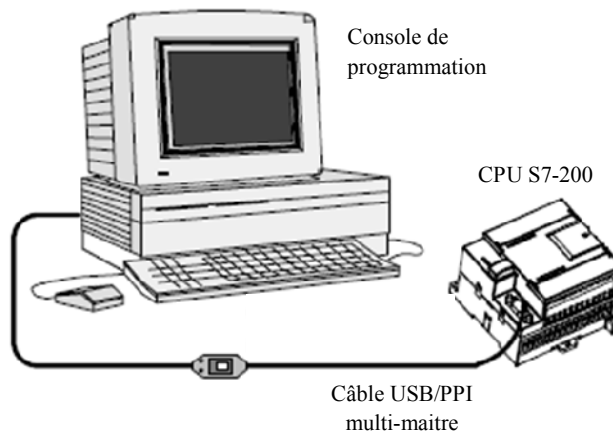


Figure II.5 : connexion du câble USB/PPI multi-maitre.

II.5 Caractéristiques du S7-200 :

Le S7-200 présente plusieurs caractéristiques spéciales qui nous permettent de personnaliser ses fonctions afin qu'elles correspondent mieux à notre application. Parmi ces caractéristiques on distingue :

II.5.1 Interruption du cycle du programme :

Le S7-200 permet au programme d'interrompre le cycle. Si on utilise des interruptions, les programmes associés à chaque événement d'interruption sont sauvegardés comme partie du

programme. Ils ne sont toutefois pas exécutés au cours du cycle normal, mais uniquement lorsque l'événement d'interruption correspondant se produit (ce qui est possible en tout point du cycle). [8]

II.5.2 Comptage rapide des E/S :

Le S7-200 fournit des compteurs rapides intégrés qui comptent des événements externes rapides sans dégrader les performances. Chaque compteur a des entrées réservées aux horloges, à la commande du sens de comptage, à la mise à zéro et au démarrage lorsque ces fonctions sont prise en charge. On peut sélectionner différents modes en quadrature pour varier la fréquence de comptage. [8]

II.6 Les bus d'information :

Les échanges d'informations entre les différents éléments de l'automate (entrées, sorties, mémoires) et les éléments de l'unité centrale sont véhiculées par des nappes de fils ou des circuits imprimés que l'on nomme les bus. La capacité d'un bus est adaptée aux informations qu'y circulent (largeur des mots, des adresses, nombre de signaux de contrôle,...). Ces bus sont représentés par la figure II.6

On distingue trois types de bus :

- Les bus de données véhiculant toutes les informations d'entrées/sorties, d'instruction...
- Les bus d'adresse véhiculant les adresses auxquelles les données doivent être lues ou écrites. Ces adresses sont les étiquettes de rangement.
- Les bus de contrôle véhiculant des niveaux logiques qui autorisent la lecture ou l'écriture des données, en mémoire par exemple. [5]

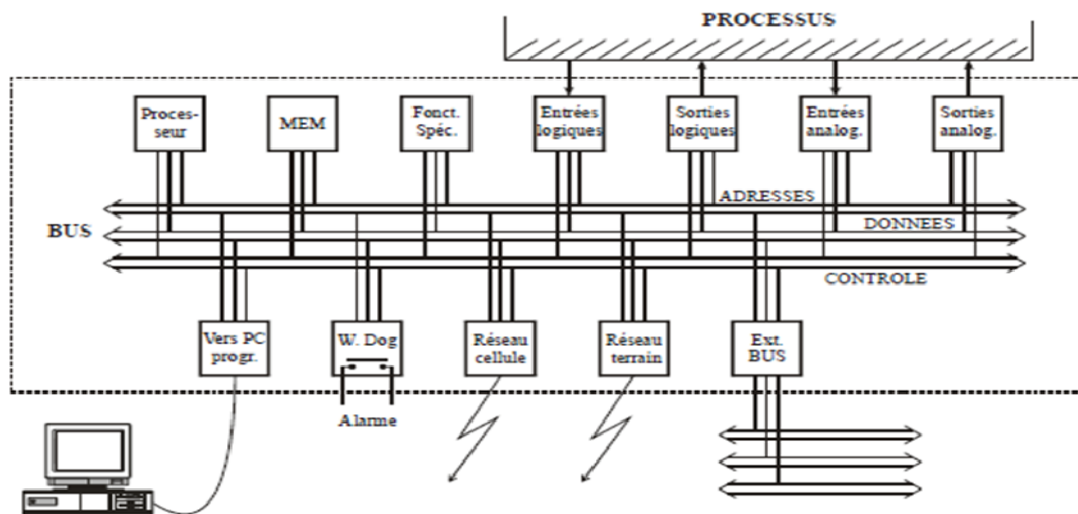


Figure II.6 : Représentation des bus internes.

II.7 Cycle de fonctionnement d'un S7-200 :

Le fonctionnement d'un automate en marche (RUN) est cyclique, Il est caractérisé par son temps de lecture, quelque millisecondes. On peut le représenter par le diagramme de la figure II.7. [9]

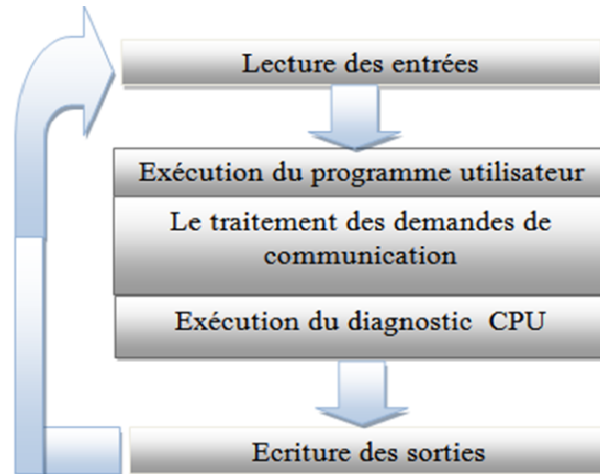


Figure II.7 : Le cycle de fonctionnement d'un S7-200.

➤ **Lecture des entrées :**

Le S7-200 copie l'état des entrées physiques, parvenues des différents capteurs, dans la mémoire image des entrées (MIE).

➤ **Exécution du programme utilisateur :**

La CPU, fait une lecture du programme utilisateur puis exécute les opérations indiquées à l'intérieur par l'appel du système d'exploitation au programme principal OB1. A la fin de l'exécution elle sauvegarde les résultats dans différentes zones de la mémoire.

➤ **Le traitement de toute demande de Communication :**

Pendant la phase de traitement des messages du cycle, le S7-200 traite tous les messages reçus de l'interface de communication ou des modules d'E/S intelligents.

➤ **Exécution du teste d'autodiagnostic de la CPU :**

La CPU S7-200 s'assure que le microprogramme, la mémoire de programme et les modules d'extension présents fonctionnent correctement.

➤ **Ecriture des sorties :**

Les valeurs enregistrées dans la mémoire image des sorties sont ensuite écrites dans les sorties physiques (les sorties raccordées aux différents prés-actionneurs).

II.8 L'adressage :

L'identification des opérandes d'opérations à l'intérieur d'un programme utilisateur s'appelle l'adressage. La CPU S7-200 range les informations à différents emplacements de la mémoire ayant chacun une adresse unique. On peut accéder à ces adresses soit par adressage absolu, soit par adressage symbolique. [9]

II.8.1 Adressage absolu :

L'adresse absolue d'un opérande est composée soit:

- D'un identificateur d'opérande, de l'adresse d'octet et du numéro de bit. Cette désignation est utilisée lorsque l'accès à la zone de données se fait sur un bit. Un exemple est donné par la figure II.7 [9]

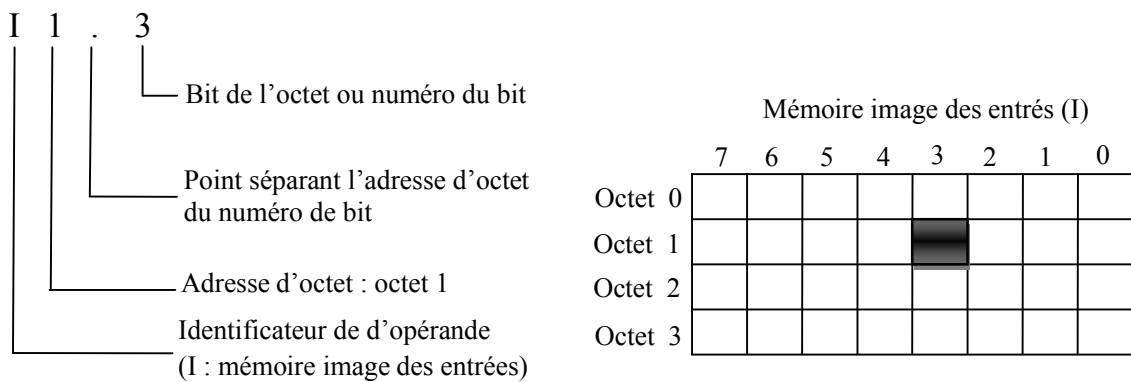


Figure II.7 : Accès par bit à la zone de donnée.

- D'un identificateur d'opérande, de la taille de donnée et de l'adresse d'octet de départ. Cette désignation est utilisée lorsque l'accès à la zone de données se fait par octet, mot ou double mot. Un exemple d'accès aux données de la mémoire image des entrées est donné par la figure II.8.

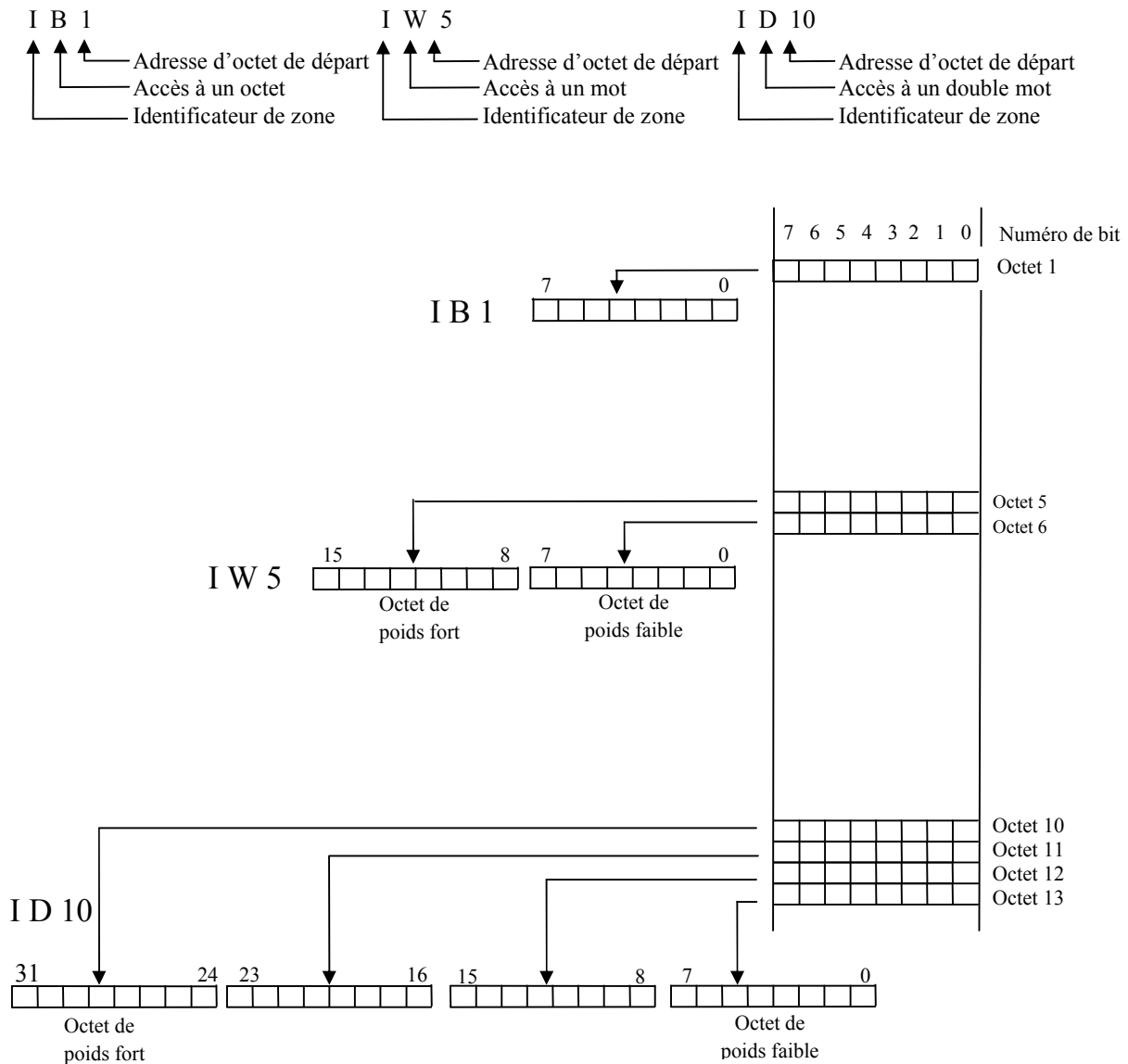


Figure II.8 : Accès à la MIE par octet, mot et double mot.

II.8.2 Adressage symbolique :

L'adressage symbolique consiste à affecter des mnémoniques (des noms symboliques) aux opérandes des opérations. Son utilisation permet de suivre plus facilement la logique du programme. Par exemple :

Adressage d'une entrée TOR: - adresse symbolique (Entrée1).
- adresse absolu (I0.0).

Adressage d'une sortie TOR: - adresse symbolique (moteur1).
-adresse absolu (Q2.2).

II.8.3 Adressage des E/S locales et des E/S d'extension:

Les CPU fournissent un nombre limité d'adresses d'entrée/sortie locale ajouter des entrées/sorties à la CPU S7-200 en y raccordant des modules d'extensions. Les adresses des entrées et sorties du module sont déterminées par le type d'E/S et la position du module dans la séquence, en tenant compte du module d'entrées ou de sorties précédent de même type. Par exemple, un module de sorties n'a pas d'influence sur les adresses d'un module d'entrées, et inversement de même. [9]

- Entrées TOR :

Au début de chaque cycle le S7-200 lit et convertit les signaux d'entrées physiques Tout Ou Rien en valeurs binaires, qui seront ensuite écrites et stockées dans la zone MIE. On accède à ces valeurs soit par bit, octets, mots ou doubles mots, dont la notation de l'identificateur de zone sur STEP7-miro/Win est 'I'.

- Sorties TOR :

A la fin de chaque cycle le S7-200 convertit les valeurs binaires stockées dans la zone MIS en signaux physiques TOR. On accède à ces valeurs soit par bit, octets, mots, et doubles mots dont la notation de l'identificateur de zone sur STEP7-miro/Win est 'Q'.

- Entrées analogiques :

La CPU S7-200 convertit une valeur analogique (telle que la température, le débit ou la tension) en valeur numérique de 16 bits (un mot). On peut accéder à ces valeurs par l'identificateur de zone (AI), la taille des données (W), et l'adresse d'octet de départ. Comme les entrées analogiques sont des mots et commencent toujours sur des octets pairs (tels que 0, 2 ou 4), on y accède par des adresses d'octet paires (par exemple : AIW0, AIW2, AIW4). Seule la lecture des entrées analogiques est possible. [9]

- Sorties analogiques :

La CPU convertit une valeur numérique de 16 bits (mot) issus de l'exécution du programme utilisateur en un courant ou une tension proportionnelle à cette valeur. On écrit ces valeurs numériques sur les sorties physiques via l'identificateur de zone (AQ), la taille des données (W), et l'adresse d'octet de départ. Comme les sorties analogiques sont des mots et commencent toujours sur des octets pairs (tels que 0, 2 ou 4), on y accède par des adresses d'octet paires (exemple : AQW0, AQW2 ou AQW4). Seule l'écriture des sorties analogiques est possible. [9]

La figure II.9 illustre un exemple de numérotation des entrées/sorties d'une CPU 224 comportant :

- 14 entées et 10 sorties locales TOR.
- 5 modules d'extension dont :
 - Module 0 : 4E /4S TOR.
 - Module1 : 8 entrées TOR.
 - Module2 : 4E analogique, 1S analogique.
 - Module3 : 8 sorties TOR.
 - Module4 : 4E analogique, 1S analogique.

| CPU 224 14E/10S | | 4E/4S | 8E | 4E analogique 1S analogique | 8S | 4 E analogique 1 S analogique |
|--------------------|-------------|-------------------------|-----------------|--------------------------------|-----------------|----------------------------------|
| I0.0 | Q0.0 | Module 0 | Module 1 | Module 2 | Module 3 | Module 4 |
| I0.1 | Q0.1 | I2.0 Q2.0 | I3.0 | AIW0 AQW0 | Q3.0 | AIW8 AQW4 |
| I0.2 | Q0.2 | I2.1 Q2.1 | I3.1 | AIW2 AQW2 | Q3.1 | AIW10 AQW6 |
| I0.3 | Q0.3 | I2.2 Q2.2 | I3.2 | AIW4 | Q3.2 | AIW12 |
| I0.4 | Q0.4 | I2.3 Q2.3 | I3.3 | AIW6 | Q3.3 | AIW14 |
| I0.5 | Q0.5 | I2.4 Q2.4 | I3.4 | | Q3.4 | |
| I0.6 | Q0.6 | I2.5 Q2.5 | I3.5 | | Q3.5 | |
| I0.7 | Q0.7 | I2.6 Q2.6 | I3.6 | | Q3.6 | |
| I1.0 | Q1.0 | I2.7 Q2.7 | I3.7 | | Q3.7 | |
| I1.1 | Q1.1 | E/S d'extension | | | | |
| I1.2 | Q1.2 | | | | | |
| I1.3 | Q1.3 | | | | | |
| I1.4 | Q1.4 | | | | | |
| I1.5 | Q1.5 | | | | | |
| I1.6 | Q1.6 | | | | | |
| I1.7 | Q1.7 | | | | | |
| E/S locales | | | | | | |

Figure II.9 : Exemple de numérotation d'E/S de la CPU 224.

II.9 Les Mémoires :

➤ **Mémoire ROM (Read Only Memory):**

Est une mémoire morte non effaçable qui contient le système d'exploitation nécessaire à l'exécution du programme d'application. [1]

➤ **Mémoire RAM (Random Access Memory):**

La mémoire RAM est une mémoire vive, qu'on peut lire et sur laquelle on peut écrire. Elle Possède un nombre bien déterminé d'emplacements ou cellules mémoires. On peut comparer une RAM à une armoire comportant un nombre finis de tiroirs (cellules mémoires), dont chaque tiroir présente une certaine taille ainsi qu'un numéro d'adresse. Pour stocker une information dans la cellule mémoire, il suffit de sélectionner son numéro d'adresse dans le programme d'application et de déposer l'information dans la cellule mémoire. [1]

Lorsqu'on aura de nouveau besoin de l'information, on appelle son adresse pour la lire dans sa cellule sans pour autant l'effacer. Et lorsque cette information ne nous est plus utile, on peut, soit effacer la cellule, soit changer l'information qu'elle contient. La mémoire RAM permet de sauvegarder les applications et les données qui peuvent être perdues en cas de coupure de courant. Et pour éviter que cela se produise, les automates sont généralement dotés d'une pile de sauvegarde rechargeable. [1]

➤ **Les mémoires EEPROM et EPROM :**

C'est sur ces mémoires qu'on conserve le programme utilisateur. Leur emplacement dans un API est situé sur l'unité centrale (CPU). Si l'automate a subi un effacement générale ou si la tension de secteur a été coupée alors que l'automate ne contenait pas la pile de sauvegarde, le programme utilisateur sera recharger automatiquement de la cartouche mémoire vers la RAM à la prochaine mise sous tension de l'automate.

EPROM (**Erasable Programmable Read Only Memory**) et EEPROM (**Electrically Erasable Programmable Only Memory**) sont des mémoires mortes, effaçables et programmables. Après élaboration et test d'un programme, on peut l'inscrire et le sauvegarder dans ces mémoires à partir desquelles on pourra ensuite le transférer vers d'autres automates.

L'effacement du contenu d'une EPROM se fait au moyen de rayon ultraviolet (UV), l'effacement est intégral et l'EPROM peut être ensuite programmée. L'EEPROM peut être effacée et modifiée électriquement. Les cellules mémoires peuvent être réécrites, c'est-à-dire que l'on peut modifier les instructions d'un programme utilisateur. [7]

II.9.1 Les différentes zones de mémoire d'une CPU S7-200 :

Le S7-200 comporte trois zones mémoires importante, utilisées pour la conservation du programme utilisateur et les données : [9]

➤ **La cartouche mémoire (EEPROM) :**

Qui est une mémoire non volatile qu'on peut débrancher une fois que le programme transféré a été stocké en mémoire permanente. Elle sert à l'enregistrement du programme utilisateur (le bloc de code, le bloc de données, le bloc de données système, des recettes, des journaux de données et des valeurs forcées) et ce avec l'utilisation de l'Explorateur S7-200 qui permet de stocker les fichiers de documentation (doc, texte, PDF...) dans la cartouche. [7]

➤ **La mémoire permanente (EEPROM) :**

Qui est une mémoire non volatile utilisée pour stocker les données transférées à partir de la cartouche mémoire tel que le bloc de code, le bloc de donnée, le bloc de données système, les valeurs forcées. Les mementos configurés pour être sauvegardés en cas de coupure de courant, ainsi que les valeurs sélectionnées écrites sous la commande du programme utilisateurs. [7]

➤ **La mémoire de données rémanente (RAM) :**

C'est une zone de mémoire de données que l'utilisateur sélectionne afin qu'elles restent inchangées en cas de mise hors tension, puis sous tension, et ce tant que le super condensateur et la cartouche pile optionnelle ne sont pas déchargés.

La RAM d'un S7-200 contient les zones mémoires suivante I, Q, V, M, T, C, AC, SM, L, S,...). Mais seul les zones V, M (zone des variables, et zone mémentos respectivement), valeur en cours des temporisations et valeurs en cours des compteurs qu'on peut configurer comme rémanentes. Par défaut, les 14 premiers octets de la zone des mémentos (MB0 à MB13) ne sont pas rémanents.

A une mise sous tension, le S7-200 restaure le bloc de code et le bloc de données système à partir de la mémoire rémanente, et vérifie si le super condensateur et la cartouche pile optionnelle ont correctement conservés les données en mémoire vive.

- Si c'est le cas, les zones rémanentes de la mémoire de données utilisateurs restent inchangées.
- Dans le cas échéant, si le contenu de la mémoire vive n'a pas été conservé (par exemple, après une longue coupure de courant), le S7-200 efface toutes les zones de données utilisateur ; met le dit SM0.2 (Données rémanentes perdues) à 1, restaure la mémoire V à partir du contenu du bloc de données en mémoire permanente, et restaure les 14 premier octets de mémentos (mémoire M) à partir de la mémoire permanente, et ce s'ils avaient été auparavant configurés comme rémanents. [9]

II.10 Avantages des automates programmables :

- L'API est indépendant du procédé, il est facilement réutilisable.
- Il présente une grande souplesse de mise en œuvre.
- Sa compressibilité conduit à une économie de place et une grande fiabilité.
- La maintenance préventive et curative des systèmes est simplifiée.
- L'API est favorable aux traitements évolués (calcul numérique, régulation).
- Il facilite la documentation des applications, donc leur maintenance.
- Il est bien adaptés à la surveillance en ligne de fonctionnement de la loi de commande (visualisation des entrées/sorties, des traitements logiques...).
- Les API permettent d'ajuster la disponibilité du système aux besoins.
- Un personnel de qualification moyenne peut le manipuler facilement.
- Il offre une sécurité totale en cas d'incendie, ou anomalie sur le processus. [8]

II.11 Inconvénients des automates programmables :

- les automates ne suppriment pas tous les reliages, car il existe toujours un circuit de puissance qui est câblé.
- Les modules de recherche sont plus chers.
- Difficultés rencontrés au niveau de choix d'un automate en fonction du besoin.
- Le déroulement cyclique du programme peut s'avérer un facteur de complexité et limite les possibilités d'organisation des tâches. [8]

II.12 La sûreté de fonctionnement :

La sûreté est une contrainte majeure des automatismes. Elle a, de ce fait été pris en compte dans la conception et l'utilisation des API, dès le début de leur existence. Pourtant parallèlement au développement de l'utilisation des automates, et profitant des améliorations technologiques qui permettent des exigences plus grandes, la sûreté est devenue un thème désormais essentiel. [2]

II.12.1 Les différents aspects du concept de sûreté :

Les API ont un champ d'application de plus en plus étendu. On les intègre dans des systèmes aux fonctions complexes ou dans des automatismes de grande envergure. Leur rôle devient essentiel et il est important d'examiner les différentes natures de dysfonctionnement qui peuvent affecter la bonne marche du système. Les principaux critères relatifs à la sûreté sont : [2]

- **La fiabilité :**

La fiabilité d'un automate dépend de ses conditions d'utilisation, mais aussi et en premier lieu de sa conception, et en second lieu la façon dont est assuré le refroidissement du matériel. On peut dire donc que la température de ses composants et ses variations, influent sur la fiabilité du matériel.

- **La sécurité :**

La sécurité est la capacité du système (API d'abord, automate ensuite) à réagir convenablement à des défaillances de ses constituants, sans provoquer d'accidents de personne ni de dommage des installations.

L'idée d'automatisme suggère l'absence de l'homme au niveau de la production. Toutefois, ce dernier intervient régulièrement pour la surveillance, la maintenance et les modifications du système.

La sécurité ne doit pas reposer sur la fiabilité du système, elle doit être assurée en toutes circonstances.

- **La disponibilité :**

Il est dans la nature des choses que des pannes se produisent, celles-ci ont cependant des conséquences intégrales sur l'automatisme et les fonctions réalisées (aspects qualitatif et quantitatif de la production).

La disponibilité d'un système est le rapport de son temps de fonctionnement normal au temps d'exploitation total. Il est en fonction de la fiabilité et de temps d'intervention nécessaire à la réparation.

II.12.2 Les dispositifs généraux de sécurité :

La sécurité est devenue de nos jours une condition primordiale dans le monde de l'industrie. Les constructeurs ne cessent de créer de nouveaux dispositifs afin d'accroître au maximum le niveau de sécurité. Ces dispositifs sont destinés à détecter les incidents

survenant à l'unité centrale et, éventuellement, à y remédier. Quatre types de problèmes doivent être pris en compte :

- le fonctionnement normal du processeur.
- l'intégrité des informations mémorisées ou échangées.
- les pannes d'alimentation.
- Les pannes des entrées sorties. [2]

➤ **sécurité du processeur :**

La sécurité du processeur porte sur la bonne exécution des instructions et sur le déroulement convenable des programmes. Pour cela la première condition est que les organes logiques soient en bon état de fonctionnement. De plus en plus l'API dispose d'une fonction de test systématique de leur logique.

Bien entendu des précautions similaires sont mises en place pour les systèmes multiprocesseurs au sein des processeurs eux mêmes (processeur général, unité centrale.....).

➤ **sécurité des alimentations :**

Les alimentations électriques des API sont la plupart du temps branchées directement sur le secteur. Ce qui provoque des variations de fréquence et de tension sur le fonctionnement des API. Pour cela le niveau d'alimentation est surveillé. Il est branché sur une interruption prioritaire interne. Une anomalie déclenche une procédure de sauvegarde, se déroulant en quelques millisecondes grâce à leur batterie de secours. Ce temps est mis à profit pour ranger le contenu des registres de l'unité centrale en mémoire. Lorsque celle-ci est volatile, une batterie de secours permet de sauvegarder son contenu pendant plusieurs jours, voir plusieurs mois.

➤ **La sécurité des entrées/sorties :**

L'expérience indique que 80 à 90% des pannes d'une logique programmable se situent au niveau des entrées/sorties. Bien que certaines machines soient garanties pour une parfaite disponibilité, il est indispensable de prendre toutes les précautions utiles pour éviter qu'un incident ne se répercute sur le fonctionnement du procédé. Un premier niveau de sécurité est assuré par l'automate lui-même, le second inclut le procédé.

II.12.3 La sûreté du procédé :

Il s'agit là d'un aspect récent de l'objectif de sureté qui consiste à mettre en œuvre la capacité logique des automates et leur fiabilité pour assurer la sureté du procédé lui-même. Cette nouvelle approche s'appuie à la fois sur les capacités des entrées/ sorties des API, sur leur moyens de gestion des données et sur les procédures du traitement approprié. [2]

II.13 Classification des pannes :

Il ne saurait être question de proposer une liste exhaustive structurée des incidents en pannes susceptibles de survenir dans un système automatisé et dans l'API qui le gère. Il est cependant possible de procéder à une classification de types d'incidents et d'offrir au travers d'un bref commentaire, des thèmes de réflexion au praticien qui analyse son problème. [2]

- **Pannes classées selon leur objet :**

Les défaillances peuvent affecter l'API lui-même, les entrées/sorties, le procédé, les dispositifs d'alimentation électrique, et autres dispositifs de configuration ou d'installation.

- **Pannes classées selon leur origine :**

Différentes causes peuvent générer des incidents ; l'électronique de l'API d'abord, puis le circuit intégré, la mémoire, les composants discrets de puissance, les bus, l'horloge interne, et les connecteurs. Ils résultent du vieillissement, d'incidents mécaniques à l'assemblage, ou lors d'une intervention de maintenance.

- **Pannes classées selon leur durée :**

On distingue deux catégories :

- les pannes permanentes : des pannes franches bien identifiables (Courts-circuits, circuit ouvert,.....).
- les pannes fugitives ou intermittentes : Connues aussi par leurs effets aléatoires, elles sont typiques des systèmes informatiques (variation anormale des tensions d'alimentation des circuits intégrés, défaut de masse, défaut de synchronisation ou de propagation).

- **Pannes selon leurs effets :**

L'expérience des sites automatisés ont permis de cerner les effets des différents dysfonctionnements et, en particulier les conséquences de perturbations externes sur certains API. On relèvera à titre d'illustration les suivantes :

- Blocage définitif d'un programme (arrêt).
- Altération des sorties, Blocage, et commutation impropre.
- Destruction des composants (UC, mémoire, E/S).

Conclusion :

L'évolution des automates programmables au cours de ces dernières années a modifié son organisation, ses performances et l'ensemble des procédés industriels qui l'ont utilisé. Sa puissance de traitement a été multipliée par dix, et sa taille mémoire a été aussi accrue. Ces progrès fulgurants de la microélectronique offrent à l'industrie et aux autres secteurs, des systèmes de commande pour des coûts constants et avec des performances qui ne cessent d'augmenter.

Chapitre III

*Description technique de la machine
d'impression KOMORI*



III.1 Introduction :

L'imprimerie est un ensemble de techniques permettant de reproduire des écrits et des illustrations en grande quantité sur des supports plans généralement en papier, et pouvoir ainsi les distribuer à grande échelle.

Les techniques fréquemment appelées (chaîne graphique) vont de la composition de texte au façonnage (découpage, reliure, pliure,...) en passant par le traitement des illustrations (photogravures, photos, dessins, gravures), l'impression et la lecture. L'évolution de la technologie de l'industrie graphique a permis l'amélioration de l'imprimerie (qualité d'impression, coût d'impression....). [13]

III.2 Description technique d'une machine rotative:

Une rotative est composée essentiellement de sept (07) éléments :

- un dérouleur.
- Le débiteur
- un groupe imprimant.
- un sécheur.
- un refroidisseur.
- une plieuse/découpeuse.



Figure III.1 : Exemple d'une machine d'impression rotative

Ces éléments sont schématisés sur la figure III.2 ci-dessous :

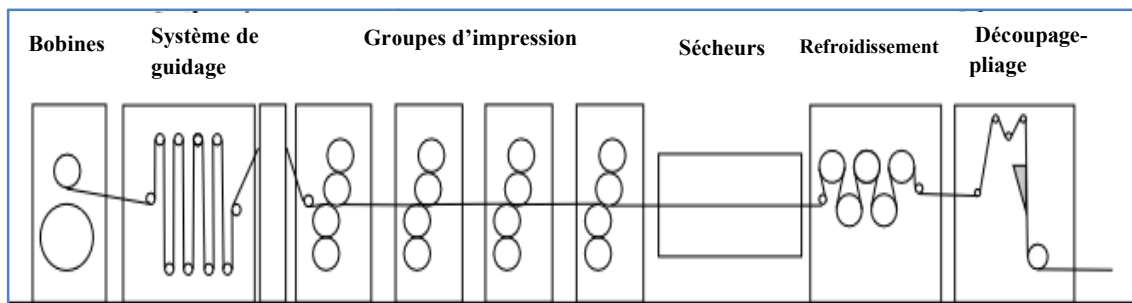


Figure III.2 : Les différents éléments d'une machine rotative.

III.2.1 Principe de fonctionnement :

Afin de répondre aux exigences du domaine de l'imprimerie et satisfaire les besoins de la demande du marché, la machine rotative assure deux fonctions importantes :

- le pliage du papier imprimé.
- le découpage du papier imprimé.

III.2.1.1 Cycle de découpage :

Le dérouleur déroule une bobine de papier en fonction de la vitesse de défilement de la bande, le débiteur assure la tension de la bande pour permettre d'avoir une bonne tension.

Les groupes d'impression sont obligatoirement au nombre de 04 pour faire de la quadrichromie et peuvent selon la commande avoir un nombre supérieur. Les séquences de la quadrichromie sont noire, rouge, vert et bleu. Elle sera respectée pour permettre d'avoir un meilleur rendement et éviter les refus d'impression. Le sécheur a pour fonction de sécher l'encre. Il retire les huiles de l'encre pour le séchage, et la bande de papier à la sortie du four sera entre 90 et 100 degrés. Le papier sera refroidi par le refroidisseur pour être enfin prêt à être découpé par le bloc de découpage. [13]

III.2.1.2 Cycle de pliage :

Le déroulement du cycle de pliage du papier imprimé est pratiquement le même que celui du découpage. Le papier passe par les blocs de dérouleur, d'impression, sécheur, refroidisseur pour passer en suite à la préparation du papier pour le pliage. La plieuse effectuera deux opérations: tirer le papier pour l'ensemble de la machine et effectuer quatre plis possibles. [13]

III.3 Problématique :

Dans ce projet, on s'intéresse particulièrement à l'étude du bloc de découpage, cette partie importante de la machine subit beaucoup de perturbations comme :

- Oscillation du papier dans le sécheur (dans ce bloc le papier est tendu sur 12 mètres et subit un séchage avec de l'air chaud qui le fait osciller).
- Variations des dimensions de la bande de papier lors du séchage et du refroidissement.
- Changement de la tension du papier.
- Apparition de défauts sur la machine (petits déplacement des rouleaux de Transferts,..)

Ces perturbations provoquent des mauvaises coupes (coupe au mauvais endroit). Et pour résoudre ces problèmes le bloc est équipé de :

- Détecteur de spot (photocellule).
- Un moteur asynchrone monophasé de régulation à deux sens de rotation.
- Une carte électronique.

Ce système assure un bon fonctionnement du bloc de découpage jusqu'à ce qu'un dysfonctionnement de la carte électronique survienne causant des erreurs de coupes de quelques millimètres allant jusqu'à un centimètre. Cela nécessite généralement l'arrêt de production vu le nombre important de feuilles défectueuses qui auront été enregistrées.

III.4 Solution proposée :

Afin de rétablir le bon fonctionnement de la machine rotative à découpage, il existe deux solutions admissibles: Remplacer la carte électronique ou installer un dispositif qui

puisse assurer la même fonction. Mais vu la non disponibilité et le prix élevé de cette dernière, on a opté pour la deuxième solution.

Le système adopté comporte les éléments suivants :

- Un automate programmable industriel S7 200.
- Deux encodeurs incrémentaux.
- Un détecteur de spot (photocellule).
- Un capteur inductif.
- Un moteur asynchrone monophasé à deux sens de rotation.

Le déroulement de ce cycle avec ces nouveaux éléments est illustré dans la figure ci-dessous :

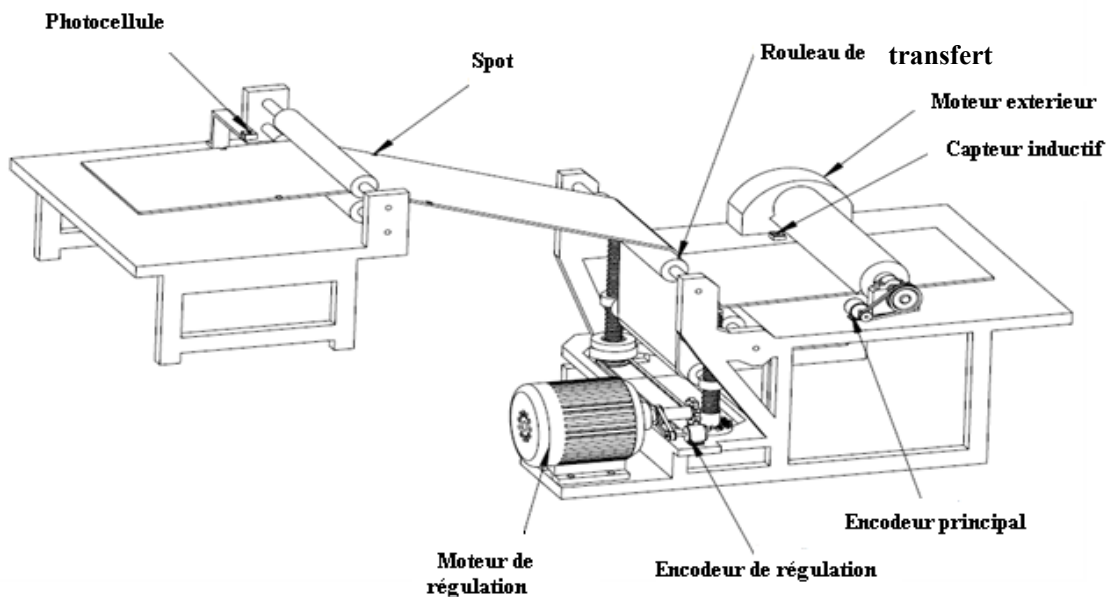


Figure III.3: Bloc de découpe après installation du dispositif.

III.5 Présentation des éléments du nouveau dispositif :

III.5.1 Automate programmable industriel S7 200 :

Voir le chapitre II.

III.5.2 L'afficheur TD 200 :

Le TD 200 est un afficheur de texte qui sert d'interface opérateur aux automates programmables de la famille S7-200. Le TD 200 fonctionne comme un maître à interface point-à-point (PPI) lorsqu'il est connecté à une ou plusieurs CPU S7-200, il peut également être combiné à un maître PPI quelconque dans un réseau. Plusieurs TD 200 peuvent être utilisés en combinaison avec une ou plusieurs CPU S7-200 connectées au même réseau. [10]

III.5.2.1 Les fonctions du TD 200 :

- Affichage des messages en provenance de la CPU S7-200.
- Modification des variables configurées.
- Forçage des entrées et sorties et annulation du forçage.
- Réglage de l'heure et de la date sur les CPU pourvues d'une horloge à temps réel.

III.5.2.2 Composants du TD 200

Le TD 200 est un petit appareil compact qui possède tous les composants nécessaires pour servir d'interface avec la CPU S7-200. La figure III.4 présente les principaux composants du TD 200. L'alimentation électrique du TD 200 est fournie soit par la CPU S7-200 via le câble TD/CPU, soit par une alimentation séparée.

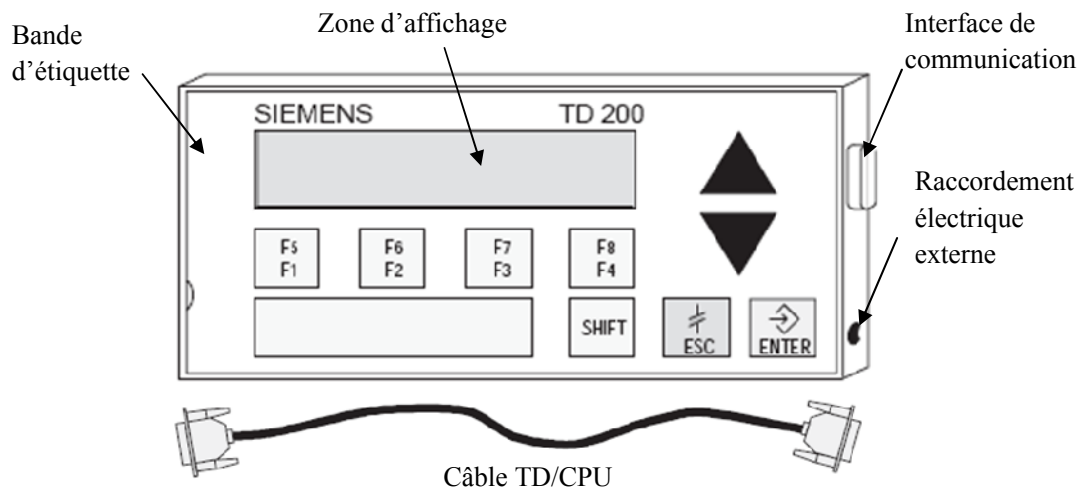


Figure III.4 : Principaux composants du TD.

- **Zone d'affichage** : La zone d'affichage est un écran à cristaux liquides (LCD) à rétroéclairage comportant deux lignes de 20 caractères. Il permet de visualiser les messages en provenance de la CPU S7-200.
- **Interface de communication** : L'interface de communication est un connecteur D à 9 points grâce auquel on peut relier le TD 200 à une CPU S7-200 via le câble TD/CPU fourni en accompagnement.
- **Raccordement électrique** : On peut raccorder le TD 200 à une alimentation externe via le connecteur situé sur le côté droit du TD 200. Ce connecteur n'est pas nécessaire lorsqu'on utilise le câble TD/CPU.
- **Câble TD/CPU** : Le câble TD/CPU rend possible l'alimentation du TD 200 et la communication avec celui-ci. Il s'agit du câble droit à 9 points fourni avec le TD 200.

- **Bande d'étiquette** : La bande d'étiquette est amovible et permet à chaque utilisateur d'adapter le libellé aux touches de fonction de l'application.
- **Clavier** : Le clavier du TD 200 comporte neuf touches. Cinq d'entre elles permettent l'accès à des fonctions prédéfinies et contextuelles, appelées (touches de commande). Les quatre autres permettent l'accès à des fonctions définies par l'utilisateur appelées les touche de fonction.

III.5.2.3 Les caractéristiques du clavier TD 200 :

- **Les caractéristiques des touches de commande :**
 - **Entrée** : Cette touche sert pour entrer de nouvelles données et pour accuser réception d'un message.
 - **Echap** : Cette touche sert pour commuter du mode d'affichage des messages au mode menu ou pour annuler une édition.
 - **Flèche vers le haut** : permet d'incrémenter une valeur et d'amener le curseur sur le message de priorité immédiatement supérieure.
 - **Flèche vers le bas** : permet de décrémenter une valeur et d'amener le curseur sur le message de priorité immédiatement inférieure.
 - **SHIFT (Maj)** : permet de dédoubler la fonction de toutes les touches de fonction. Lorsqu'on appuie sur la touche SHIFT, un "S" clignotant apparaît dans la partie inférieure droite du TD 200.
- **Les caractéristiques des touches de fonction :**
 - **F1** : La touche de fonction F1 met le bit Vx.0 à 1. Si on appuie sur la touche SHIFT en même temps ou avant la touche F1, la touche de fonction F1 met le bit Vx.4 à 1.
 - **F2** : La touche de fonction F2 met le bit Vx.1 à 1. Si on appuie sur la touche SHIFT en même temps ou avant la touche F2, la touche de fonction F2 met le bit Vx.5 à 1.
 - **F3** : La touche de fonction F3 met le bit Vx.2 à 1. Si on appuie sur la touche SHIFT en même temps ou avant la touche F3, la touche de fonction F3 met le bit Vx.6 à 1.
 - **F4** : La touche de fonction F4 met le bit Vx.3 à 1. Si on appuie sur la touche SHIFT en même temps ou avant la touche F4, la touche de fonction F4 met le bit Vx.7 à 1.

III.5.2.4 Les caractéristiques du TD 200 Version 3.0 :

Le TD 200 V3.0 accepte plusieurs jeux de langue, des menus utilisateurs personnalisés, des données CPU affichées sous forme de chaînes, des messages avec icônes intégrées et 9 boutons ayant des fonctions fixes. [9]

III.5.2.5 Fonctionnement des touches de l'afficheur :

Les touches du clavier de notre afficheur TD 200, sont configurées de telle manière à accomplir des fonctions bien précises, représentées dans le tableau ci-dessous :

| Bouton poussoir | L'adresse dans la mémoire V | Fonctionnement |
|-----------------|-----------------------------|--|
| F3 | V961.2 | Active la régulation. |
| SHIFT_F3 | V963.6 | Désactive la régulation. |
| F1 | V961.0 | Active le mode de régulation manuelle et désactive le mode de régulation automatique. |
| SHIFT_F1 | V963.4 | Active le mode de régulation automatique et désactive le mode de régulation manuelle. |
| F4 | V961.3 | - En mode manuel permet de commander la montée manuelle du rouleau de transfère (en activant Q0.0). - En mode automatique utilisé comme l'entrée de décrémentation du compteur incrémentale/décémentale CTUD (C30). |
| F2 | V961.1 | - En mode manuel permet de commander la descente manuelle du rouleau de transfère (en activant Q0.1). - En mode automatique utilisé comme l'entrée d'incrémentement du compteur incrémentale/décémentale CTUD (C30) |
| SHIFT_F4 | V963.7 | - Permet l'affichage de la position réelle du spot par rapport à la deuxième coupe d'une feuille. - Permet la remise à zéro du compteur C30. |
| SHIFT_F2 | V963.5 | Permet la réinitialisation de la table (TBL). |

III.5.3 Encodeurs rotatifs industriels (encodeurs optiques) :

La croissance de la puissance des systèmes de traitement ainsi que les impératifs de productivité appellent dans tous les domaines de production industrielle un besoin d'information continue sur :

- le déplacement.
- la position.
- la vitesse des outils ou des produits.

Les systèmes de détection conventionnels (interrupteurs et détecteurs de positions) qui ne peuvent fournir que des informations Tout Ou Rien à des endroits prédéterminés, ne répondent que partiellement aux besoins de précision et de flexibilité.

Dans le cas d'un encodeur, le positionnement du mobile est entièrement maîtrisé par les systèmes. la figure III.5 illustre un exemple d'utilisation d'un encodeur. [15]

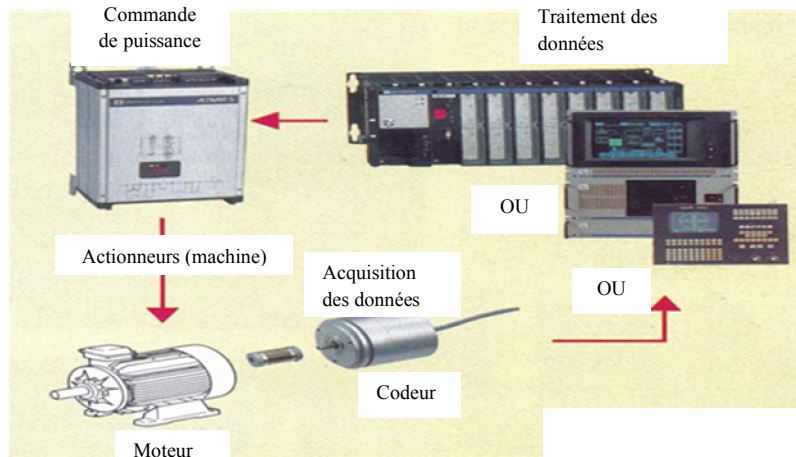


Figure III.5 : Exemple d'utilisation d'un codeur.

- **Encodeur optique :**

Un encodeur optique est un dispositif électromécanique dont la sortie électrique est représentée sous forme numérique. L'axe d'entrée lié mécaniquement à un arbre qui l'entraîne, fait tourner un disque qui lui est solidaire. Le disque comporte une succession de parties opaques et transparentes. Une lumière émise par des Diodes Electroluminescentes (DEL). Traverse les fentes de ce disque créant sur les photodiodes réceptrices un signal analogique. Électroniquement ce signal amplifié puis converti en signal carré est alors transmis à un système de traitement.

Un encodeur optique est donc composé de trois parties essentielles comme le montre la figure suivante (III.6). [15]

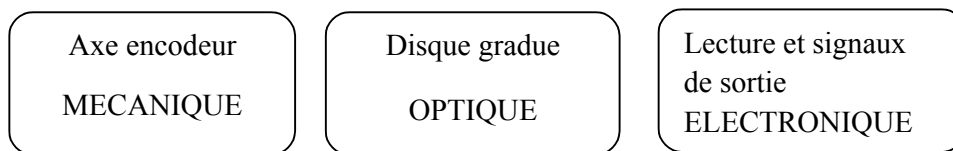


Figure III.6 : Les trois parties d'un encodeur optique.

III.5.3.1 Les différents types d'encodeurs

Il existe deux types d'encodeurs optiques : « incrémental » et « absolu ».

- **Encodeur incrémental (générateur d'impulsion):**

On a recours à ce type d'encodeur car il est très utile là où la précision est exigée, comme c'est le cas dans notre travail. Cet appareil délivre un nombre d'impulsions qui va correspondre à la position exacte de l'arbre du moteur qui l'entraîne.

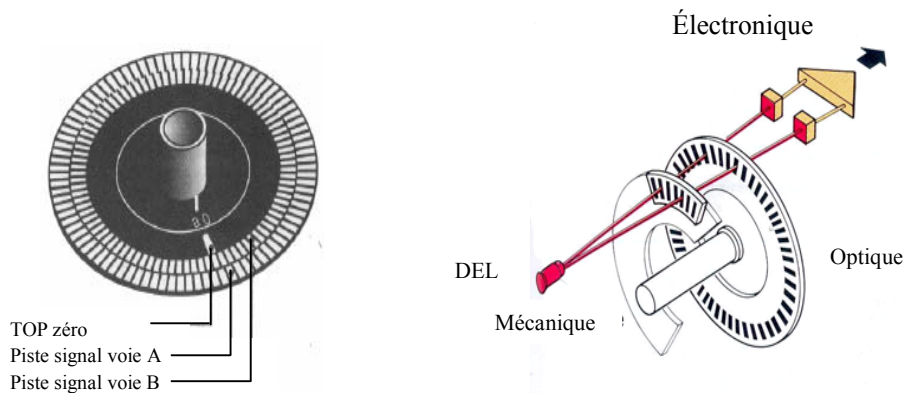


Figure III.7 : Encodeur incrémental.

Le disque comporte au maximum 3 pistes, Une ou deux de ses pistes extérieures sont divisées en (n) intervalles d'angles égaux alternativement opaques et transparents.

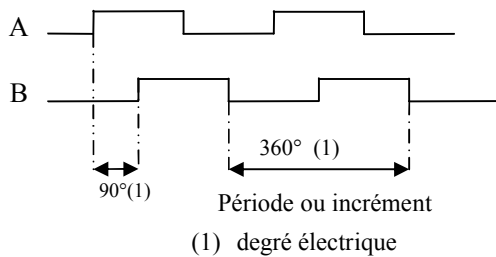


Figure III.8 : Les horloges A et B de l'encodeur.

-Pour un tour complet de l'encodeur, le faisceau lumineux est interrompu (n) fois et délivre (n) signaux carrés (A et B) en quadrature.

-Le déphasage de 90° électrique des signaux A et B permet de déterminer le sens de rotation qui est représenté sur la figure III.9

- Dans un sens pendant le front montant du signal A, le signal B est à zéro.
- Dans l'autre sens pendant le front montant du signal A, le signal B est à un,

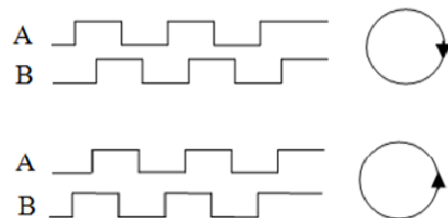


Figure III.9 : déduction du sens de rotation

- La piste intérieure (Z : top zéro) comporte une seule fenêtre transparente et délivre un seul signal par tour. Ce signal Z d'une durée de 90° électrique, détermine une position de référence et permet la réinitialisation à chaque tour. [15]

➤ **Encodeur absolu :**

Ce concept a été développé pour pallier à des contraintes générées par l'encodeur incrémental, comme par exemple la sensibilité aux coupures du réseau. Tous les segments étant de longueur égale et représentés de la même manière par les signaux A et B, chaque coupure du courant fait perdre la position réelle du mobile. Il faut alors procéder à la réinitialisation. Ce temps de réinitialisation peut être pénalisant pour certaines applications.

• **Principe théorique de fonctionnement :**

Le disque des encodeurs absolus comporte un nombre « n » de pistes concentriques divisées en segments égaux Alternativement opaques et transparents. A chaque piste est associé un couple émetteur / récepteur optique. Ce qui veut dire que chaque piste a donc son propre système de lecture.

- La piste intérieure est composée d'une moitié opaque et d'une moitié transparente. La lecture de cette piste (bit de poids le plus fort), **MSB = Most Significant Bit**, permet de déterminer dans quel demi-tour on se situe.
- La piste suivante est divisée en quatre quarts alternativement opaques et transparents.
- La lecture de cette piste combinée avec la lecture de la piste précédente permet alors de déterminer dans quel quart de tour ($1/4$) on se situe.
- Les pistes suivantes permettent successivement de déterminer dans quel huitième de tour ($1/8$), seizième de tour ($1/16$), ... etc.... on se situe.
- La piste extérieure donne la précision finale et est appelée LSB = Least Significant Bit (bit de poids le plus faible). Cette piste comporte 2^n correspondant à la résolution du codeur. Pour chaque position angulaire de l'axe, le disque fournit un « code binaire » de longueur « n » correspondant à $1/2^n$ ème de tour.

Un encodeur absolu délivre en permanence un code qui est l'image de la position réelle du mobile à contrôler. [15]

Remarque :

Dans notre projet on a opté pour l'encodeur incrémental car on n'a pas besoin de savoir la position exacte du mobile mais du changement de position de ce dernier. Les deux encodeurs utilisés sont de même type et illustré sur la figure III.10

Caractéristiques techniques :

- tension d'alimentation : 10 à 30VDC
- vitesse maximal : 6000tr/min
- résolution max 512impulsions/tour . [12]



Figure III.10: Encodeur incrémental.

Remarque :

La résolution de ce type d'encodeur est réglable selon le besoin, et dans notre cas on a utilisé les résolutions suivantes :

Encodeur principal : 512 impulsions/tour qui va nous donner au mode quadrature (mode 9) 1024 impulsions/tour

Encodeur de régulation : 256 impulsions/tour on mode normal (mode0).

III.5.4 Photocellule :**Définition :**

Les photocellules sont des capteurs optiques en terme général. Ce sont des dispositifs optoélectroniques qui détectent un rayonnement dans un domaine de longueur d'ondes qui va de l'infrarouge à l'ultraviolet, et les convertissent en signaux électriques.

Ces photocellules sont faites à base de semi-conducteurs. Lorsque l'énergie des rayons est suffisamment grande pour franchir la barrière du potentiel il va y'avoir apparition d'une différence de potentiel, on reliant ses deux pôles de potentiel un courant s'écoule d'où la conversion électrique. [15]

La figure suivante illustre la photo cellule utilisée :



Figure III.11: Image photocellule.

Cette photocellule est de la série TL10, elle se compose de détecteurs de contraste à base analogique et émission par lampes à incandescence de longue durée et simple interchangeabilité.

On trouve sur le capteur un potentiomètre mécanique multi tour pour le réglage optique de la sensibilité, ainsi qu'un commutateur pour la sélection du mode de fonctionnement sombre/clair et enfin une LED rouge qui signale l'activation de la sortie.

Caractéristiques techniques :

- Tension d'alimentation : 10 à 30 VCC.
- Emission : lampe à incandescence 4,5 V (0,8 A)
- Dimension spot : 5 x 1 mm (objectif 8 mm), 6 x 2,5 mm (objectif 28 mm), 8 x 2,5 mm (objectif 50 mm)
- Distance opérationnelle : 8 à 12 mm (objectif 8 mm), 28 à 36 mm (objectif 28 mm) 46 à 54 mm (objectif 50 mm)
- Indicateurs : LED rouge SORTIE
- Type de sortie : NPN
- Temps de réponse : 50 μ s maxi [12]

- Fréquence de commutation : 10 KHz maxi
- Mode opérationnel : cette photocellule possède deux modes opérationnels sélectionnables:
 - Sombre
 - clair

III.5.5 capteur inductif :

Définition :

Le capteur inductif détecte tout objet métallique se trouvant à sa portée. Cette détection permet de constater la présence, le passage, le déroulement, le positionnement et le comptage d'objets métalliques. Le capteur utilisé dans notre cas est représenté dans la figure suivante : [15]

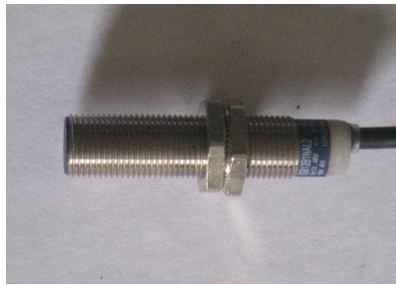


Figure III.12 : capteur inductif.

Principe de fonctionnement

Les capteurs inductifs produisent à l'extrémité de leur tête de détection un champ magnétique oscillant. Ce champ est généré par une inductance et un condensateur monté en parallèle. Lorsqu'un corps conducteur métallique est placé dans ce champ, des courants de Foucault prennent naissance dans la masse du métal. Il y'a alors perturbation de ce champ qui entraîne une réduction de l'amplitude des oscillations au fur et à mesure de l'approche de l'objet métallique, ce jusqu'à blocage complet. Cette variation est exploitée par un amplificateur qui délivre un signal de sortie, le capteur commute. [15]

Caractéristiques techniques :

- Tension d'alimentation : 12 à 48VDC
- courant de commutation : 100ma
- distance de détection : 4mm
- Type de sortie : NPN [12]

III.5.6 Moteur de régulation :

Le moteur de régulation utilisé est un moteur asynchrone monophasé alimenté par une tension de 110VAC. Il est constitué de deux enroulements (A et B) électriquement séparé de manière à ce que si on alimente la bobine 'A' le moteur va tourner dans un sens et si on alimente la bobine 'B' le moteur va tourner dans l'autre sens. Le démarrage et l'arrêt de ce moteur est contrôlé à l'aide de deux relais comme le montre la figure suivante :

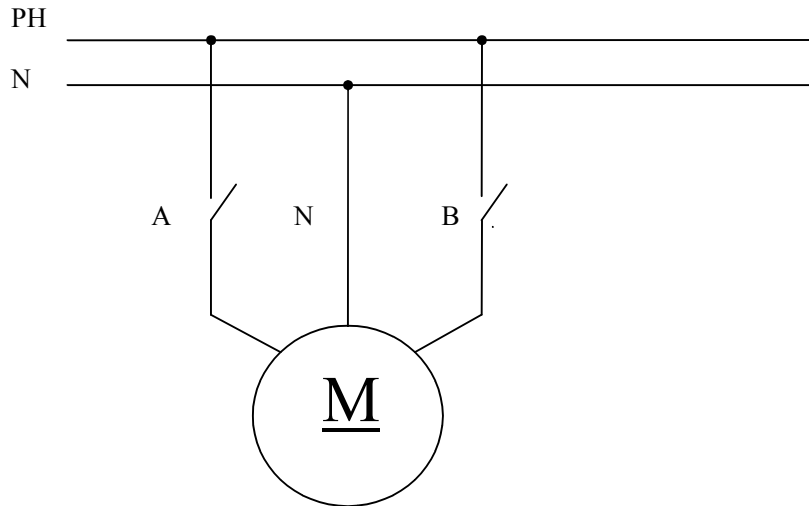


Figure III.13 : Commande du sens de rotation du moteur.

III.6 Principe de fonctionnement du dispositif :

Le papier passe par les différentes étapes citées précédemment jusqu'à la phase finale (coupe). Il est entraîné à l'aide des rouleaux de Transferts en passant par les blocs d'impressions, séchage et refroidissement qui causent principalement les perturbations. Pour y remédier, notre programme effectue quatre étapes principales.

Etape 1 : mesure de la référence (D) :

Dans le fonctionnement sans perturbations, la distance entre la position du capteur inductif et la photocellule est de sept feuilles, l'essai pratique consiste à mesurer la distance (D) en terme d'impulsions. La première instruction s'articule à faire tendre la bonde du papier d'une façon à avoir la première coupe de la première feuille au-dessous de la lame de coupe, et la huitième coupe (deuxième coupe de la septième feuille) au-dessous de la photocellule. La figure II.15 schématise le positionnement de la bonde du papier considéré avant le démarrage de l'impression.

A la mise en marche de la machine, l'encodeur principal (compteur rapide HC0) va compter le nombre d'impulsions (D2) qu'on est censé avoir entre le moment de coupe (front descendant de l'impulsion du capteur inductif) et la détection du spot (front montant de l'impulsion de la photocellule) de la huitième feuille. Ce nombre d'impulsions (D2) est soustrait de la longueur totale de la feuille (résolution de l'encodeur principal) $D1=1024$ impulsions, on obtient alors notre distance (D) qui sera affichée sur l'écran de TD 200 en appuyant sur la touche du clavier SHIFT_F4.

Etape 2 : Calcul du décalage en terme d'impulsions :

A la présence des perturbations la distance réelle (X) entre le spot et la deuxième coupe d'une feuille est empruntée d'erreurs, notre programme va effectuer l'opération (X-D) et on aura notre décalage (Y) en terme d'impulsion qui peut être positif (coupe en retard), négatif (coupe en avance) ou bien nul (coupe au bon endroit).

Etape3 : régulation :

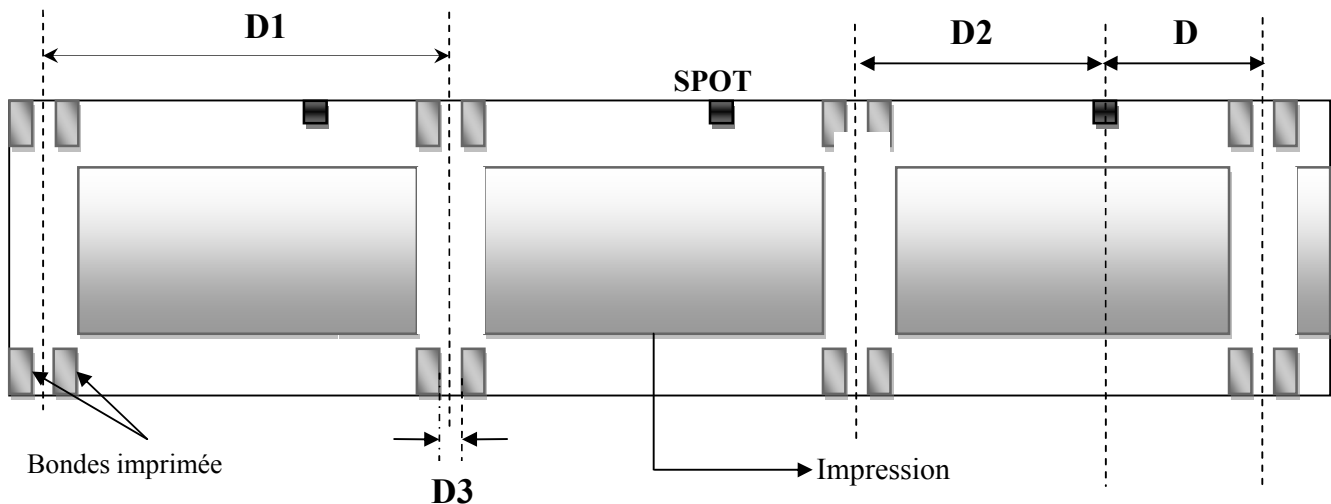
Lorsque Y est positif, l'automate donne l'ordre au moteur de régulation, par l'intermédiaire d'un relais KA1, de tourner dans le sens qui fait descendre le rouleau de transfert afin de ralentir le défilement du papier et rattraper l'avance de ce dernier.

Lorsque on a Y négatif l'automate donne l'ordre au moteur de régulation, par l'intermédiaire d'un relais KA0, de tourner dans le sens qui fait monter le rouleau de transfert pour accélérer le défilement du papier et rattraper le retard de celui-ci.

Etape 4 : fin de régulation :

Lorsque la régulation est terminée, le moteur doit s'arrêter et pour cela on la muni d'un encodeur de régulation au quel on va associer un compteur(Z) qui va compter les impulsions venant du moteur de régulation, quand la valeur du compteur atteint le nombre d'impulsion qui caractérise le décalage Y, l'automate donnera l'ordre au moteur de s'arrêter.

La figure suivante nous donne les distances entre différents repaires, longueur de la feuille imprimée, et permet aussi de mieux comprendre le nouveau dispositif.



- D** : distance spot de référence.
- D1** : longueur de feuille : 624mm.
- D2** : distance à calculer.
- D3** : erreur admissible : 4mm.

Figure III.14 : Dimensionnement de la bande du papier.

Remarque :

- Dans notre programme on a tenu compte du glissement du moteur de régulation.
- la distance exacte entre le capteur inductif et la photocellule est de 7 feuilles.

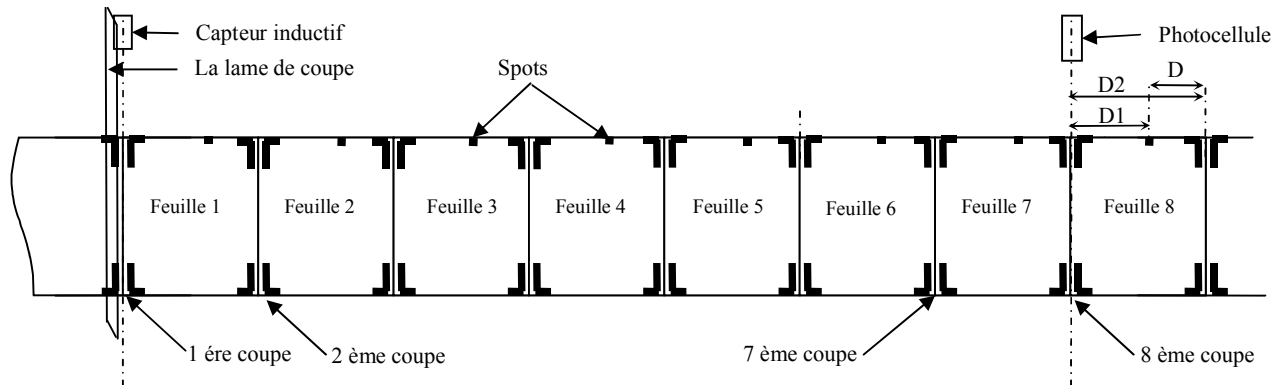


Figure II.15 : Positionnement de la bonde du papier avant le démarrage de la production

Conclusion :

Dans ce chapitre, on a procédé à la description technologique et fonctionnelle des différents éléments qui compose notre nouveau système de régulation du bloc de découpage de la machine rotative, et on a donné les arguments du choix du matériel à utiliser dicté par les besoins.

Après l'adaptation d'un nouveau système de régulation à base d'un automate programmable industriel S7 200 on passe à la programmation, et pour cela on a utilisé STEP7-MICRO/WIN avec le langage LADDER qui est adapté à ce type d'opération.

Chapitre IV

*Programme de commande
de la Machine*

IV.1 Introduction :

STEP7-Micro/WIN 32 est un logiciel de programmation conçu spécialement pour les CPU S7-200. Il fournit un environnement convivial pour concevoir, éditer et surveiller la logique nécessaire à la commande d'une application. Il comprend trois éditeurs de programme. [09]

IV.2 Les éditeurs de programmation de STEP7-Micro/WIN 32:

STEP7-Micro/WIN 32 propose trois éditeurs pour la création d'un programme, pour choisir un éditeur quelconque, on sélectionne la commande **Affichage** et on clique sur **LIST**, **CONT** ou **LOG**. [11]

IV.2.1 Editeur liste d'instruction (LIST) :

L'éditeur LIST affiche le programme sous forme textuelle. Avec l'éditeur LIST, on introduit des abréviations d'opérations quand on crée des programmes de commande, l'éditeur LIST permet également de créer des programmes qu'il n'est pas possible de représenter avec les éditeurs CONT/LD ou LOG/FBD. Il convient le mieux aux programmeurs expérimentés. La figure VI.1 porte un exemple d'un programme LIST. [11]



Figure IV.1 : Exemple d'un programme LIST.

IV.2.2 Editeur logigramme (LOG/FBD) :

L'éditeur LOG (ou FBD) affiche le programme sous forme de représentation graphique similaire aux schémas de portes logiques courantes (OR, AND, NOR...) qui sont représentées sous forme de boîtes d'opérations. La figure VI.2 donne un exemple. [11]

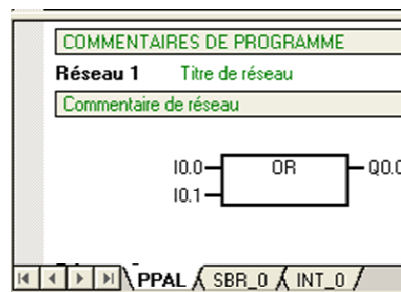


Figure IV.2: Exemple de programme LOG.

IV.2.3 Editeur schéma à contacts (CONT/LD) :

L'éditeur CONT (ou LD) affiche le programme sous forme de représentation graphique similaire aux schémas de câblage électriques. La programmation en CONT est la méthode choisie par de nombreux programmeurs d'automates programmables et par le personnel de maintenance. C'est un langage qui convient également très bien aux programmeurs débutants. Les programmes CONT permettent au programme d'émuler le flux du courant électrique partant d'une source de tension, à travers une série de conditions d'entrées logiques validant, à leur tour, des conditions de sortie logiques. Un programme CONT comprend une barre d'alimentation à gauche qui est alimentée en courant. Les contacts fermés permettent au courant de circuler à travers eux, vers l'élément suivant alors que les contacts ouverts bloquent le trajet du courant.

La logique est subdivisée en réseaux. Le programme est exécuté réseau par réseau, de la gauche vers la droite et de haut en bas comme indiqué par le programme. La figure VI.4 montre un exemple de programme CONT. [11]

Les différentes opérations sont représentées par des symboles graphiques de trois types fondamentaux :

- **Contacts :**

Les contacts représentent des commutateurs à travers lesquels le courant peut circuler. Le courant circule à travers un contact à fermeture uniquement si ce contact est fermé (1logique). Le courant circule à travers un contact à ouverture ou inversé (NOT) uniquement si ce contact est ouvert (0 logique).

- **Bobine :**

Les bobines représentent généralement des résultats de sorties logiques, tels que les lampes, les démarreurs de moteur, les relais intermédiaires et les conditions de sortie internes.

- **Boîtes :**

Les boîtes représentent des opérations supplémentaires, telles que temporisations, compteurs ou opérations arithmétiques. La figure IV.3 donne un exemple.

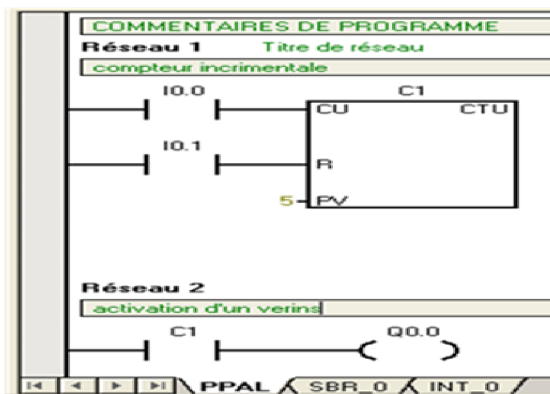


Figure IV.3 : Exemple de programme CONT.

IV.3 Organisation du programme utilisateur :

Le programme utilisateur pour une CPU S7-200 comporte trois types d'unités d'organisation de programme (UOP), et qui se présentent sous forme d'onglets distincts dans la fenêtre de l'éditeur de programme sur STEP7-Micro/WIN 32. [11]

- **Programme principal (OB1) :**

Le programme principal contient les opérations qui commanderont l'application, Le S7-200 exécute ces opérations séquentiellement, une fois par cycle.

- **Sous-programmes (SBRi) :**

Les sous programmes sont des éléments facultatifs du programme utilisateur qui sont exécutés uniquement lorsqu'ils sont appelés par le programme principal, par un programme d'interruption ou par un autre sous- programme. On écrit le code une fois dans un sous-programme puis on l'appelle autant de fois que nécessaire dans le programme principal.

- **Programmes d'interruption (INTi) :**

Les programmes d'interruption sont des éléments facultatifs du programme utilisateur qui réagissent à des événements d'interruption spécifiques. En effet chaque programme d'interruption est associé à un événement d'interruption. le S7-200 exécute les opérations écrites dans le programme d'interruption uniquement lorsque cet événement se produit.

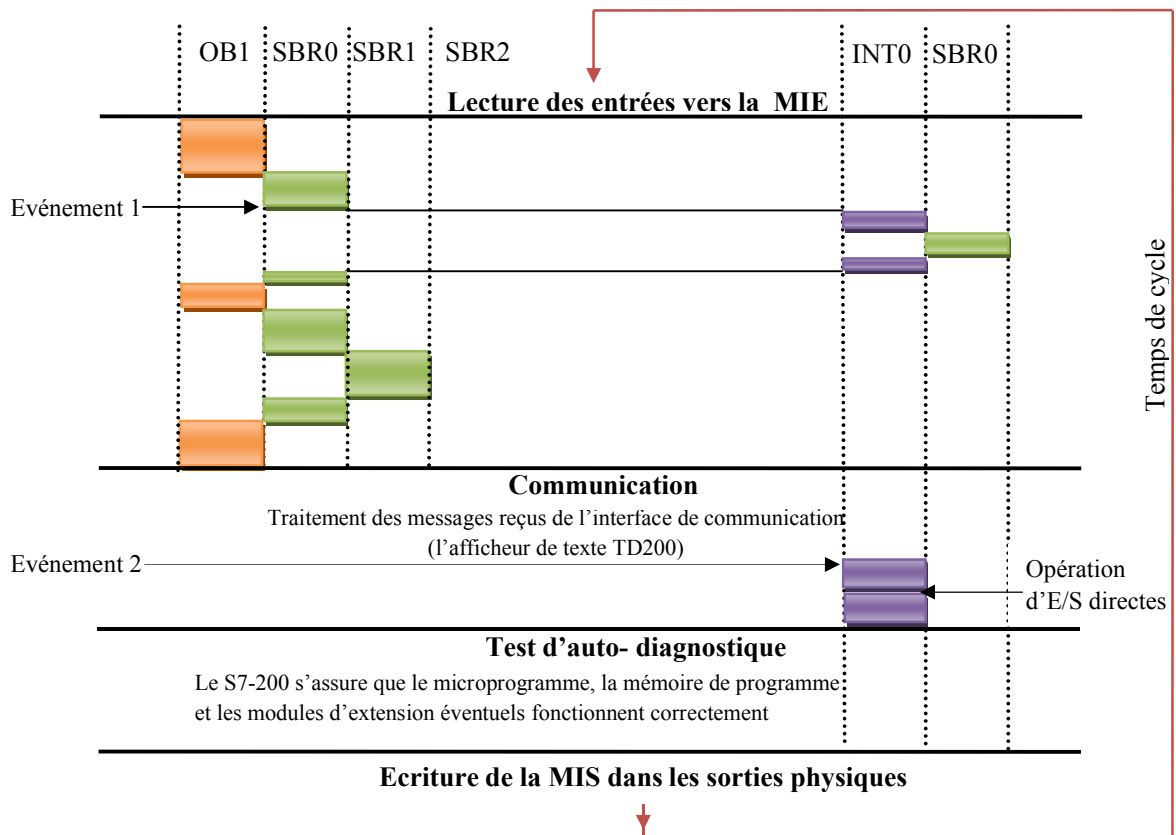


Figure IV.4 : Schéma de déroulement d'un cycle.

IV.4 Edition d'un programme sous STEP7-Micro/WIN32 :

IV.4.1 création d'un projet :

Un projet STEP7-Micro/WIN32 comporte les éléments suivants : [11]

- **Bloc de code :**



Bloc de code

Le bloc de code est composé de code exécutable et de commentaires. Le bloc exécutable comprend un programme principal (OB1), des sous-programmes (SBRi) et des programmes d'interruption facultatifs (INTj). Ce code est compilé et chargé ensuite dans la mémoire permanente de la CPU S7-200, ce qui n'est pas le cas des commentaires de programme.

- **Bloc de données (facultatif):**



Bloc de données

Le bloc de données est composé de données (valeurs de mémoire V initiales, valeurs des constantes) et de commentaires. Les données sont compilées et chargées dans la mémoire permanente de la CPU S7-200, ce qui n'est pas le cas des commentaires de programme.

- **Bloc de données système (facultatif) :**



Bloc de données systèmes

Le bloc de données système contient les informations de configuration, telles que les paramètres de communication, les zones de données rémanentes, les filtres d'entrées analogiques et TOR, et les valeurs de sorties en cas de passage à l'arrêt. Les informations du bloc de données sont chargées ensuite dans la mémoire permanente de la CPU S7-200.

- **Tables des mnémoniques :**



Tables des mnémoniques

La table des mnémoniques permet au programmeur d'utiliser l'adressage symbolique afin de suivre plus facilement la logique du programme, elle définit des mnémoniques accessibles par leur nom symbolique de n'importe quel endroit du programme. Le programme compilé, chargé dans la CPU S7-200 convertit tous les mnémoniques en adresse absolues. Par contre les informations de la table ne seront pas chargées dans la CPU.

- **Tables de visualisation d'état :**



Tables de visualisation d'état

La table de visualisation d'état permet de visualiser ou de forcer les valeurs des variables du processus lorsque le S7-200 exécute le programme de commande. On peut visualiser l'état des entrées, des sorties et des variables du programme en affichant leur valeur en cours. La table de visualisation d'état permet également de forcer les valeurs des variables du processus. Les informations des tables de visualisation d'état ne seront pas charger dans la CPU.


- **Références croisées :**



Références croisées

La fenêtre des références croisées contient des tableaux qui indiquent où les opérandes sont utilisés dans le programme et quelles zones de mémoire ont déjà été affectées (bits et octets utilisés). Pendant l'édition d'un programme à l'état "Marche" (RUN), les références croisées et les informations sur les éléments utilisés ne sont pas chargées dans l'API.

IV.4.2 Démarrage de STEP7-Micro/WIN 32 :

Il suffit d'un double clic sur l'icône STEP7-Micro/WIN 32 du bureau de Windows  ou bien en sélectionnant « SIMATIC > STEP7-Micro/WIN 32 » dans le menu « Démarrer » afin d'exécuter l'application. La figure VI.5 s'ouvre [11]

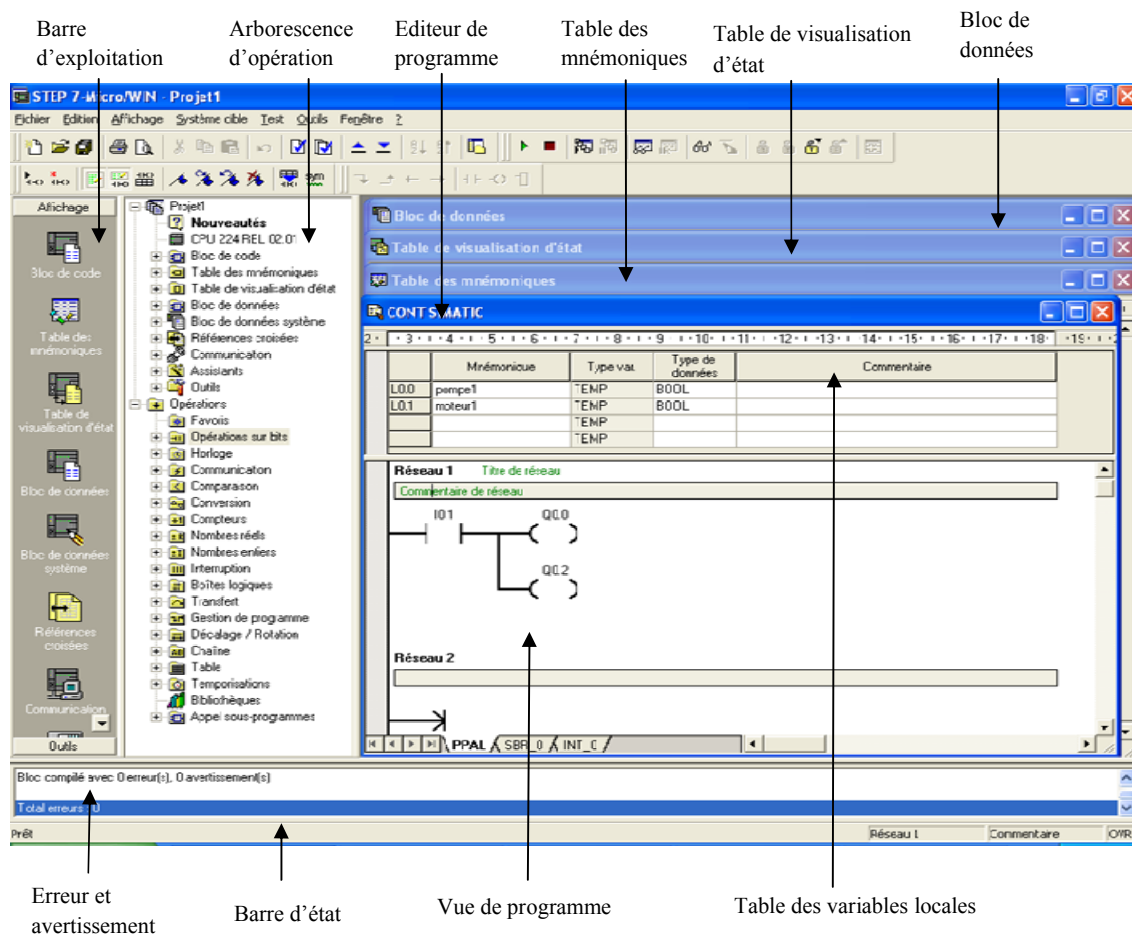


Figure IV.5 : Nouveau projet STEP7-Micro/WIN 32.

- **Les barres d'outils :**
Elles contiennent des boutons constituant des raccourcis pour les commandes de menus fréquemment utilisées. On peut afficher ou masquer n'importe quelle barre d'outils.
- **La barre d'exploitation :**
Elle présente des groupes d'icônes permettent d'accéder à différentes fonctions de programmation de STEP7-Micro/WIN 32.
- **Arborescence d'opération :**
Elle affiche tous les objets et les opérations nécessaires pour la création du programme de commande. On peut glisser et déplacer des opérations individuelles de l'arborescence d'opérations dans le programme ou bien double clic sur une opération afin de l'insérer à la position en cours du curseur dans l'éditeur de programme.
- **L'éditeur de programme :**
L'éditeur de programme contient la logique du programme et une table de variables locales dans laquelle on affecte des mnémoniques aux variables locales temporaires. Les sous-programmes (SBR) et les programmes d'interruption (INT) apparaissent sous forme d'onglets au bas de la fenêtre de l'éditeur de programme. Ces onglets permettent d'aller et venir entre sous-programmes, programmes d'interruption et programme principal.

IV.4.3 Edition du programme en langage CONT:

❖ saisie du programme :

- On Sélectionne les contacts, les bobines ou les boîtes dans la barre d'outils de l'éditeur de programme afin de saisir les équations sur les réseaux.
- On Enregistre les valeurs des paramètres '??.' avec les adresses correspondantes.

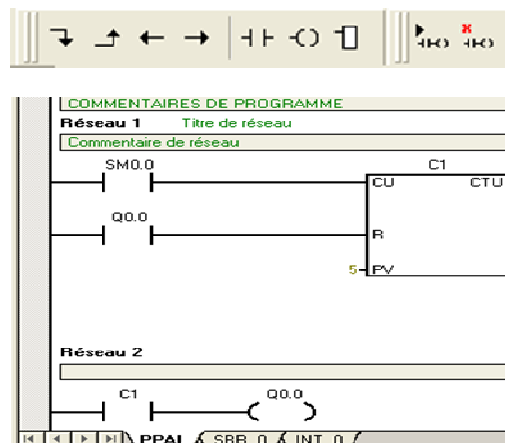


Figure IV.6 : Exemple de saisie d'un programme.

❖ **Compilation du programme :**

Une fois le programme est saisi, la compilation s'effectue à l'aide des boutons de la barre d'outils suivants :



- permet de compiler uniquement la fenêtre active (bloc de code ou bloc de données).




- Permet de compiler tous les composants du projet (bloc de code, bloc de données et bloc de données système).

❖ **Correction d'erreurs à l'aide de la fenêtre « Erreurs et avertissements » :**

Lors de la compilation, la fenêtre "Erreurs et avertissements" énumère toutes les erreurs qui apparaissent. Ces erreurs sont identifiées par leur emplacement (réseau, ligne, ou colonne). Un double clic sur une ligne d'erreur affiche le réseau contenant cette erreur dans l'éditeur de programme.

❖ **Enregistrement du programme :**

- Pour enregistrer le programme on clique sur le bouton  dans la barre d'outils, ou exécuter la commande enregistrer ou enregistrer sous du menu fichier, et on affecte ensuite un nom pour le projet.

IV.4.4 Établissement de la communication avec le S7-200:

L'icône **Communication** qui se trouve sur la barre d'exploitation ouvre la boîte de communication, présenté sur la figure VI.7, qui permet :

- La configuration des paramètres de communication :
 - L'adresse du câble PC/PPI multi-maître.
 - Interface pour le Paramètre de réseau (USB, COM1, COM2).
 - La vitesse de transmission (9.6, 19.2, 187.5 kbit/s).
- La connexion avec la CPU S7-200 :
 - Un double-clic sur l'icône de rafraîchissement (STEP7-Micro/WIN 32 fait recherche la station S7-200 connectée).
 - On sélectionne le S7-200 et on clique sur 'OK'.

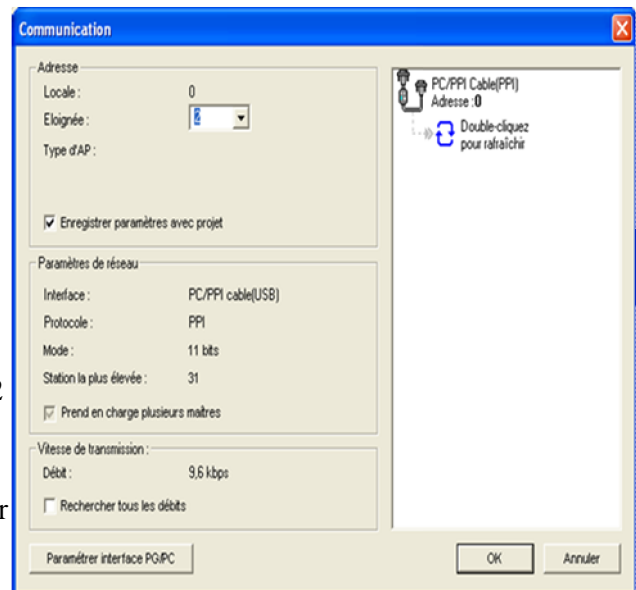




Figure IV.7 : Fenêtre de la boîte de communication.

IV.4.5 Chargement du programme dans la CPU :

- On met le S7-200 à l'état "Arrêt" avec le bouton marche dans la barre d'outils , ou par la sélection de la commande du menu **système cible »Arrêt**.

- On clique sur l'icône **Charger dans la CPU** dans la barre d'outils , ou on sélectionne la commande du menu **Fichier » Charger dans la CPU** afin de charger le programme dans la CPU. La fenêtre de chargement de la figure VI.8 s'ouvre.
- On sélectionne les éléments de programme à charger dans la CPU et on clique sur "OK" pour démarrer le chargement.

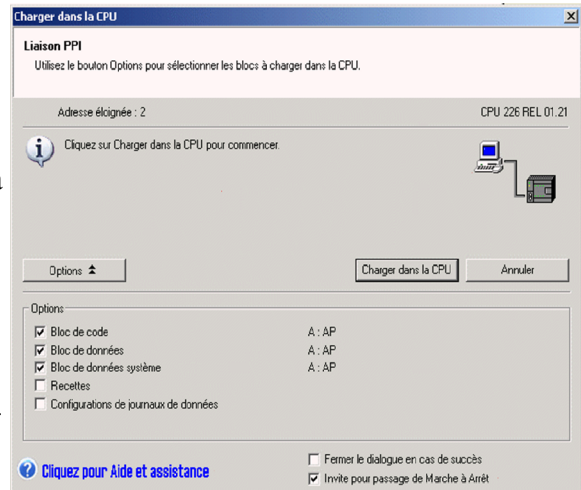


Figure IV.8 : Chargement du programme dans la CPU.

IV.4.6 Mise à l'état marche du S7-200 :

Une fois que le programme chargé dans la CPU et que le commutateur de mode du S7-200 se trouve sur la position TERME ou RUN, on fait passer le S7-200 à l'état "Marche" pour l'exécution du programme. Pour cela :


- On clique sur l'icône MARCHÉ  dans la barre d'outils ou on sélectionne la commande du menu **Système cible »Marche**. La fenêtre de la figure VI.9 s'affiche.
- On clique sur "Oui" pour changer l'état de fonctionnement du S7-200.



Figure IV.9 : passage du S7-200 à l'état marche.

IV.4.7 Surveillance du programme à l'aide de la table de visualisation d'état :

Pour accéder à la table de visualisation d'état, on sélectionne la commande « **Affichage Composante »Table de visualisation d'état** ou on clique sur l'icône **Table de visualisation d'état** dans la barre d'exploration.

Lorsqu'on crée une table de visualisation d'état, on entre les adresses des variables du processus à surveiller. On peut par exemple afficher une valeur de temporisation ou valeur d'un compteur en tant que bit ou en tant que mot. L'affichage de la valeur en tant que bit montre l'état du bit de temporisation ou de compteur alors que l'affichage de la valeur en tant que mot montre la valeur de temporisation ou de comptage. La figure VI.10 montre un exemple de visualisation d'état d'exécution dans un schéma CONT. [11]

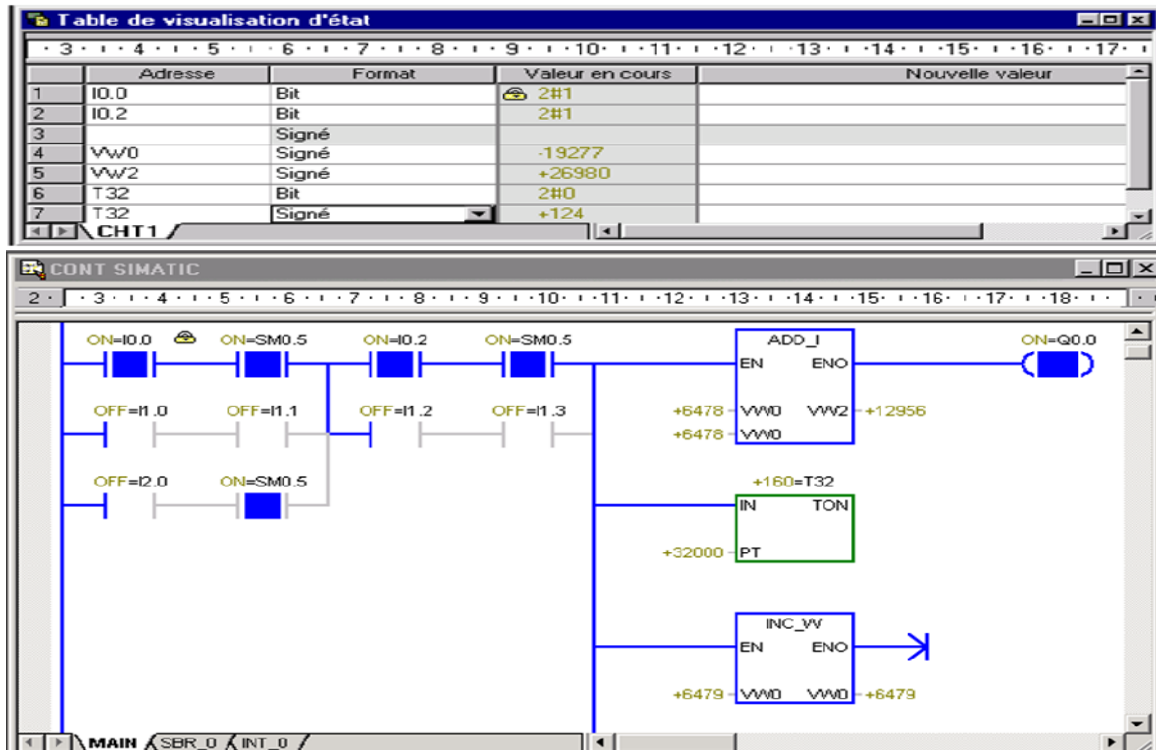


Figure IV.10 : Exemple de visualisation d'état d'exécution dans un schéma CONT.

IV.5 Les opérandes utilisés dans le programme :

Entrées TOR de l'API :

- I0.0** : Horloge de la phase A du compteur rapide HSC0/Mode 9 (sortie de l'encodeur principal).
- I0.1** : Horloge de la phase B du compteur rapide HSC0/Mode 9 (sortie de l'encodeur principale).
- I0.2** : L'adresse du bit qui signale le passage du spot qui est imprimé sur les feuilles du papier (sortie de la photocellule).
- I0.3** : L'adresse du bit qui signale le passage de la lame de coupe (sortie du capteur inductif).
- I1.2** : Sortie de l'encodeur de régulation.
- I1.3 = I0.2**

➤ Sorties TOR de l'API:

- Q0.0** : adresse du bit de sortie qui commande le relais du sens avant du moteur (la montée du rouleau de transfert).
- Q0.1** : adresse du bit de sortie qui commande le relais du sens arrière du moteur (la descente du rouleau de transfert).

➤ **Les mémentos spéciaux (SM) :**

SM0.0 : Ce bit est toujours à 1.

SM0.1 : Ce bit est à 1 au premier cycle uniquement. Il sert, entre autres, à l'appel d'un sous programme d'initialisation.

SMB37 : L'octet de commande du compteur rapide HSC0.

SMB38 : Le double mot, qui contient la valeur initiale en cours du compteur rapide HSC0.

SMD42 : Le double mot, qui contient la valeur prédéfinie du compteur rapide HSC0.

SM4.1 : Ce bit est à 1 lorsque la file d'attente des interruptions d'entrée a débordé.

➤ **Les touches du clavier de l'afficheur TD 200 :**

V961.0 : l'adresse du bouton poussoir F1 utilisé pour activer le mode de régulation manuelle et désactive le mode de régulation automatique.

V961.1 : l'adresse du bouton poussoir F2. Utilisé :

- En mode manuel pour commander la descente manuelle du rouleau de transfert (en activant Q0.1).

- En mode automatique comme entrée d'incréméntation du compteur CTUD (C30).

V961.2 : Bit de la mémoire V, activé avec le bouton poussoir F3 servant à l'activation de la régulation.

V961.3 : l'adresse du bouton poussoir F4 utilisé :

- En mode manuel pour commander la montée manuelle du rouleau de transfert (en activant Q0.0).

- En mode automatique comme entrée de décrémentation du compteur CTUD (C30).

V963.4 : l'adresse du bouton poussoir SHIFT_F1 utilisé pour activer le mode de régulation automatique et désactive le mode de régulation manuelle.

V963.5 : l'adresse du bouton poussoir SHIFT_F2 utilisé pour réinitialiser la table (TBL).

V963.6 : l'adresse du bouton poussoir SHIFT_F3 utilisé pour désactiver la régulation.

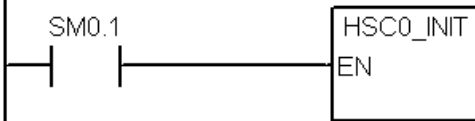
V963.7 : l'adresse du bouton poussoir SHIFT_F4 qui permet l'affichage de la position réelle du spot par rapport à la deuxième coupe d'une feuille et la remise à zéro du compteur CTUD (C30).

IV.6 Structure du programme :

LE PROGRAMME PRINCIPAL (OB1)

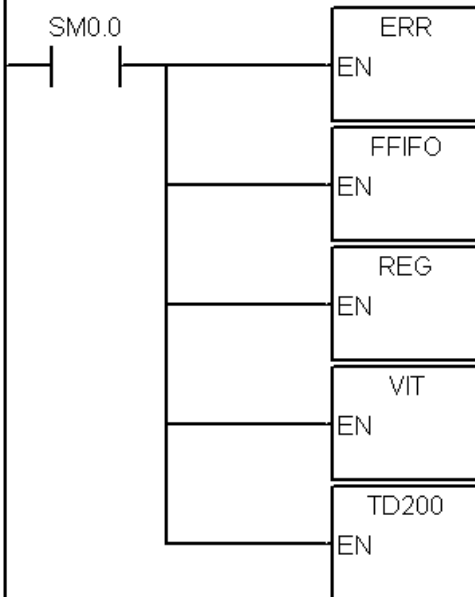
Réseau 1

Initialisation du compteur rapide HC0/mode9 uniquement au 1er cycle.



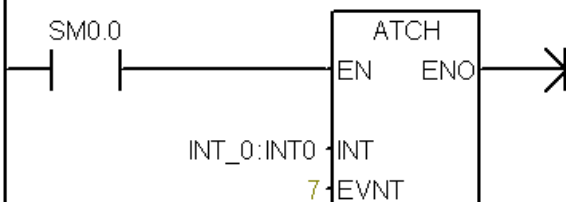
Réseau 2

Appel des sous programmes (ERR, REG, VIT, FFIFO, TD200).



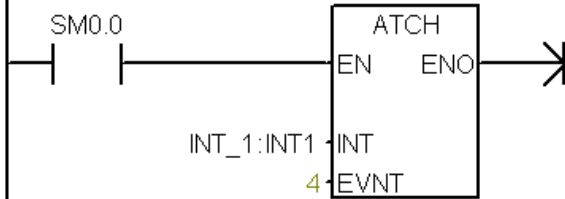
Réseau 3

A chaque front descendant de I0.3 - Capteur inductif - on associe un programme d'interruption INT0 qui permet l'initialisation du compteur rapide HC0.



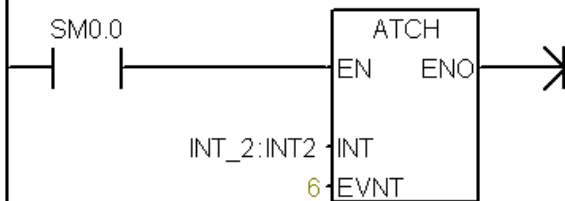
Réseau 4

A chaque front montant de I0.2 - photocellule - on associe un programme d'interruption INT1 qui permet la lecture de la position du spot par rapport à la 1^{ère} coupe d'une feuille.



Réseau 5

A chaque front montant de I0.3 - capteur inductif - on associe un programme d'interruption INT2 qui permet de calculer la distance entre le spot et la 2^{ème} coupe d'une feuille.



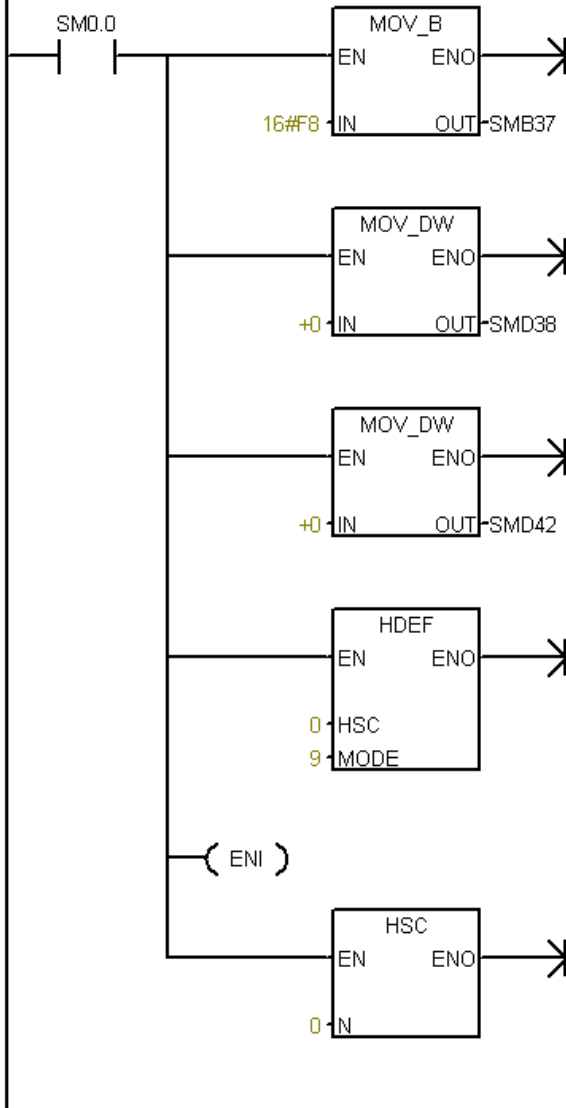
Sous programme d'initialisation du compteur rapide HCO,

Réseau 1

le sous-programme d'initialisation du compteur rapide HCO/mode9 utilise les entrées de l'encodeur principal suivantes: IO.0 et IO.1.

les étape de configuration du compteur HCO/mode9 :

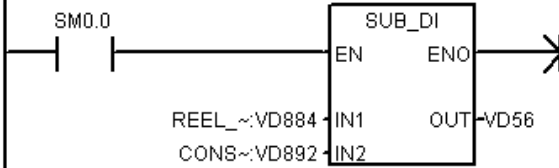
- 1- SMB37= 16#F8 cād ; - vitesse de comptage quadruple, valide le compteur, écrit une nouvelle valeur en cours, écrit une nouvelle valeur prédéfinie, définit (l'incréméntation) comme sens de comptage initial.
- 2- charger la valeur initiale en cours désirée- 0 pour l'effacer - dans SMD38 (double mot).(vc=0).
- 3- charger la valeur prédéfinie désirée dans SMD42 (VP=0)
- 4- exécute l'opération HDEF sans mise à 0 externe ni de démarrage.
- 6- valider tous les événements d'interruption avec l'opération ENI.
- 7- HSC permet de programmer le compteur HCO.



calcul du décalage.

Réseau 1

VD892: distance consigne (de référence) entre le spot et la 2^{ème} coupe d'une feuille en impulsions.
 VD 884: distance réelle entre le spot et la 2^{ème} coupe d'une feuille en impulsions.
 VD56 le décalage en terme d'impulsions.



Réseau 2

Conversion du double mot VD56 à un mot VW410 (conversion entière 32 bits à un entier de 16 bits).
 VW410 est le décalage qui sera ensuite rangé dans la table à chaque front montant de I0.2 (chaque détection du spot).

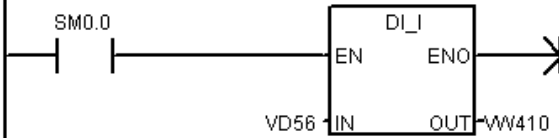


Table de données

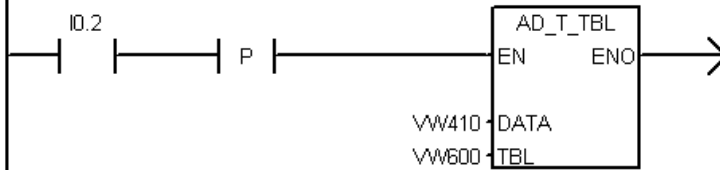
Réseau 1

Création d'une table de 8 données en commençant à partir de l'adresse VW604



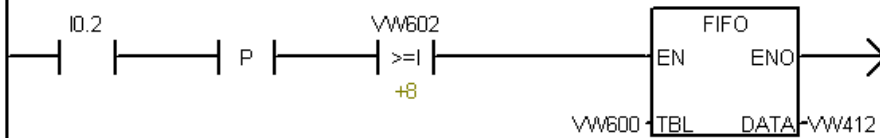
Réseau 2

pour chaque détection du spot (front montant de I0.2), le S7-200 enregistre les données issues du calcul de décalage dans la table.



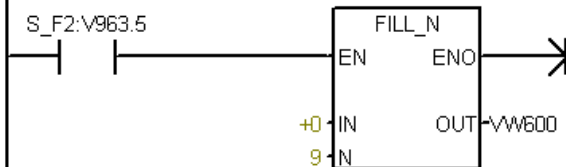
Réseau 3

- Au front montant de I0.2 si le décompte des entrées enregistrées est supérieur ou égal à 8, l'opération FIFO est activée (première entrée, première sortie).
- FIFO transfère la plus ancienne valeur d'une table (TBL=VW600), dans l'adresse de mémoire de sortie (DATA=VW412).



Réseau 4

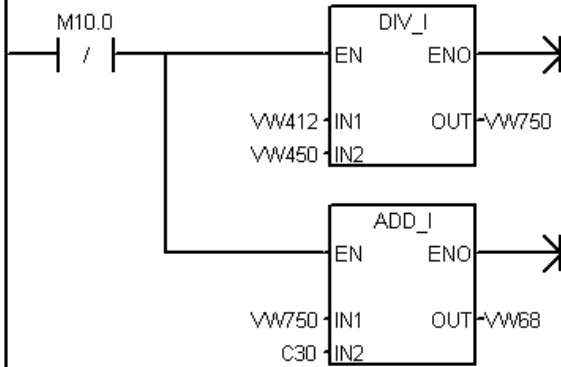
on peut initialiser la table via la touche du clavier SHIFT_F2.



sous programme de régulation

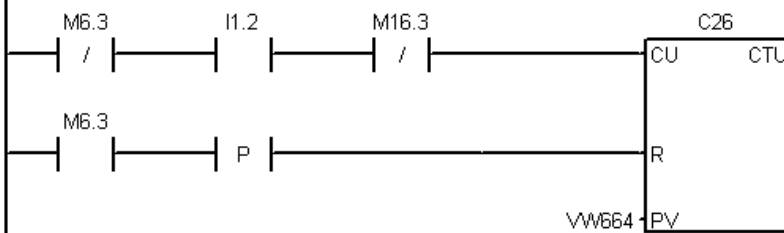
Réseau 1

correction du décalage..



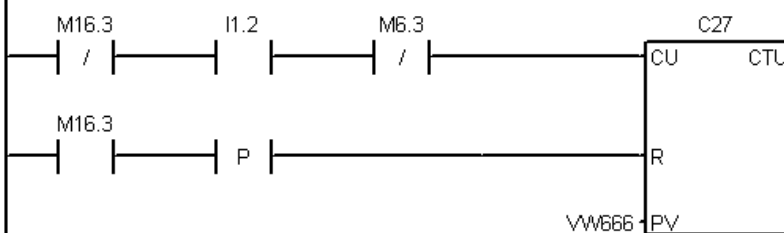
Réseau 2

comptage du glissement et reinitialisation à la montée du rouleau de transfért



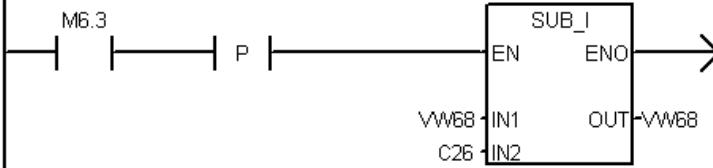
Réseau 3

comptage du glissement et reinitialisation à la descente du rouleau de transfért



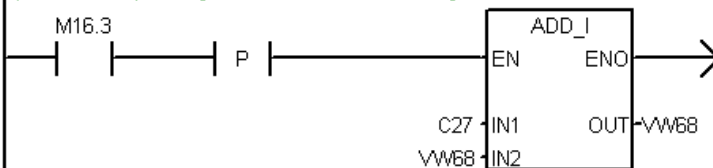
Réseau 4

prise en compte du glissement du moteur de régulation lors de la montée du rouleau de transfért



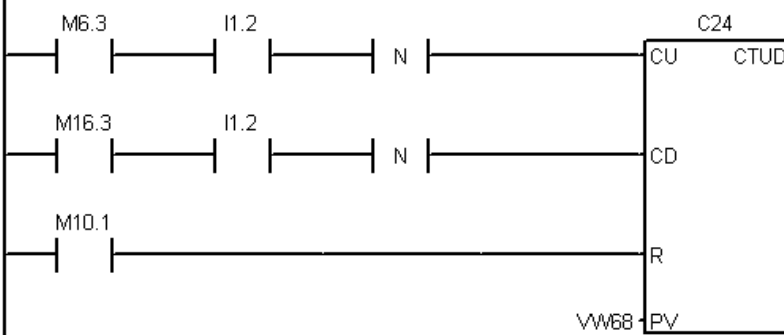
Réseau 5

prise en compte du glissement du moteur de régulation lors de la descente du rouleau de transfért



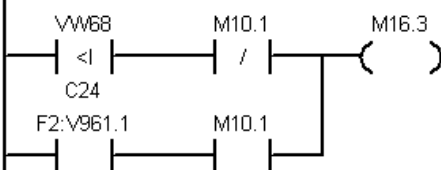
Réseau 6

commande du point d'arrêt du moteur de régulation



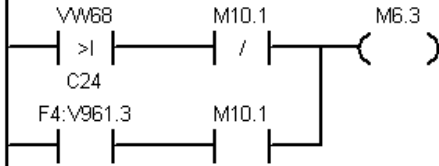
Réseau 7

mode automatique ou manuel d'activation de la descente du rouleau de transfért



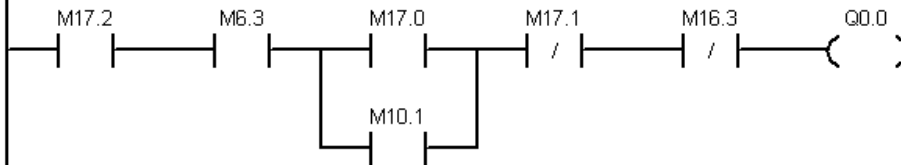
Réseau 8

mode automatique ou manuel d'activation de la montée du rouleau de transfért.



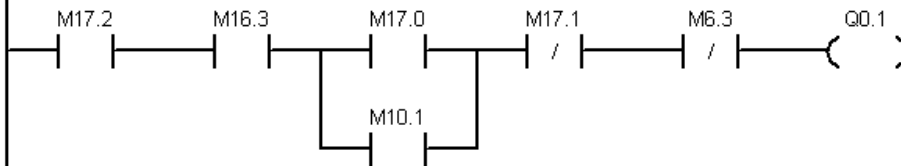
Réseau 9

-- Q0.0 : sortie physique de régulation qui commande le relais KA0 donnant l'ordre au moteur pour tourner dans le sens qui fait monter le rouleau de transfért.



Réseau 10

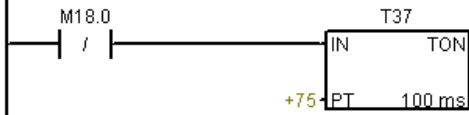
-- Q0.1 : sortie physique de régulation qui commande le relais KA1 donnant l'ordre au moteur pour tourner dans le sens qui fait descendre le rouleau de transfért.



sous programme qui calcule la vitesse moyenne du moteur principal en coupes/heure.

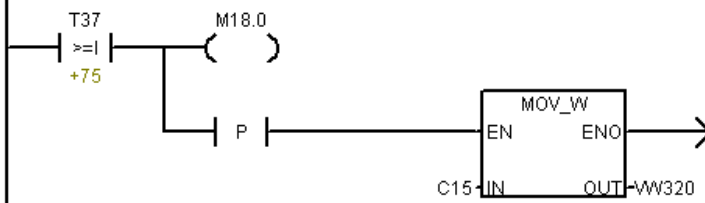
Réseau 1

Temporisation sous forme de retard à la montée, à redémarrage automatique.



Réseau 2

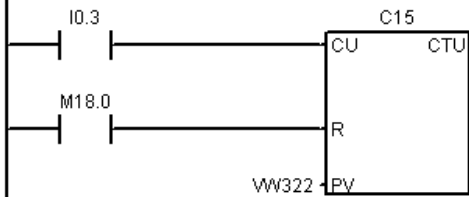
-- Lorsque le temps en cours atteint le temps predifini PT, l'entrée de validation sera désactivée, Ce qui efface la valeur en cours de la T37 (remise à 0), et permet en 1 cycle uniquement le transfert de la valeur en cours du compteur C15 dans VW320.



Réseau 3

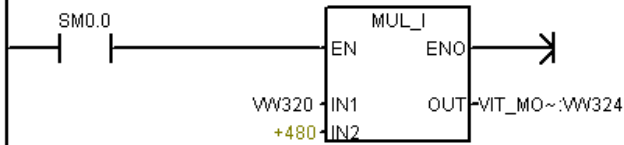
Comptage du nombre de coupes.

-- C15 : Compteur incremental qui calcule le nombre de coupes à chaque 7,5 seconde



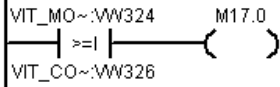
Réseau 4

VW324= vitesse moyenne du moteur en coupes par heure (C/H).



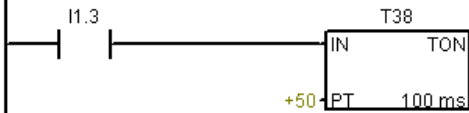
Réseau 5

La régulation ne commence que lorsque la vitesse moyenne de la lame de coupe dépasse une certaine vitesse (vitesse correcte) VW326 que l'opérateur imposera à partir de l'afficheur.

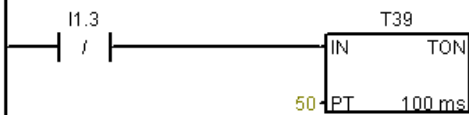


Réseau 6

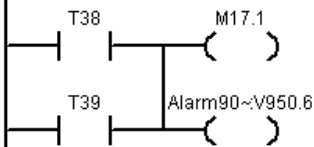
Lorsqu'il y a défaillance du capteur inductif la régulation sera désactivée et un message d'erreur s'affichera sur l'écran du TD200.



Réseau 7



Réseau 8



Aficheur du texte

Réseau 1

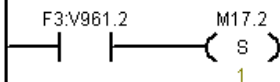
-- les fonctions des boutons du clavier afficheur TD200:

- F1: Active le mode de régulation manuelle.
- SHIFT+F1: Active le mode de régulation automatique.
- F2: : En commande manuelle utilisée pour la descente du rouleau de transfert en activant Q0.1.

En commande automatique utilisé comme entrée d'incrementation du compteur CTUD C30.

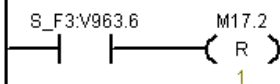
- SHIFT+F2: Initialisation de la table TBL..
- F3: : Bouton poussoir qui active la régulation.
- SHIFT+F3: : Bouton poussoir qui désactive la régulation.
- F4: : En commande manuelle utilisée pour la montée du rouleau de transfert en activant Q0.0.

- En commande automatique utilisée comme entrée de décrementation du compteur CTUD C30.
- SHIFT+F4: Permet l'affichage de la distance consigne VD892.
- Permet aussi la remise à 0 du compteur CTUD C30.



Réseau 2

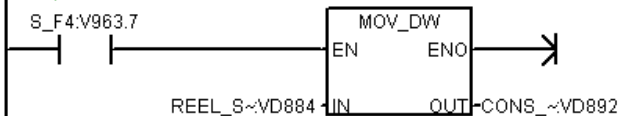
Désactiver la régulation.



Réseau 3

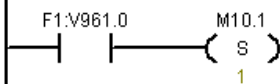
Affichage de la distance consigne sur l'ecran CONSIGNE SPOT.

cette fonction est exécutée au démarrage de l'impression uniquement afin de déterminer la distance consigne en terme d'impulsions.



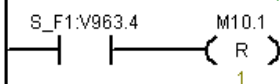
Réseau 4

Activer la commande manuelle.et désactiver la commande automatique de régulation.



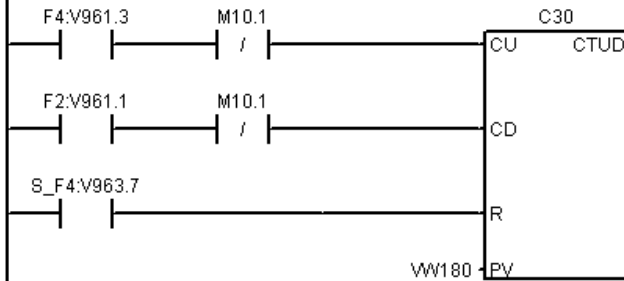
Réseau 5

Active la commande automatique et désactiver la commande manuelle de régulation.



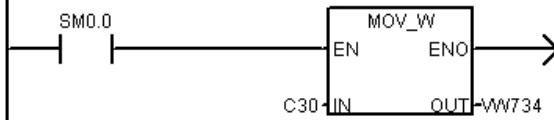
Réseau 6

C30 permet l'intervention sur la valeur de décalage (on l'a utilisée durant des essais)



Réseau 7

la valeur en cours du compteur C30 qui sera affichée sur l'écran du TD200.



Réseau 8

-- Lorsque le temps entre deux spots dépasse 2 secondes le S7-200 signale l'erreur du spot. et le message ERREUR SPOT s'affiche sur l'écran .



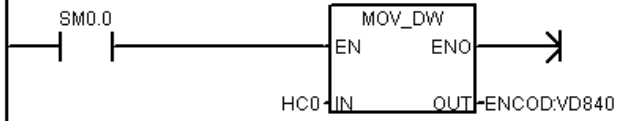
Réseau 9



Initialisation du compteur rapide HC0 à chaque front descendant de I0.3 -- capteur inductif -.

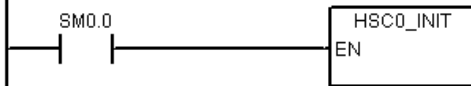
Réseau 1

Affichage de la résolution de l'encodeur principal.



Réseau 2

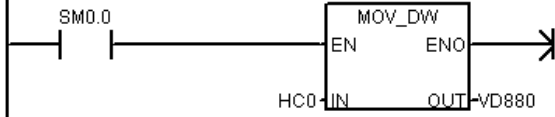
Appel au sous programme d'initialisation du compteur rapide HC0 .



lecture de la distance réelle entre le spot et la 1 ère coupe d'une feuille à chaque front montant de I0.2 (photocellule)

Réseau 1

lecture de la position du spot par rapport à la 1 ère coupe de la feuille.



Calcul de la distance réelle entre le spot et la 2^{ème} coupe d'une feuille.à chaque front montant de I0.3 (capteur inductif).

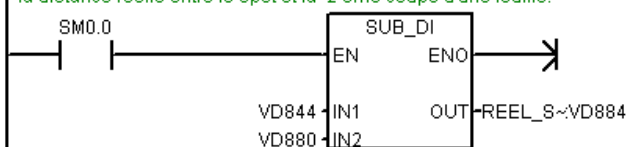
Réseau 1

Au front montant de I0.3 et s'il n'y a pas de débordement de la file d'attente des interruptions, alors le S7-200 charge la valeur en cours du compteur HC0 dans VD844 .



Réseau 2

la distance réelle entre le spot et la 2^{ème} coupe d'une feuille.



KOMORI - AURES EMBALAGE- / TD_CTRL_904 (SBR6)





Cette UOP a été générée par l'assistant Afficheur de texte comme partie de la configuration : Configuration TD 200 904.
L'opération TD_CTRL_X (Commande TD) sert à effacer toutes les validations d'alarmes. Il faut alors valider les alarmes individuelles selon les besoins.

KOMORI - AURES EMBALAGE- / TD_ALM_904 (SBR7)



Cette UOP a été générée par l'assistant Afficheur de texte comme partie de la configuration : Configuration TD 200 904.
L'opération TD_ALM_X (Valider alarme) sert à valider l'affichage d'une alarme.













KOMORI - AURES EMBALAGE- / USER1

|   Mnémonique | Adresse | Commentaire |
|--|---------|--|
| ENCOD | VD840 | la résolution de l'encodeur principal |
| REEL_SPOT | VD884 | la distance réelle entre le spot et la 2 ^{ème} coupe d'une feuille. |
| CONS_SPOT | VD892 | la distance consigne entre le spot et la 2 ^{ème} coupe d'une feuille. |
| VIT_MOY | VW324 | la vitesse moyenne de rotation de la lame de coupe. |
| VIT_CORECT | VW326 | la vitesse moyenne que la lame doit dépasser pour permettre la régulation. |

KOMORI - AURES EMBALAGE- / Mnémoniques d'UOP

| Mnémonique | Adresse | Commentaire |
|-------------|---------|---|
| HSC0_INIT | SBR0 | Sous programme d'initialisation du compteur rapide HCO, |
| ERR | SBR1 | calcul du décalage. |
| FFIFO | SBR2 | Table de données |
| REG | SBR3 | sous programme de régulation |
| VIT | SBR4 | sous programme qui calcule la vitesse moyenne du moteur principal en coupes/heure. |
| TD200 | SBR5 | Aficheur du texte |
| TD_CTRL_904 | SBR6 | Cette UOP a été générée par l'assistant Afficheur de texte comme partie de la configuration : Configuration TD 200 904. L'opération TD_CTRL_X (Commande TD) sert à effacer toutes les validations d'alarmes. Il faut alors valider les alarmes individuelles selon les besoins. |
| TD_ALM_904 | SBR7 | Cette UOP a été générée par l'assistant Afficheur de texte comme partie de la configuration : Configuration TD 200 904. L'opération TD_ALM_X (Valider alarme) sert à valider l'affichage d'une alarme. |
| INT_0 | INT0 | Initialisation du compteur rapide HCO à chaque front descendant de IO.3 -- capteur inductif -. |
| INT_1 | INT1 | lecture de la distance réelle entre le spot et la 1 ^{ère} coupe d'une feuille à chaque front montant de IO.2 (photocellule) |
| INT_2 | INT2 | Calcul de la distance réelle entre le spot et la 2 ^{ème} coupe d'une feuille. à chaque front montant de IO.3 (capteur inductif). |
| PPAL | OB1 | LE PROGRAMME PRINCIPAL (OB1) |

KOMORI - AURES EMBALAGE- / TD_SYM_904

|   Mnémonique | Adresse | Commentaire |
|--|---------|---|
| S_F4 | V963.7 | Mnémonique pour bouton de clavier 'SHIFT+F4' enfoncé (contact sans maintien) |
| F4 | V961.3 | Mnémonique pour bouton de clavier 'F4' enfoncé (contact sans maintien) |
| S_F3 | V963.6 | Mnémonique pour bouton de clavier 'SHIFT+F3' enfoncé (contact sans maintien) |
| F3 | V961.2 | Mnémonique pour bouton de clavier 'F3' enfoncé (contact sans maintien) |
| S_F2 | V963.5 | Mnémonique pour bouton de clavier 'SHIFT+F2' enfoncé (contact sans maintien) |
| F2 | V961.1 | Mnémonique pour bouton de clavier 'F2' enfoncé (contact sans maintien) |
| S_F1 | V963.4 | Mnémonique pour bouton de clavier 'SHIFT+F1' enfoncé (contact sans maintien) |
| F1 | V961.0 | Mnémonique pour bouton de clavier 'F1' enfoncé (contact sans maintien) |
|  TD_CurScreen_904 | VB967 | Ecran en cours affiché pour configuration TD 200 en VB904. Mettre à 16#FF si l'écran n'est pas affiché. |
|  TD_Left_Arrow_Key_904 | V960.4 | Mis à 1 lorsque l'utilisateur appuie sur la touche Flèche vers la gauche |
|  TD_Right_Arrow_Key_904 | V960.3 | Mis à 1 lorsque l'utilisateur appuie sur la touche Flèche vers la droite |
|  TD_Enter_904 | V960.2 | Mis à 1 lorsque l'utilisateur appuie sur la touche Entrée |
|  TD_Down_Arrow_Key_904 | V960.1 | Mis à 1 lorsque l'utilisateur appuie sur la touche Flèche vers le bas |
|  TD_Up_Arrow_Key_904 | V960.0 | Mis à 1 lorsque l'utilisateur appuie sur la touche Flèche vers le haut |
|  TD_Reset_904 | V949.0 | Si ce bit est mis à 1, le TD 200 relira sa configuration à partir de VB904. |
|  Data_1_2 | V975.1 | Notification d'édition pour VV326 |
|  Data_0_0 | V972.0 | Notification d'édition pour VV450 |
|  Data_0 | V971.0 | Notification d'édition pour VD892 |
| Alarm904_1 | V950.6 | Bit de validation pour alarme 1 |
| Alarm904_1_0 | V950.7 | Bit de validation pour alarme 0 |

```

//-----
//Cette page a été générée par l'assistantAfficheur de texte pour la configuration'ConfigurationTD 200 904'.
//Elle contient la configurationTD et tous les messagesutilisateur.
//-----
VV0 904 //
VB904 'td' //
VB906 22 //
VB907 1 //
VB908 0 //
VB909 16#0F //
VB910 0 //
VB911 16#04 //
VB912 0 //
VB913 0 //
VB914 1 //
VB915 2 //
VB916 4 //
VB917 6 //
VB918 0 //
VV919 0 //
VB921 2#1111 //
VB922 2#0 //
VB923 2#11110000 //
VB924 2#0 //
VB925 2#0 //
VB926 'Français' $03$00$02 //
//-----
//Bloc de validation d'alarme
//-----
VB949 1 //Commandede réinitialisationTD
VB950 0 //Bits de validation d'alarme
VB951 0 //Bits de validation d'alarme
VB952 0 //Bits de validation d'alarme
VB953 0 //Bits de validation d'alarme
VB954 0 //Bits de validation d'alarme
VB955 0 //Bits de validation d'alarme
VB956 0 //Bits de validation d'alarme
VB957 0 //Bits de validation d'alarme
VB958 0 //Bits de validation d'alarme
VB959 0 //Bits de validation d'alarme
VB960 0 //
VB961 0 //
VB962 0 //
VB963 0 //
VB964 0 //
VB965 0 //
VB966 0 //
TD_CurScreen_904vVB967 0 //
VB968 16#00 //
VB969 16#00 //
VB970 16#C0 //
VB971 16#80 //
VB972 16#00 //
VB973 16#40 //
VB974 16#C0 //
VB975 16#C0 //
//-----
//Messagesutilisateur
//-----
VB976 'ERREURSPOT' //
VB996 'ERREURCAPTEUR' //
VB1016 'REGISTRE BLAVA' $00 //
VB1037 'CONSIGNE' $01 //
VB1058 'COMP' $04 //
VB1079 'VITESSE' $05 //
VB1100 'SNC GM AURES EMBALAGE' //
VB1140 'CONS SPOT$03 $03/$00$03] REEL SPOT$00 $03]S F4' //
VB1180 'MULTIP $07$10$01A/$00$10$01A ERREUR..$00$10$00D' //
VB1220 'ENCODEUR$00$03H' //
VB1260 'DECALAGE + $00$10$02P' //
VB1300 'VITESSE $00$10$01DC/h VIT CORREC.$03$10$01F' //

```

Type de configuration d'assistant

Assistant Afficheur de texte

Nom de configuration d'assistant

Configuration TD 200 904

Date de création : 21.11.09 10:45:23

Dernière modification : 09.08.10 14:51:47

Éléments de projet générés par cette configuration d'assistant

- Sous-programme "TD_ALM_904"
- Sous-programme "TD_CTRL_904"
- Table de mnémoniques globaux "TD_SYM_904"
- Page de données "TD_DATA_904" pour la configuration TD à (VB904-VB1339)

Instructions d'utilisation

Le sous-programme "TD_CTRL_904" doit être appelé à chaque cycle de programme et n'être utilisé qu'une fois dans votre programme. Le sous-programme "TD_ALM_904" doit être appelé pour valider une alarme spécifique.

Options

| | |
|--|-------------------------|
| Modèle et version TD sélectionnés | TD 200 version 3.0 |
| Placer le décalage de bloc à \VWD ? | Oui |
| Valider le menu Heure du jour. | Oui |
| Valider le menu Forçage | Oui |
| Valider la protection par mot de passe | Oui |
| Mot de passe | 0000 |
| Fréquence d'actualisation | Aussi vite que possible |
| Afficher réaction sur l'écran TD 200 lorsqu'un bouton Mettre bit Activé à 1 est enfoncé. | |
| Longueur d'alarme | 20 octets (une ligne) |
| Les alarmes doivent avoir priorité sur les autres affichages. | Oui |

Dans quelle langue les invites et menus standard doivent-ils s'afficher ? Français
 Quel jeu de caractères désirez-vous utiliser pour les messages utilisateur ? Latin 1 (gras)

Alarmes

Cette configuration définit 2 alarmes.

Texte pour l'alarme 0

ERREUR SPOT

Options pour l'alarme 0

| | |
|--|--------------|
| Mnémonique pour cette alarme | Alarm904_1_0 |
| L'alarme nécessite l'acquittement de l'opérateur | Non |

Texte pour l'alarme 1

ERREUR CAPTEUR

Options pour l'alarme 1

| | |
|--|------------|
| Mnémonique pour cette alarme | Alarm904_1 |
| L'alarme nécessite l'acquittement de l'opérateur | Non |

Menu personnalisé

Cette configuration définit les choix de menu suivants :

| Index | Texte du menu |
|-------|----------------|
| (0) | REGISTRE BLAVA |
| (1) | CONSIGNE |
| (2) | COMP |
| (3) | VITESSE |

Ecrans pour le choix de menu 0 (REGISTRE BLAVA)

Texte pour l'écran 0

SNC GM
AURES EMBALAGE

Cet écran doit être l'affichage par défaut en l'absence d'interaction avec l'opérateur.

Ecrans pour le choix de menu 1 (CONSIGNE)

Texte pour l'écran 0

CONS SPOT <#0>/ <#2>
REEL SPOT <#1> S F4

Ce message utilisateur contient 3 références à des données AP.

- <#0> Adresse de données: VD892
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données autorisée
Les données peuvent automatiquement être éditées lorsque le message est validé.
Edition de données protégée par mot de passe.
Mnémonique pour le bit de notification d'édition pour ces données.: Data_0
- <#1> Adresse de données: VD884
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.
- <#2> Adresse de données: VD892
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.

Texte pour l'écran 1

MULTIP <#0>/ <#2>
ERREUR.. <#1>

Ce message utilisateur contient 3 références à des données AP.

- <#0> Adresse de données: VW450
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données autorisée
Les données peuvent automatiquement être éditées lorsque le message est validé.
Edition de données protégée par mot de passe.
Mnémonique pour le bit de notification d'édition pour ces données.: Data_0_0
- <#1> Adresse de données: VW68
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.
- <#2> Adresse de données: VW450
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.

Texte pour l'écran 2

ENCODEUR <#0>

Ce message utilisateur contient 1 références à des données AP.

<#0> Adresse de données: \VD840
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.

Ecrans pour le choix de menu 2 (COMP)

Texte pour l'écran 0

DECALAGE + <#0>

Ce message utilisateur contient 1 références à des données AP.

<#0> Adresse de données: \VW734
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.

Ecrans pour le choix de menu 3 (VITESSE)

Texte pour l'écran 0

VITESSE <#0> C/h
VIT CORREC. <#1>

Ce message utilisateur contient 2 références à des données AP.

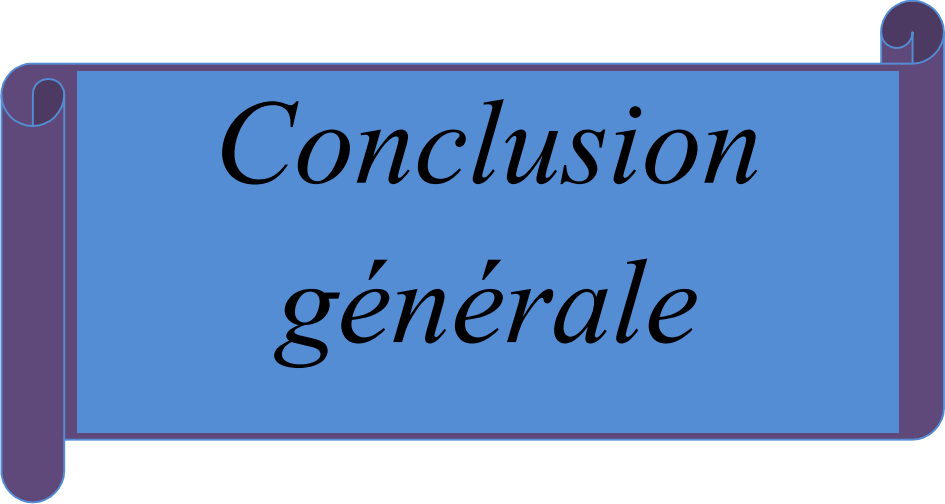
<#0> Adresse de données: \VW324
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données interdite.

<#1> Adresse de données: \VW326
Format des données: Signé
Chiffre/s à droite du séparateur décimal: 0
Edition de données autorisée
Les données peuvent automatiquement être éditées lorsque le message est validé.
Edition de données protégée par mot de passe.
Mnémonique pour le bit de notification d'édition pour ces données.: Data_1_2

Configuration de bouton de clavier

Il y a 8 boutons Mettre à 1 configurés

| Nom du bouton | Mnémonique | Action |
|---------------|------------|-----------------------|
| F1 | F1 | Contact sans maintien |
| SHIFT+F1 | S_F1 | Contact sans maintien |
| F2 | F2 | Contact sans maintien |
| SHIFT+F2 | S_F2 | Contact sans maintien |
| F3 | F3 | Contact sans maintien |
| SHIFT+F3 | S_F3 | Contact sans maintien |
| F4 | F4 | Contact sans maintien |
| SHIFT+F4 | S_F4 | Contact sans maintien |



*Conclusion
générale*

Conclusion générale :

L'évolution remarquable de la technologie pousse la créativité vers de meilleures méthodes de résolutions des problèmes. Des méthodes simples, efficaces et peu coûteuses dont le but est d'améliorer la production et augmenter la sécurité.

Le travail qui nous a été proposé nous a permis de passer beaucoup de temps sur le terrain et nous à donné l'occasion de visiter plusieurs sites où nous avons appris beaucoup de choses surtout sur l'imprimerie en général et sur les automates programmables en particulier; et cela grâce au contact des ingénieurs et des techniciens qui se sont montrés très professionnels hésitant pas à nous transmettre leur savoir, et nous aider dans notre travail. En plus de l'aspect pratique, ce travail consiste en l'étude de l'automate utilisé, et de la partie sur laquelle nous avons opéré.

Après quoi nous nous sommes intéressés plus particulièrement à la partie à commander. Nous avons alors mis en évidence tous les appareils qui interviennent dans le processus de l'impression, et qui nécessitent d'être contrôlés, comme la vitesse du défilement des feuilles et du moteur de coupe dans le bloc de découpage. Ce travail nous à permis aussi de découvrir le monde de l'impression et de définir l'API S7-200 ainsi que ses différents composants, de donner sa configuration et définir les caractéristiques du système (nombre d'entrées/sorties et capacité)...etc.

Le choix de l'automate programmable comme organe de commande nous a conduits à l'étudier. Nous avons découvert un outil performant et très compact qui permet de remplacer des commandes très complexes et surtout volumineuses. Cela nécessite néanmoins la maîtrise du langage spécifique à l'API utilisé (s7-200) qui est STEP7-micro /WIN.

Enfin, nous espérons avoir été à la hauteur des attentes et aussi que notre mémoire sera utile aux étudiants qui nous succéderont voulant travailler dans le domaine des API.



Annexe

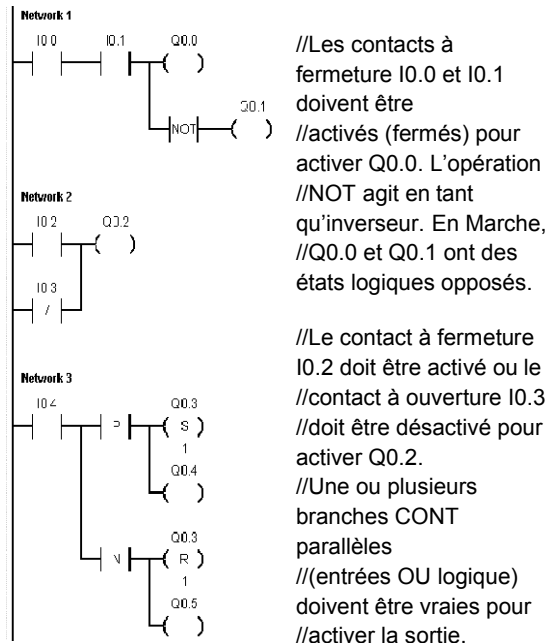
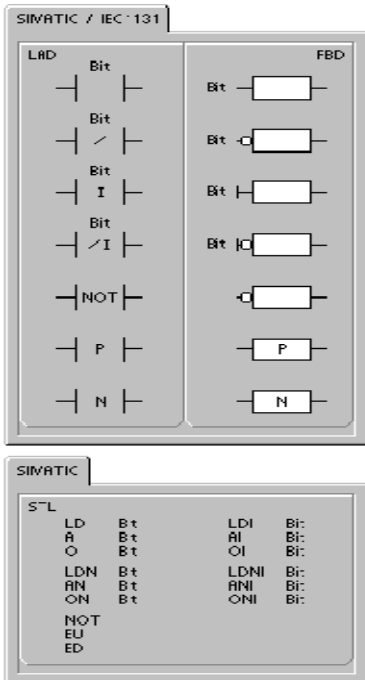
I Jeu d'opérations S7-200

I.1 Opérations combinatoires sur bits

I.1.1 Contacts

➤ **Contacts standards :**

Les opérations Contact à fermeture (LD, A, O) et Contact à ouverture (LDN, AN, ON) obtiennent la valeur référencée dans la mémoire image du processus, Les opérations Contacts standard obtiennent la valeur référencée dans la mémoire ou dans la mémoire image si le type de données est I ou Q. Le contact à fermeture est fermé (activé) lorsque le bit est égal à 1 et le contact à ouverture est fermé (activé) lorsque le bit est égal à 0. En LOG, on peut rajouter des entrées aux boîtes AND et OR à condition de ne pas dépasser 32 entrées au maximum. En LIST, l'opération LD charge la valeur du bit en haut de la pile et les opérations A et O combinent la valeur du bit à la valeur supérieure de la pile selon ET et OU respectivement. L'opération LDN charge la négation de la valeur du bit en haut de la pile et les opérations AN et ON combinent cette valeur inversée à la valeur supérieure de la pile selon ET et OU respectivement.



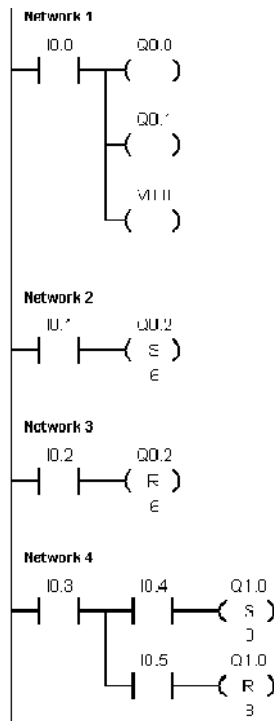
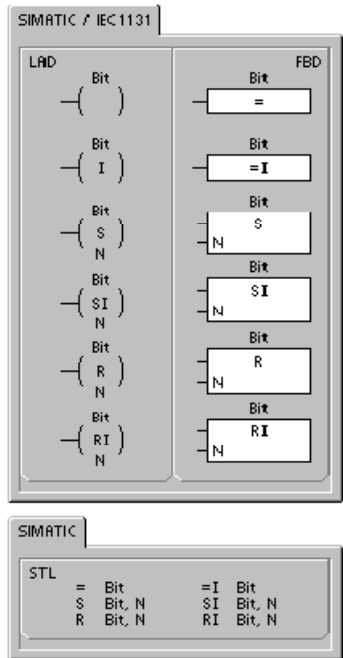
Exemple 1: contacts.

I.1.2 Bobine :

➤ **Sortie :**

L'opération Sortie (=) écrit la nouvelle valeur du bit de sortie dans la mémoire image. Lors de l'exécution de l'opération Sortie, le S7-200 met le bit de sortie dans la

mémoire image à 1 ou à 0. En CONT et en LOG, le bit indiqué est posé égal au flux de signal. En LIST, la valeur supérieure de la pile est copiée dans le bit indiqué.



//Les opérations de sortie affectent des valeurs de bit à des //E/S externes (I, Q) et à des adresses de mémoire interne //(M, SM, T, C, V, S, L).

//Mettre à 1 un groupe séquentiel de 6 bits. Indiquer une //adresse de bit de début et le nombre de bits à mettre à 1. //L'indicateur d'état de programme pour 1 est activé lorsque la //valeur du premier bit (Q0.2) est 1.

Exemple2 : Bobines.

I.2 Operations de comptages :

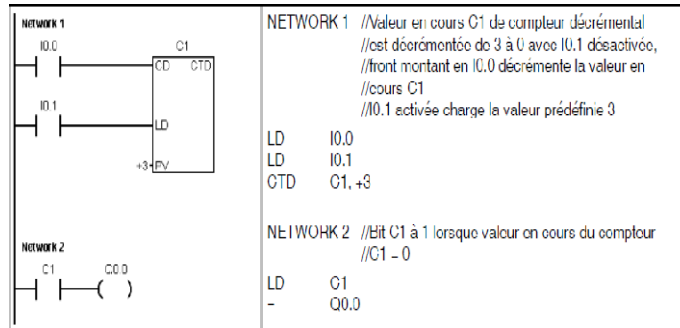
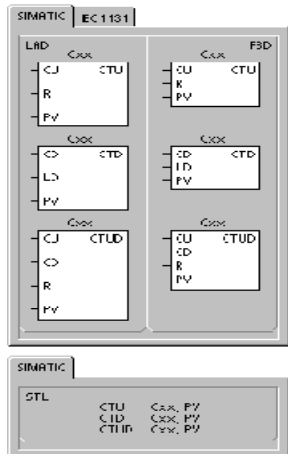
I.2.1 Compteur incrémental :

L'opération Compteur incrémental (CTU) incrémente en partant de la valeur en cours à chaque front montant de l'entrée d'incrémentation CU, Lorsque la valeur en cours "Cxxx" est supérieure ou égale à la valeur prédéfinie PV le bit de compteur Cxx est activé. Le compteur est remis à zéro lorsque l'entrée de remise à zéro R est activée ou que l'opération "Mettre à 0" est exécutée. Le compteur incrémental arrête le comptage lorsqu'il atteint la valeur maximale 32 767.

I.2.2 Compteur décrémental :

L'opération Compteur décrémental (CTD) décrémente en partant de la valeur en cours à chaque front montant de l'entrée de décrémentation CD. Lorsque la valeur en cours Cxx est égale à zéro, le bit de compteur Cxx est activé.

Le compteur remet le bit de compteur Cxx à 0 et charge la valeur prédéfinie PV dans la valeur en cours lorsque l'entrée de chargement LD est activée. Le compteur s'arrête lorsqu'il atteint zéro et le bit de compteur Cxx est alors mis à 1.



Exemple 3: compteur décremental.

I.3 Les compteurs rapides :

Tous les compteurs fonctionnent de la même manière dans un mode donné ; Il existe quatre types fondamentaux de compteurs : compteur monophasé avec gestion interne du sens de comptage, compteur monophasé avec gestion externe du sens de comptage, compteur biphasé avec deux entrées d'horloge et compteur en quadrature de phase A/B, tous les compteurs n'acceptent pas tous les modes on peut utiliser chaque type de compteur sans entrées de mise à zéro et de démarrage, avec entrée de mise à zéro mais sans entrée de démarrage, ou bien avec à la fois entrées de mise à zéro et de démarrage.

I.3.1 Définir mode pour compteur rapide :

L'opération Définir mode pour compteur rapide (HDEF) sélectionne le mode de fonctionnement (MODE) d'un compteur rapide spécifique (HSCx), la sélection du mode définit les fonctions d'horloge, de sens de comptage, de démarrage et de remise à zéro du compteur rapide, on peut utiliser une seule opération HDEF par compteur rapide.

I.3.2 Activer compteur rapide :

L'opération Activer compteur rapide (HSC) configure et commande le compteur rapide. Selon l'état des mementos spéciaux pour compteurs rapides le paramètre N précise le numéro du compteur rapide.

Les compteurs rapides servent typiquement d'entraînement pour dispositifs de comptage où un arbre en rotation à vitesse constante est muni d'un codeur angulaire incrémental, le codeur angulaire fournit un nombre défini de valeurs de comptage par tour et une impulsion de remise à zéro par tour. Les horloges et l'impulsion de remise à zéro du codeur angulaire constituent les entrées du compteur rapide.

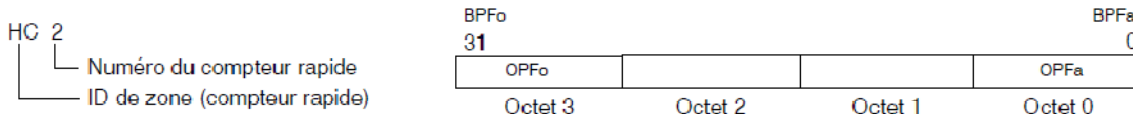
Il est possible de configurer les compteurs rapides pour douze modes de fonctionnement différents au maximum (voir tableau ci dessous).

| Mode | Description | Entrées | | | | |
|------|--|-------------------------|-------------------------|----------|-----------|------|
| | | HSC0 | I0.0 | I0.1 | I0.2 | |
| | | HSC1 | I0.6 | I0.7 | I1.0 | I1.1 |
| | | HSC2 | I1.2 | I1.3 | I1.4 | I1.5 |
| | | HSC3 | I0.1 | | | |
| | | HSC4 | I0.3 | I0.4 | I0.5 | |
| | | HSC5 | I0.4 | | | |
| 0 | Compteur monophasé avec gestion interne du sens de comptage | Horloge | | | | |
| 1 | | Horloge | | Mise à 0 | | |
| 2 | | Horloge | | Mise à 0 | Démarrage | |
| 3 | Compteur monophasé avec gestion externe du sens de comptage | Horloge | Sens de comptage | | | |
| 4 | | Horloge | Sens de comptage | Mise à 0 | | |
| 5 | | Horloge | Sens de comptage | Mise à 0 | Démarrage | |
| 6 | Compteur biphasé avec deux entrées d'horloge | Horloge, incrémentation | Horloge, décrémentation | | | |
| 7 | | Horloge, incrémentation | Horloge, décrémentation | Mise à 0 | | |
| 8 | | Horloge, incrémentation | Horloge, décrémentation | Mise à 0 | Démarrage | |
| 9 | Compteur en quadrature de phase A/B | Horloge phase A | Horloge phase B | | | |
| 10 | | Horloge phase A | Horloge phase B | Mise à 0 | | |
| 11 | | Horloge phase A | Horloge phase B | Mise à 0 | Démarrage | |
| 12 | Seuls HSC0 et HSC3 acceptent le mode 12. HSC0 compte le nombre d'impulsions sortant de Q0.0. HSC3 compte le nombre d'impulsions sortant de Q0.1. | | | | | |

Chaque compteur a des entrées réservées aux horloges, à la commande du sens de comptage, à la mise à zéro et au démarrage. Lorsque ces fonctions sont prises en charge les horloges des compteurs biphasés peuvent fonctionner toutes deux à leur fréquence maximale, En mode de quadrature de phase, on peut sélectionner des fréquences de comptage maximales simple (1x) ou quadruple (4x). Tous les compteurs fonctionnent à leur vitesse maximale sans interférer l'un avec l'autre.

I.3.3 Accès aux compteurs rapides (HC)

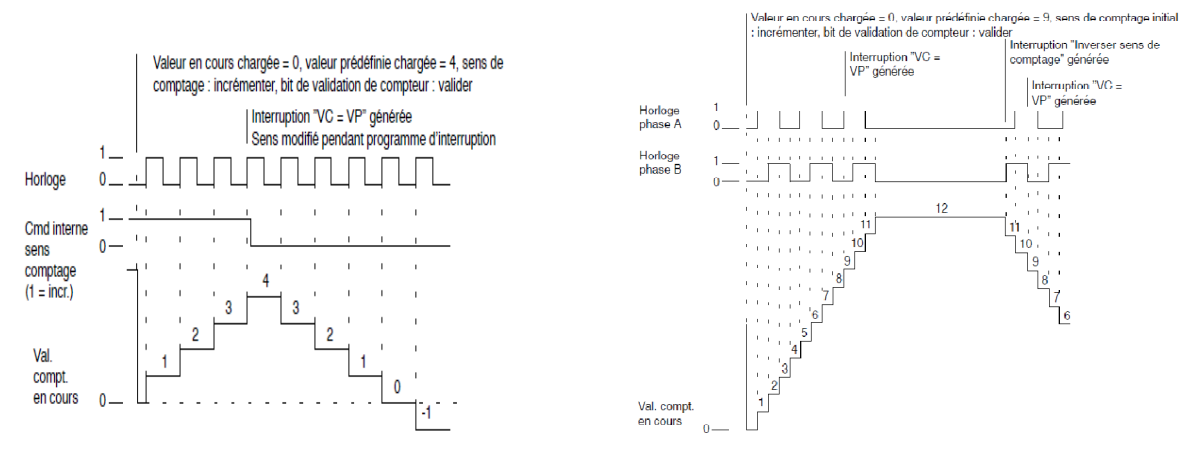
Pour accéder à la valeur de comptage d'un compteur rapide, on indique l'adresse du compteur rapide, comprenant l'identificateur de zone HC et le numéro du compteur (HC0, par exemple). La valeur en cours des compteurs rapides est une valeur en lecture seule à laquelle on peut accéder qu'en format de double mot (32 bits ; voir figure ci-dessous).



« Accès aux valeurs en cours des compteurs rapides ».

➤ Exemples de modes pour les compteurs rapides

Lorsque on utilise les modes de comptage 6, 7 ou 8 et qu'un front montant apparaît en moins de 0,3 microseconde d'intervalle à l'entrée d'incrémentation et à l'entrée de décrémentation, il peut arriver que le compteur rapide considère ces deux événements comme simultanés. La valeur en cours reste alors inchangée et le sens de comptage n'est pas inversé. En revanche, tant que les fronts montants apparaissant à l'entrée d'incrémentation et à l'entrée de décrémentation sont distants de plus de 0,3 microseconde, le compteur rapide les appréhende comme événements distincts. Aucune erreur n'est générée dans ces deux cas et le compteur conserve la valeur de comptage correcte.



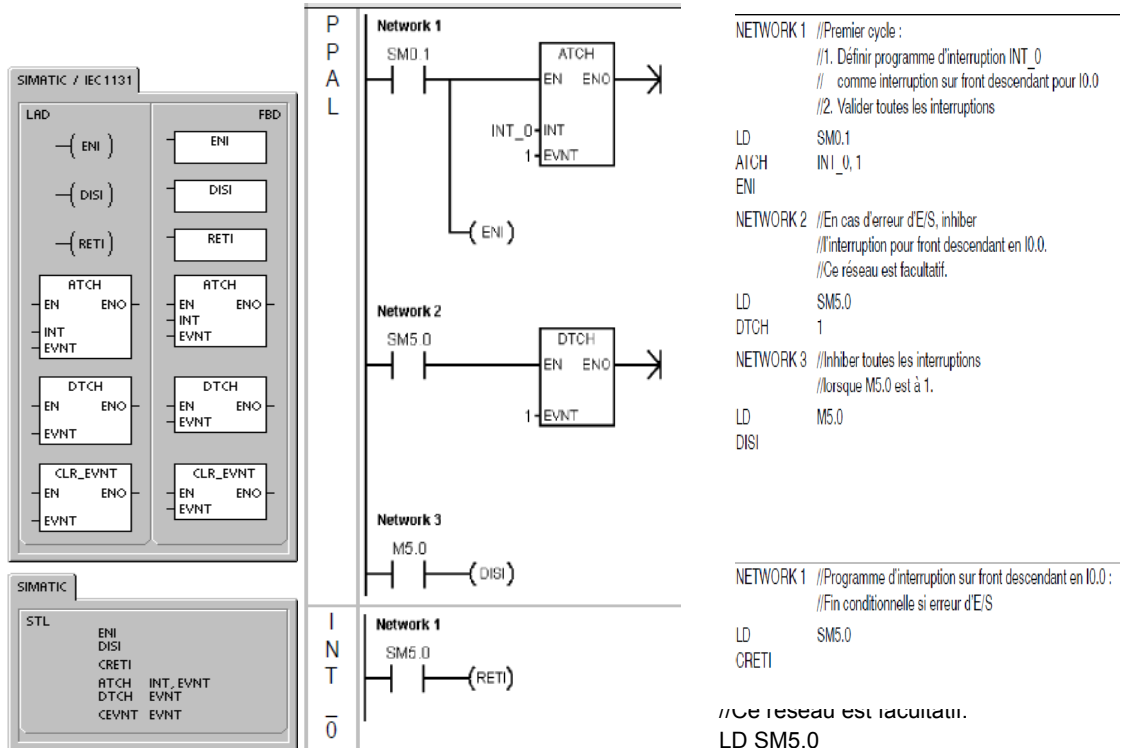
Fonctionnement en modes 0,1 ou 2.

Fonctionnement en modes 9,10 ou 11 (quadrature de phase).

I.4 Opérations d'interruption

I.4.1 Les événements d'interruptions :

L'opération valider tous les événements d'interruption (ENI) valide le traitement de tous les événements d'interruption et l'opération inhiber tous les événements d'interruption (DISI) inhibe le traitement de tous les événements d'interruption. Le passage à l'état de fonctionnement "Marche" (RUN) inhibe les interruptions. Une fois à l'état "Marche", on peut valider le traitement des interruptions à l'aide de l'opération ENI. L'exécution de l'opération DISI inhibe le traitement des interruptions ; toutefois, les événements d'interruption actifs continuent à être mis en attente.



Exemple 4: op

I.4.2 Affectation d'interruptions :

Tous les modes des compteurs acceptent un événement cours du compteur égale la valeur prédéfinie chargée. Les à zéro externe disposent également d'une interruption externe.

Tous les modes, excepté les modes 0, 1 et 2, acceptent de comptage est modifié. On peut valider ou inhiber ces conditions d'interruption séparément.

ir en
nise
zéro
d'interruption sur front descendant
en I0.0 :
s de
s de

I.4.3 Le traitement des programmes d'interruption par le S7-200

Le programme d'interruption s'exécute en réaction à un événement interne ou externe associé, la main est rendue au programme principal une fois la dernière opération du programme d'interruption exécutée. On peut quitter le programme d'interruption en exécutant une opération de fin conditionnelle du programme d'interruption.

I.4.4 Types d'interruptions prises en charge par le S7-200

Le S7-200 accepte les types suivants de programmes d'interruption :

- Interruptions de communication : Le S7-200 génère des événements permettant à un programme de gérer l'interface de communication.

- Interruptions d'E/S : Le S7-200 génère des événements pour différents changements d'état

de diverses E/S. Ces événements permettent à un programme de réagir aux compteurs rapides, aux sorties d'impulsions ou aux fronts montants ou descendants des entrées

➤ **Interruptions de communication :**

Notre programme peut commander le port de communication série du S7-200. Ce mode d'exploitation de cette interface est désigné par "communication programmable". Dans ce mode, notre programme définit le débit, le nombre de bits par caractère, la parité et le protocole. Les interruptions d'émission et de réception disponibles facilitent cette communication gérée par programme.

➤ **Interruptions d'E/S :**

Parmi les interruptions d'entrée/sortie, on compte les interruptions sur front montant ou descendant, les interruptions de compteur rapide et les interruptions de sortie de trains d'impulsions.

Le S7-200 peut générer une interruption sur front montant ou descendant d'une entrée (I0.0, I0.1, I0.2 ou I0.3), le système peut détecter les événements "Front montant" et "Front descendant" pour chacune de ces entrées. Ces événements permettent de signaler une situation exigeant une attention immédiate à l'apparition de l'événement.

Les interruptions de compteur rapide permettent de réagir à des situations telles que valeur en cours égale à valeur prédéfinie, modification du sens de comptage qui peut correspondre à une inversion du sens de rotation d'un arbre, ainsi que remise à zéro externe du compteur, chacun de ces événements de compteur rapide permet de réagir rapidement à des événements rapides que la vitesse de cycle de l'automate programmable ne permet pas de gérer.

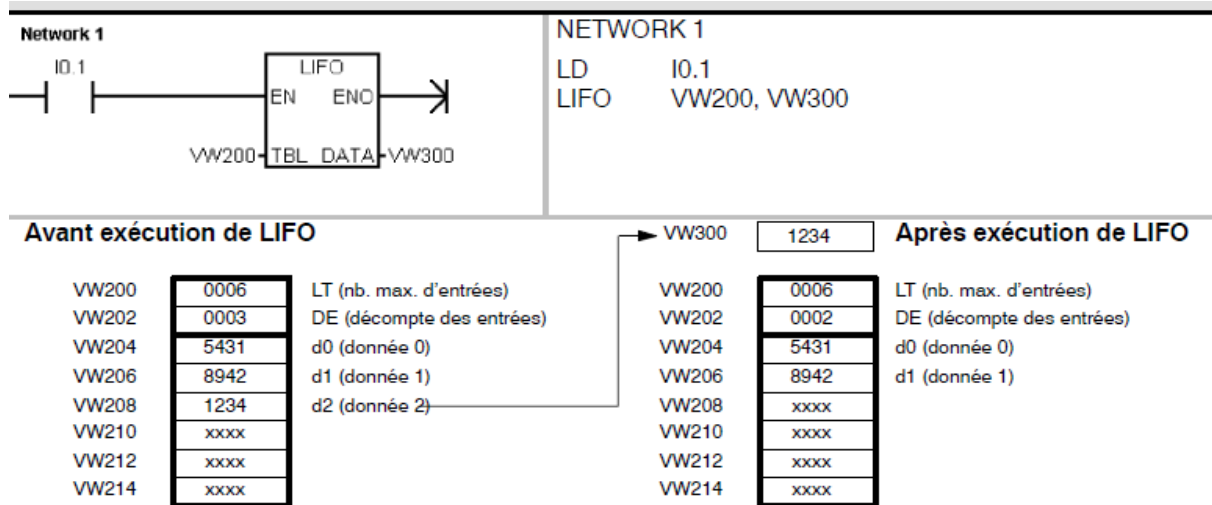
I.5 Premier entré, premier sorti et Dernier entré, premier sorti

I.5.1 Premier entré, premier sorti (first in, first out)

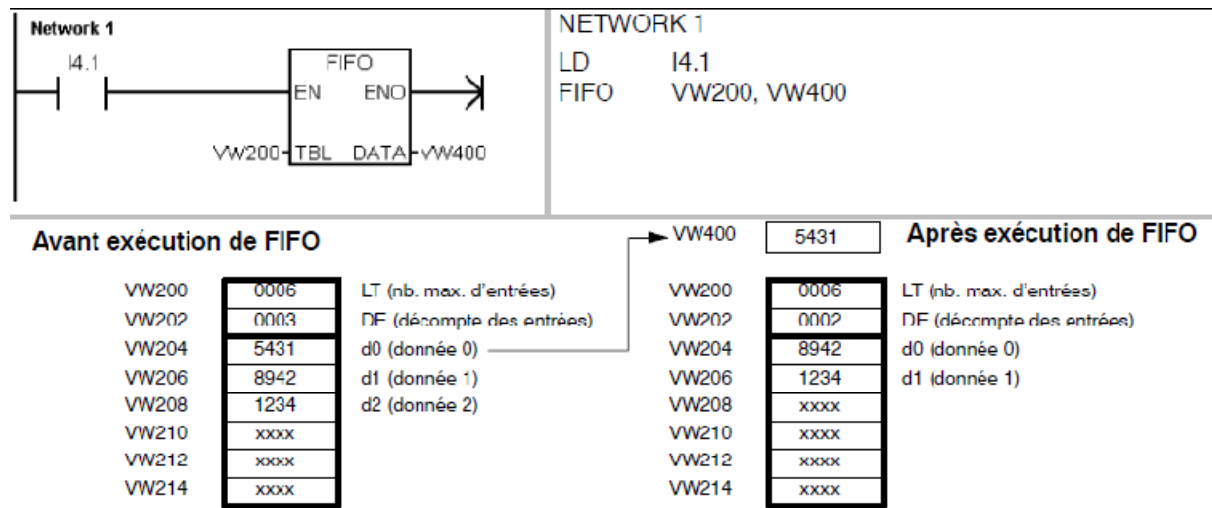
L'opération Premier entré, premier sorti (FIFO) transfère l'entrée la plus ancienne d'une table dans l'adresse de mémoire de sortie en extrayant la première entrée dans la table (TBL) et en l'écrivant dans l'adresse indiquée par DATA. Les entrées restant dans la table sont décalées d'une position vers le haut. Le décompte des entrées (DE) est décrémenté à chaque exécution de cette opération.

I.5.2 Dernier entré, premier sorti (last in, first out)

L'opération Dernier entré, premier sorti (LIFO) transfère l'entrée la plus récente d'une table dans l'adresse de mémoire de sortie en extrayant la dernière entrée dans la table (TBL) et en l'écrivant dans l'adresse indiquée par DATA. Le décompte des entrées (DE) est décrémenté à chaque exécution de cette opération.



Exemple 5 : Opération Dernier entré, premier sorti (LIFO).



Exemple 6: Opération Premier entré, premier sorti (FIFO).

I.6 Opérations arithmétiques

I.6.1 Opérations Additionner, Soustraire, Multiplier et Diviser

Additionner

$IN1 + IN2 = OUT$

Soustraire

$IN1 - IN2 = OUT$

Multiplier

$IN1 * IN2 = OUT$

Diviser

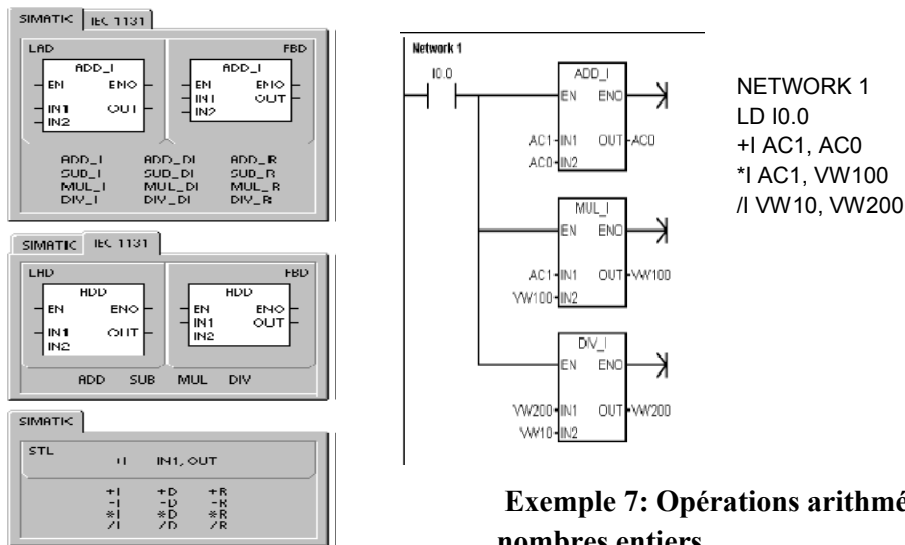
$IN1 / IN2$

$= OUT$

Les opérations Additionner entiers de 16 bits (+I) et Soustraire entiers de 16 bits (-I) additionnent et soustraient respectivement deux nombres entiers de 16 bits et fournissent un résultat de 16 bits. Les opérations Additionner entiers de 32 bits (+D) et Soustraire entiers de 32 bits (-D) additionnent et soustraient respectivement deux nombres entiers de 32 bits et fournissent un résultat de 32 bits. Les opérations Additionner nombres réels (+R) et

Soustraire nombres réels (-R) additionnent et soustraient respectivement deux nombres réels de 32 bits et fournissent un résultat réel de 32 bits.

Les opérations Multiplier entiers de 16 bits (*I) et Diviser entiers de 16 bits (/I) multiplient et divisent respectivement deux nombres entiers de 16 bits et fournissent un résultat de 16 bits. Pour la division, aucun reste n'est conservé. Les opérations Multiplier entiers de 32 bits (*D) et Diviser entiers de 32 bits (/D) multiplient et divisent respectivement deux nombres entiers de 32 bits et fournissent un résultat de 32 bits. Pour la division, aucun reste n'est conservé. Les opérations Multiplier réels (*R) et Diviser réels (/R) multiplient et divisent respectivement deux nombres réels de 32 bits et fournissent un résultat réel de 32 bits.



Exemple 7: Opérations arithmétiques sur nombres entiers.

I.7 Opérations de transfert

I.7.1 Transférer octet, mot, double mot ou réel

Les opérations Transférer octet (MOVB), Transférer mot (MOVW), Transférer double mot (MOVD) et Transférer réel (MOVR) transfèrent une valeur d'une adresse IN à une nouvelle adresse OUT sans modifier la valeur d'origine.

L'opération Transférer double mot permet de créer un Pointeur, Pour l'opération de transfert CEI, les types de données d'entrée et de sortie peuvent varier, mais doivent avoir la même taille.

I.8 Opérations de temporisation

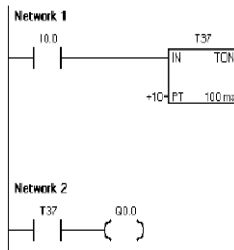
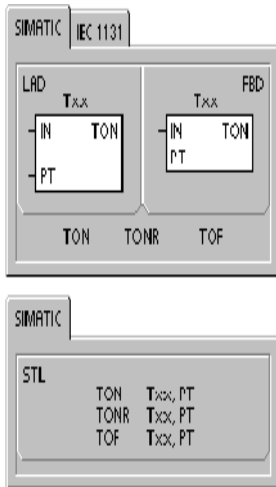
I.8.1 Démarrer temporisation sous forme de retard à la montée

Les opérations Démarrer temporisation sous forme de retard à la montée (TON) et Démarrer temporisation sous forme de retard à la montée mémorisé (TONR) comptent le temps qui s'écoule lorsque l'entrée de validation est activée.

Le numéro de la temporisation (Txx) détermine la résolution de la temporisation et la résolution est maintenant montrée dans la boîte de l'opération.

I.8.2 Temporisation sous forme de retard à la retombée

L'opération Démarrer temporisation sous forme de retard à la retombée (TOF) sert à retarder la désactivation d'une sortie pour un intervalle de temps donné après que l'entrée a été désactivée. Le numéro de la temporisation (Txx) détermine la résolution de la temporisation.



NETWORK 1 //Tempo
T37 à 100 ms expire
après
//(10 x 100 ms = 1s)
//Front montant en
I0.0=T37 validée,
//Front descendant en
I0.0=désactivation et
//remise à 0 de T37
LD I0.0
TON T37, +10
NETWORK 2 //Le bit T37
est géré par la
temporisation T37

Exemple 8: Temporisation sous forme de retard à la montée.

➤ Détermination de la résolution de la temporisation

Les temporisations comptent des intervalles de temps. La résolution (ou période) de la temporisation détermine la durée dans chaque intervalle. Par exemple, une temporisation TON ayant une résolution de 10 ms compte le nombre d'intervalles de 10 ms qui s'écoulent, une fois la temporisation TON validée : une valeur de comptage de 50 pour une temporisation de 10 ms correspond à 500 ms. Les temporisations SIMATIC sont disponibles avec trois résolutions : 1 ms, 10 ms et 100 ms. La résolution est déterminée par le numéro de la temporisation, comme illustré dans le tableau suivant :

| Type de temporisation | Résolution | Valeur maximale | Numéro de la temporisation |
|---------------------------|------------|----------------------|----------------------------|
| TONR (rémanente) | 1 ms | 32,767 s (0,546 min) | T0, T64 |
| | 10 ms | 327,67 s (5,46 min) | T1 à T4, T65 à T68 |
| | 100 ms | 3276,7 s (54,6 min) | T5 à T31, T69 à T95 |
| TON, TOF (non rémanentes) | 1 ms | 32,767 s (0,546 min) | T32, T96 |
| | 10 ms | 327,67 s (5,46 min) | T33 à T36, T97 à T100 |
| | 100 ms | 3276,7 s (54,6 min) | T37 à T63, T101 à T255 |

I.9 Quelques mémentos spéciaux

Les mémentos spéciaux (SM) offrent diverses fonctions d'état et de commande et servent également pour l'échange d'informations entre le S7-200 et le programme. Il peut s'agir de bits, d'octets, de mots ou de doubles mots.

| Bits SM | Description (lecture seule) |
|---------|---|
| SM0.0 | Ce bit est toujours à 1. |
| SM0.1 | Ce bit est à 1 au premier cycle. Il sert, entre autres, à l'appel d'un sous-programme d'initialisation. |
| SM0.2 | Ce bit est mis à 1 pour la durée d'un cycle si des données rémanentes ont été perdues. Vous pouvez l'utiliser comme memento d'erreur ou pour appeler une séquence de mise en route particulière. |
| SM0.3 | Ce bit est mis à 1 pour la durée d'un cycle si une mise sous tension entraîne le passage à l'état "Marche" (RUN). Il permet, par exemple, de fournir un temps de chauffe de l'installation avant de commencer l'exploitation. |
| SM0.4 | Ce bit fournit une impulsion d'horloge en fonction pendant 30 secondes et hors fonction pendant 30 secondes, pour une période d'une minute. Vous disposez ainsi d'un retard d'emploi simple ou d'une impulsion d'horloge d'une minute. |
| SM0.5 | Ce bit fournit une impulsion d'horloge en fonction pendant 0,5 seconde et hors fonction pendant 0,5 seconde, pour une période d'une seconde. Vous disposez ainsi d'un retard d'emploi simple ou d'une impulsion d'horloge d'une seconde. |
| SM0.6 | Ce bit est une horloge de cycle en fonction pendant un cycle et hors fonction pendant le cycle suivant. Vous pouvez l'utiliser comme entrée de comptage d'un cycle. |
| SM0.7 | Ce bit indique la position du commutateur de mode (0 correspondant à la position TERM et 1 à la position RUN). Si vous validez la communication programmable à l'aide de ce bit lorsque le commutateur de mode est sur RUN, vous pouvez valider la communication normale avec la console de programmation en mettant le commutateur en position TERM. |

« Octet de memento spécial SMB0 (SM0.0 à SM0.7) ».

I.10 Les assistant pour opération :

I.10.1 Assistant pour opération HSC :

Le S7-200 met à notre disposition l'opération HSC (Activer compteur rapide) en CONT, LOG et LIST pour gérer les événements qui se produisant à une fréquence plus rapide que le temps de cycle de la CPU. Le fonctionnement d'un compteur rapide est très souple, avec de nombreuses options d'initialisation déterminant le comportement du compteur. Pour configurer correctement un compteur rapide, il faut définir plusieurs paramètres en mémoire SM, configurer un compteur spécifique pour chaque mode de fonctionnement, associer des programmes d'interruption aux différents événements de comptage, puis valider le fonctionnement du compteur.


I.10.2 Les étapes d'initialisation d'un compteur rapide :

La procédure suivante décrit comment initialiser HSC0 comme compteur en quadrature de phase A/B en mode 9 (sans entrées de mise à zéro et de démarrage).

1. A l'aide du mémento "Premier cycle SM0.1", on fait appelle à un sous-programme où l'initialisation est exécutée. L'appel d'un sous-programme permet de mieux structurer le programme et, comme il n'est pas exécuté lors des cycles suivants, le temps de cycle n'est pas allongé.
2. Dans le sous-programme d'initialisation, on charge dans SMB37 les valeurs correspondant au fonctionnement désiré.
Exemple (vitesse de comptage simple) : SMB37 = 16#FC :
Cet octet de commande:
 - valide le compteur,
 - écrit une nouvelle valeur en cours.
écrit une nouvelle valeur prédéfinie.
 - définit l'incréméntation comme sens de comptage initial.Exemple (vitesse de comptage quadruple) : SMB47 = 16#F8
Cet octet de commande
 - valide le compteur,
 - écrit une nouvelle valeur en cours.
 - écrit une nouvelle valeur prédéfinie.
 - définit l'incréméntation comme sens de comptage initial.
3. On exécute l'opération HDEF avec l'entrée HSC à 0 et l'entrée MODE définie à la valeur 9 pour pas de mise à zéro externe ni de démarrage.
4. On charge la valeur en cours désirée - 0 pour l'effacer - dans SMD38 (double mot).
5. On charge la valeur prédéfinie désirée dans SMD42 (double mot).
6. Afin de détecter si la valeur en cours est égale à la valeur prédéfinie, on programme une interruption en associant l'événement d'interruption 13 (Valeur en cours = valeur prédéfinie) à un programme d'interruption.
7. Afin de détecter si le sens de comptage change, on peut programmer une interruption en associant l'événement d'interruption 14 (Sens de comptage modifié) à un programme d'interruption.
8. Afin de détecter une mise à zéro externe, programmer une interruption en associant l'événement d'interruption 15 (Mise à zéro externe) à un programme d'interruption.
9. on exécute l'opération ENI (Valider tous les événements d'interruption) afin de valider les interruptions.
10. On exécute l'opération HSC afin que le S7-200 programme le compteur HSC0.

En outre STEP 7-Micro/WIN fournit les assistants (ou outils de configuration) pour faciliter et automatiser la programmation. L'assistant d'opération HSC nous aide à choisir entre les différents modes et événements d'interruption en fonction des besoins du processus dans les applications nécessitant un compteur rapide.

I.10.3 Utilisation de l'assistant HSC :

L'assistant HSC nous facilite l'initialisation d'un compteur rapide en suivant les mêmes étapes précédentes, il utilise les informations suivantes : type et mode de compteur, valeur prédéfinie du compteur, valeur en cours du compteur et sens de comptage initial. Pour lancer l'assistant HSC, on sélectionne la commande Outils > Assistant pour opérations... puis on sélectionne HSC, ou on clique directement sur l'icône Assistants pour opérations dans la barre d'exploration  puis on sélectionne HSC ou on ouvre le dossier Assistants dans l'arborescence d'opérations puis on sélectionne l'assistant HSC.

I.11 Assistant afficheur de texte :

STEP 7-Micro/WIN dispose d'un assistant grâce auquel il est aisé de configurer le bloc de paramètres ainsi que les messages dans la zone de mémoire de la CPU S7-200. L'assistant de configuration TD 200 inscrit automatiquement le bloc de paramètres ainsi que les textes de message dans l'éditeur de blocs de données, après avoir sélectionné les options et créé les messages. Ce bloc de données peut ensuite être chargé dans la CPU.

I.11.1 Les étapes de création d'un bloc de code :

1. Démarrage de l'assistant TD 200 :
Pour lancer l'assistant Afficheur de texte, on sélectionne la commande Outils » Assistant Afficheur de texte, ou on clique sur l'icône Assistant Afficheur de texte dans la barre d'exploration Ou on ouvre le dossier Assistants dans l'arborescence d'opérations puis on sélectionne 'assistant'. La boîte de dialogue sur la Figure s'affiche.



Figure.1 : démarrage de l'assistant TD200.

2. Sélection du modèle et de la version TD :
On cliquant sur suivant, la boîte de dialogue pour le choix du modèle et version de l'afficheur s'affiche sur la figure . Les différents modèles et versions de TD prennent en charge des fonctions différentes, pour notre cas on sélectionne l'afficheur TD 200 version 3.0.

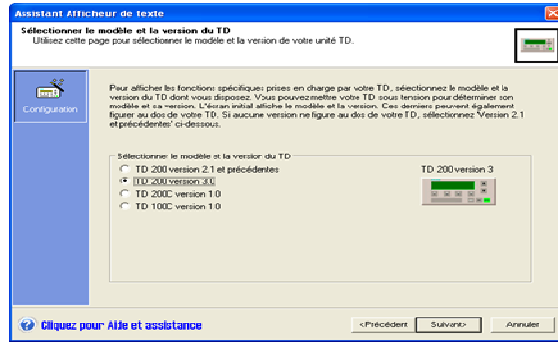
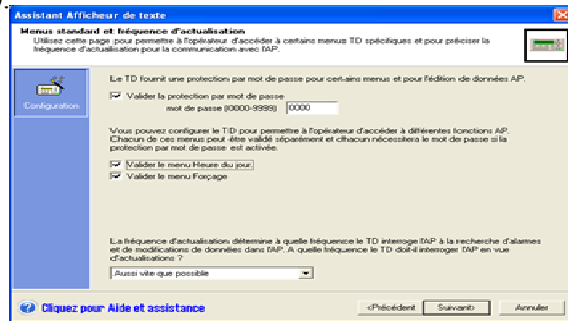


Figure I.2 : Sélection du modèle et version du TD.

3. activation de menus standard, mot de passe et définition de la fréquence de mise à jour :
on cliquant sur suivant la boîte de dialogue de la figure s'affiche.
- La sélection des menus heure du jour et forçage, nous permet d'activer le menu heure du jour et/ou le menu forçage. Une fois qu'on aura activé une sélection, on peut accéder au menu correspondant sur le TD 200, si le menu n'est pas activé, il ne sera pas proposé dans le mode menu du TD 200.



- La sélection de la protection par mot de passe nous permet d'activer un mot de passe a quatre chiffre (de 0000 à 9999). Le mot de passe permet de vérifier si l'opérateur est autorisé à éditer des variables intégrées dans un message et à accéder au mode menu.

- La sélection de la fréquence d'actualisation permet de définir avec quelle fréquence le TD 200 interroge la CPU S7-200 pour afficher les messages.

4. Localisation de l'affichage :
Consiste à la sélection de la langue, que le TD 200 utilise pour les invités et les menus, et du jeu de caractères qui doit être utilisé pour les messages utilisateur.

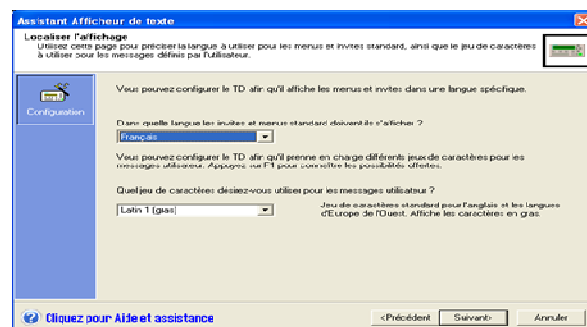


Figure.3 : Localisation de l'affichage.

5. configuration des boutons du clavier :

Le clavier TD 200 définit des boutons pour commander 8 bits de mémoire V. Chaque bouton peut être configuré pour mettre le bit de la CPU à 1 ou bien comme contact sans maintien.

- Si « Mettre bit à 1 » est sélectionné, le bit de mémoire V sera mis à 1 lorsque l'utilisateur enfoncera le bouton correspondant sur le clavier du TD 200. Ce bit restera à 1 et ne pourra être effacé que par la logique du programme.
- Si un bouton est défini comme « Contact sans maintien », le bit de mémoire V sera mis à 1 lorsque l'utilisateur enfoncera le bouton correspondant sur le clavier du TD, mais ne restera à 1 que tant que le bouton du TD sera enfoncé. Lorsque l'utilisateur relâchera le bouton, le bit qui lui est associé en mémoire V sera remis à 0.

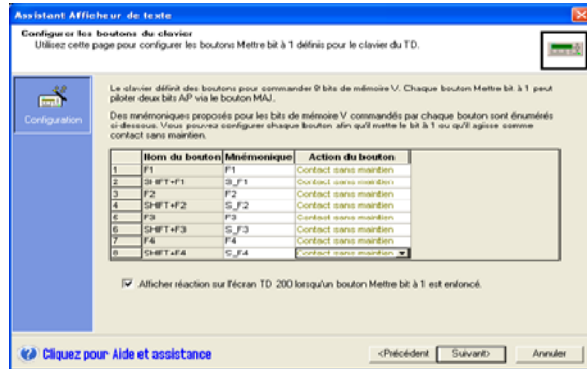


Figure.4 : configuration des boutons du clavier.

Pour notre projet on définit tous les boutons comme « contact sans maintien ».

6. Définition de menus et des textes

L'afficheur TD 200 V3.0 : prene en charge un menu personnalisé qui est affichée sous le contrôle de l'utilisateur au moment de l'exécution. Le TD 200 peut contenir jusqu'à 40 caractères ASCII de texte.

On peut configurer jusqu'à 8 commande de menu, chaque commande de menu pouvant afficher 1 à 8 écrans, pour crée une commande de menu, on entre le texte pour chaque commande que nous désirons utiliser, puis on clique sur le bouton « Editer écrans » pour définir les écrans de texte pour cette commande de menu.

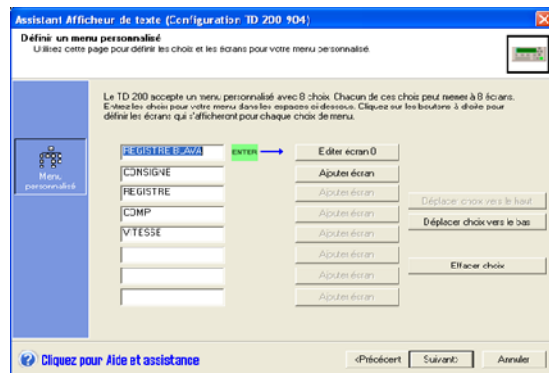


Figure I.5 : Définition de menus et des textes personnalisés.

7. Définition d'option d'alarme et de texte :

Les alarmes sont des messages déclenchés sur bit et ayant une priorité, qui sont affichés sous le contrôle du programme utilisateur. On peut configurer jusqu'à 80 alarmes.

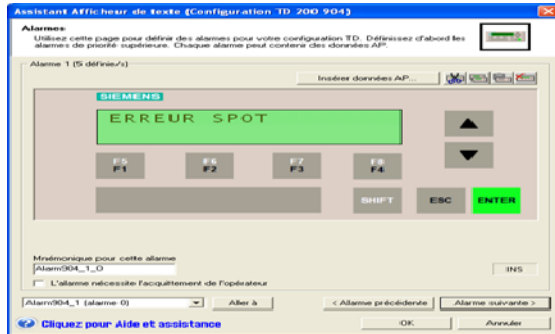


Figure I.6 : Définition d'option d'alarme et de texte.

8. Affectation de mémoire à la configuration:

L'adresse de mémoire V est automatiquement suggérée pour la configuration. L'adresse suggérée subit un contrôle via les références croisées du programme pour garantir que cet emplacement de mémoire n'est pas utilisé par le programme.

Quand on clique sur le bouton Suggérer adresse demande à l'assistant de rechercher le bloc de mémoire V suivant disponible (ayant une taille suffisante). La taille de notre configuration est 602 Octet de mémoire V qui commence à VB904.

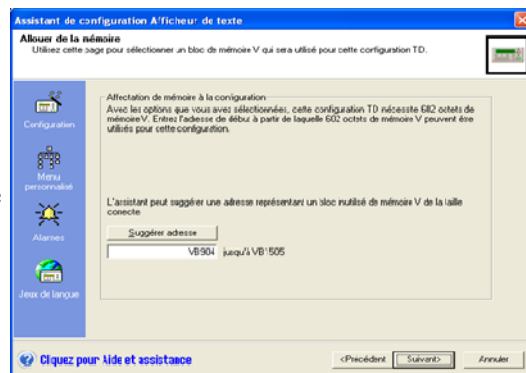
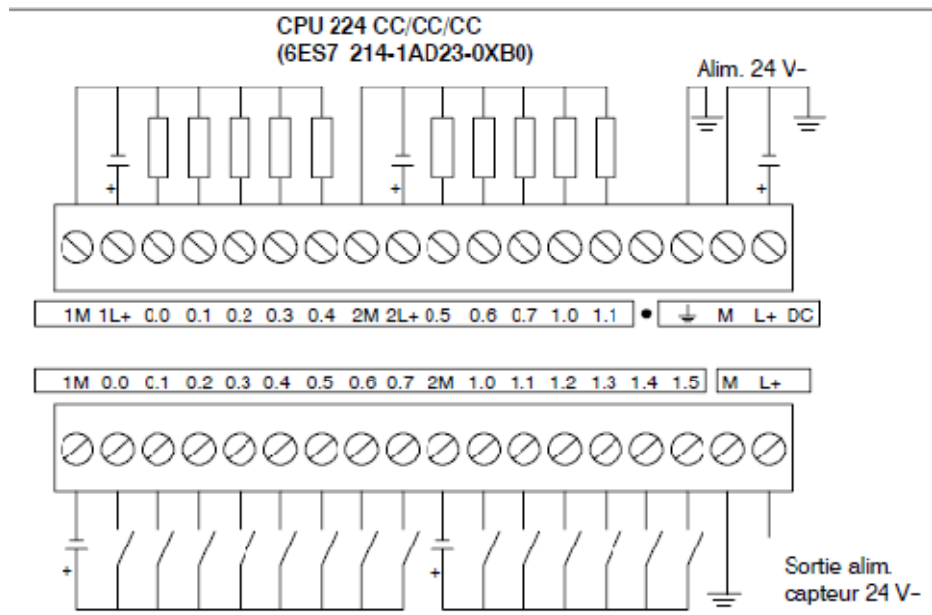


Figure I.7 : Définition d'option d'alarme et de texte.

❖ Les schémas électriques :



« Schéma du câblage de la CPU 224 en courant continu ».

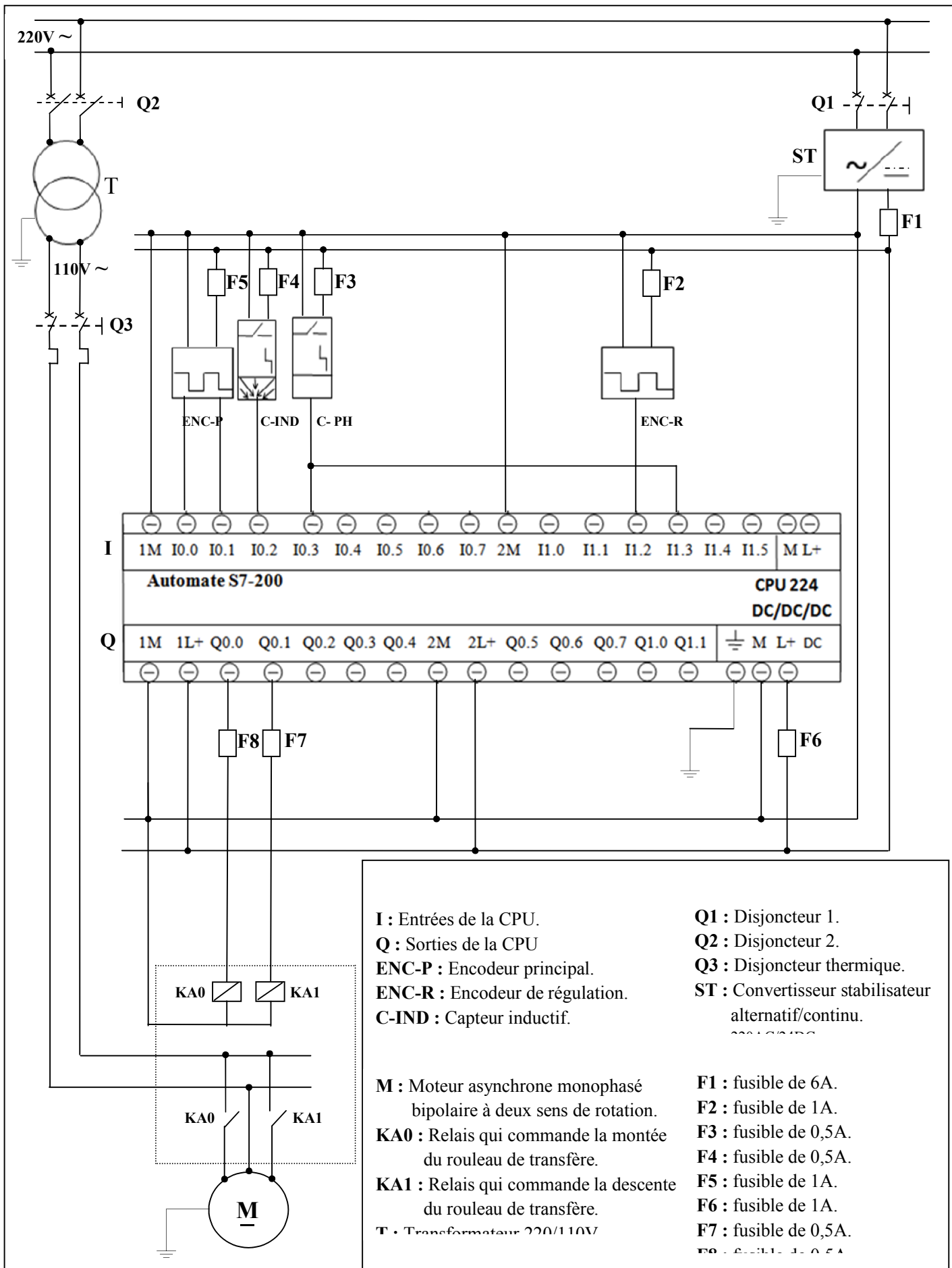


Schéma de commande du bloc de découpage.



Bibliographie

- [1] Y. LICOURTIER et B. SAINT-JEAN
Introduction aux automates programmable industriels.
Edition MASSON 1989.
- [2] GILLES MICHEL
Edition BORDAS, PARIS 1988
API : Architecture et applications des automates programmables industriels.
- [3] AMAR.YAHIA CHERIF
Logique et calculateurs.
Les éditions l'Abeille, 2003.
- [4]
Automatismes, Outils graphiques, composants logique, commandes (pneumatiques, électrique). Année 1983. AUTOMATIQUE108
- [5] BELIK RAMDANE
Programmation d'un palettiseur par un automate programmable SIEMENS.
Mémoire de fin d'études d'ingénieur en AUTOMATIQUE, promotion 2006.
- [6] Utilisation de réseaux industriels pour l'automatisation d'un procédé complexe.
Mémoire de fin d'études d'ingénieur en automatique, promotion 2005.
- [7] S. HADDAD, F. MANANE et S. GUICI
Etude d'une commande d'une station de pompage d'eau pilotée par automate programmable industriel.
Mémoire de fin d'études d'ingénieur en ELECTROTECHNIQUE, promotion 2005.
- [8] BOUDISSA.M et BENLALLI.A
Conception d'une commande programmable d'une plieuse de tôle à base d'un automate Programmable S7-300.
Mémoire e fin d'études d'ingénieur en AUTOMATIQUE, promotion 2006
- [9] SIEMENS SIMATIC.
Manuel système Automate programmable S7-200, Edition 08/2008,
Numéro de référence 6ES 7298-8FA24-8CHO.
- [10] SIEMENS SIMATIC.
Guide de l'utilisateur Afficheur TD200, 5^{eme} édition 10/99.
- [11] CD STEP 7 Micro/WIN 32 Logiciel et Aide.
- [12] WWW.datasheet.com
Répertoire de datasheet des composants électroniques sur internet.

- [13] Document API :
WWW.SIEMENS.fr
WWW.google.fr /les Automates Programmables Industriels.
WWW.google.fr/STEP7micro/WIN.
- [14] document imprimerie :
WWW.wikipedia.com
- [15] « Guide pratique des capteurs ».Télemécanique.