Automatique :

Les systèmes automatisés de production
Les automates programmables

Laboratoire

Nathalie Gillieaux-Vetcour
Christine Stiernon-Giard

BAC 3

Table des matières

Les systèmes automatisés de production
Les automates programmables

Objectifs du laboratoire

Découvrir, analyser, mettre en œuvre les différents composants d'un système automatisé de production (SAP) et en particulier pour la partie commande : les automates programmables (API)

<u>1ère partie : Partie commande</u>

- *<u>3 séances</u>*

- Structure du SAP : description de ses principaux composants.
- Fonctionnement et programmation des automates programmables.
    1. Fonctionnement et adressage de l'AP, utilisation du logiciel de programmation
    2. Fonctions logiques combinatoires et séquentielles : ET, OU, bascules RS, temporisations, compteurs, comparateurs

    => Notions appliquées à la gestion d'une chaîne d'emballage

    => Applications : exercices divers

- *<u>1 séance</u>* : exercice récapitulatif de la programmation de base

Travail en groupes de 2

<u>2ème partie : partie opérative et partie commande</u>

Etude et Programmation de la mini-chaîne FESTO

- *<u>2 séances :</u>* Aperçu des principaux composants d'un système automatisé de production

  => notions de pneumatique, périphérie décentralisée, AU
  => réalisation d'un dossier de programmation
- *<u>2 séances :</u>* Application de la programmation
- *<u>1 séance :</u>* test labo sur la programmation structurée
- *<u>3 séances :</u>* Etude du grafcet et programmation de l'installation selon la méthode grafcet.

Travail « non dirigé », en groupes, recherche d'informations dans la documentation fournie, solutions personnalisées ...

- *<u>1 séance :</u>* Présentation orale du travail réalisé

Introduction : Structure des systèmes automatisés

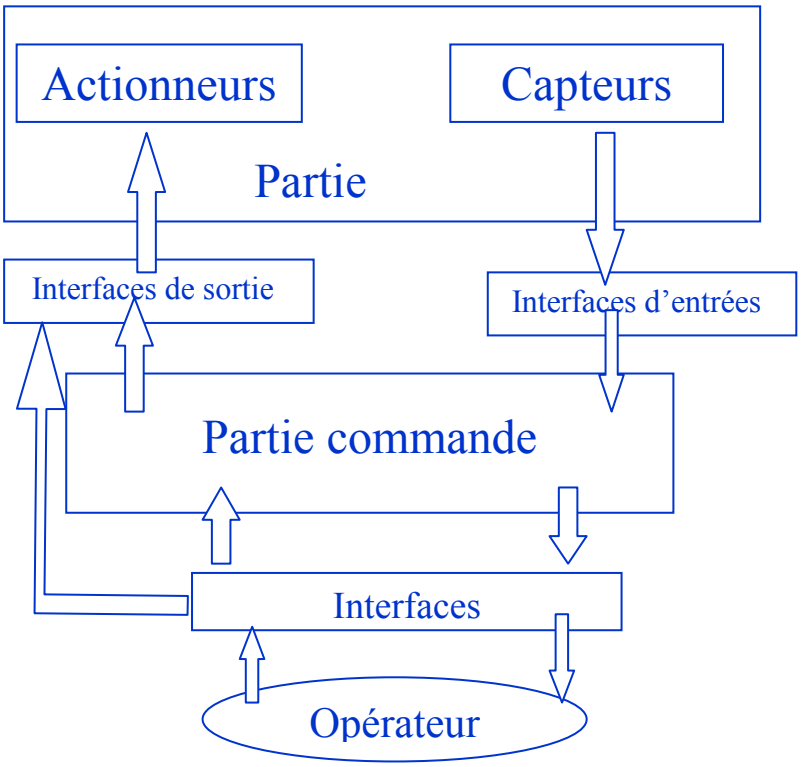
Automatique = qui opère, fonctionne sans intervention humaine
Selon des règles préétablies

L'automatisation est une nécessité du monde industriel dans beaucoup de secteurs d'activités. Ses avantages se retrouvent au *niveau technique* : compétitivité, qualité des produits, souplesse d'évolution, flexibilité des tâches, précision… mais aussi au niveau de *l'amélioration des conditions de travail* : gestion de tâches répétitives, augmentation des cadences de production, sécurité des opérations lourdes, …

L'automatisation est un facteur d'amélioration du bien-être dans le domaine des dispositifs domestiques (chauffage central, …).

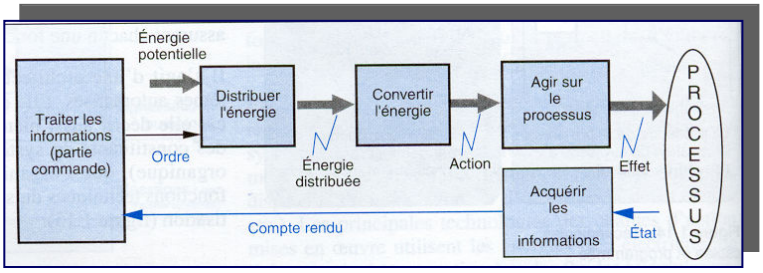
En ce qui concerne les technologies mises en œuvre, l'automatique des systèmes logiques exploite des composants issus de l'électricité, l'électronique, la pneumatique, l'hydraulique, la mécanique, les automates programmables, l'informatique…

Schéma-bloc d'un système automatisé :



Partie commande :

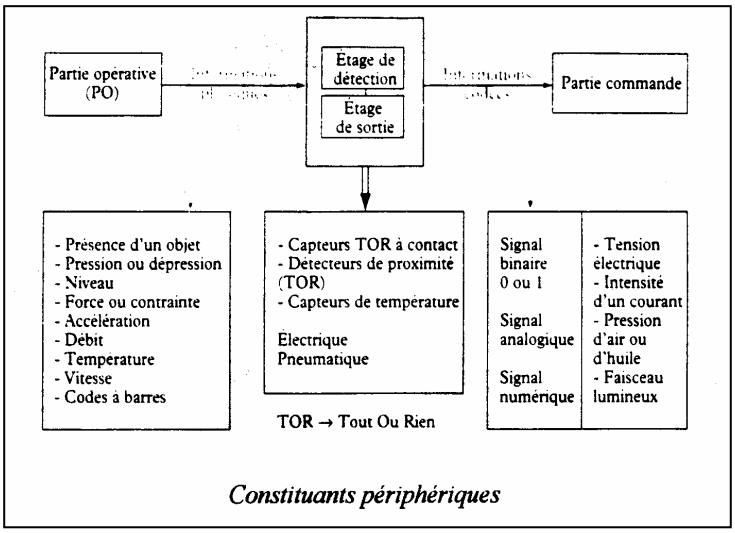
Un « expert », extérieur au système automatisé, a mémorisé son savoir-faire (son expérience) dans la partie commande. La partie commande reçoit des ordres de marche et consignes de l'opérateur par l'intermédiaire d'un *pupitre*. L'opérateur est informé par des voyants, afficheurs, écrans, girophares, buzzers, …
La partie commande est composée d'une partie traitement, «cerveau» logique de la machine. Cette partie est constituée de cartes électroniques ou à microprocesseurs, automates programmables industriels, séquenceurs pneumatiques, électriques, relais ou programmateurs cycliques... Elle envoie des ordres à la partie opérative. Elle reçoit des comptes rendus ou informations de capteurs de fin de course, de pression, de température, de vitesse... Elle peut dialoguer, ou reboucler des informations avec d'autres parties commandes.
Elle est composée d'interfaces, les pré-actionneurs, permettant de transmettre les ordres de commande (en faible tension, 24 V par exemple) au circuit de commande de puissance des actionneurs (en forte tension, 380 V - par exemple). Le pré-actionneur pour un vérin est un distributeur. Le pré-actionneur pour un moteur électrique est un contacteur.



Partie opérative :

La partie opérative, càd machine ou installation de production, est composée d'actionneurs: moteurs, vérins, électrovannes, lampes, résistances, électro-aimants... Elle réalise les opérations sur la matière par l'intermédiaire d'objets techniques ou effecteurs, actionnés par les actionneurs. Elle informe la partie commande du déroulement des opérations par l'intermédiaire des capteurs qui mesurent et codent des grandeurs physiques en informations logiques:

- capteurs à contact : à galet, à tige, à poussoir…
- capteurs sans contact : de proximité, à infra-rouge, à ultra-sons...



Technologie des constituants de la partie commande

Quatre technologies se partagent actuellement le «marché» des automatismes séquentiels : la pneumatique câblée, l'électromécanique câblée, l'électronique par cartes spécifiques et la logique programmée avec les automates programmables et la micro ou mini informatique spécifique.

- **Systèmes logiques pneumatiques.**

Un ensemble de vannes, distributeurs, cellules logiques, capteurs et vérins reliés entre eux par tuyauterie pneumatique constitue l'automatisme. Le système est idéal pour les applications dans lesquelles les actionneurs (vérins) sont pneumatiques ou lorsque l'ambiance dans laquelle est installée la machinerie est explosive (aucun risque d'étincelles). L'encombrement des équipements augmente très rapidement avec la complexité du système et le câblage (installation de la tuyauterie) est fastidieux. Il nécessite la présence d'une alimentation pneumatique adéquate.

- **Systèmes à relais et contacteurs.**

Ils représentaient les systèmes automatiques les plus employés dans le milieu du vingtième siècle. Ils permettent la commande des moteurs (courant continu et alternatif). Le coût du matériel et des installations est relativement bon marché mais la complexité est limitée par l'encombrement et les problèmes de câblage (voir armoires), il en résulte des systèmes peu flexibles.

- **Les circuits séquenceurs.**

Un circuit séquenceur permet de commander et de contrôler le déroulement du cycle d'une machine ou d'une installation. Mais une fois déterminé, le programme est fixe et se répète dans le même ordre chronologique. Le séquenceur est un système câblé qui peut être réalisé en technologie électrique, électronique ou pneumatique.

- **Systèmes à circuits intégrés logiques et cartes électroniques.**

Ils permettent la miniaturisation des automatismes. Ils sont néanmoins sensibles aux parasites, aux températures supérieures à 60°C, aux surcharges électrostatiques et nécessitent des interfaces pour piloter les actionneurs de puissance. Les systèmes conçus à base de circuits intégrés sont très rapides et mènent à des dispositifs de conception relativement bon marché. Distinguons :

- Les **cartes électroniques spécifiques** : réalisées spécialement pour le problème posé; compte tenu des investissements nécessaires pour chaque type de carte, on peut estimer que les séries doivent être d'au moins 500 cartes identiques pour être compétitives.
- Les **cartes électroniques standard** : dont la mise en œuvre (sélection, implantation, programmation, mise au point) exige que les systèmes soient réalisés au minimum à 50 exemplaires identiques pour être compétitifs. Fréquentes en automatismes grand public (parking, machines à billets...), ces séries sont rares en automatismes de production.

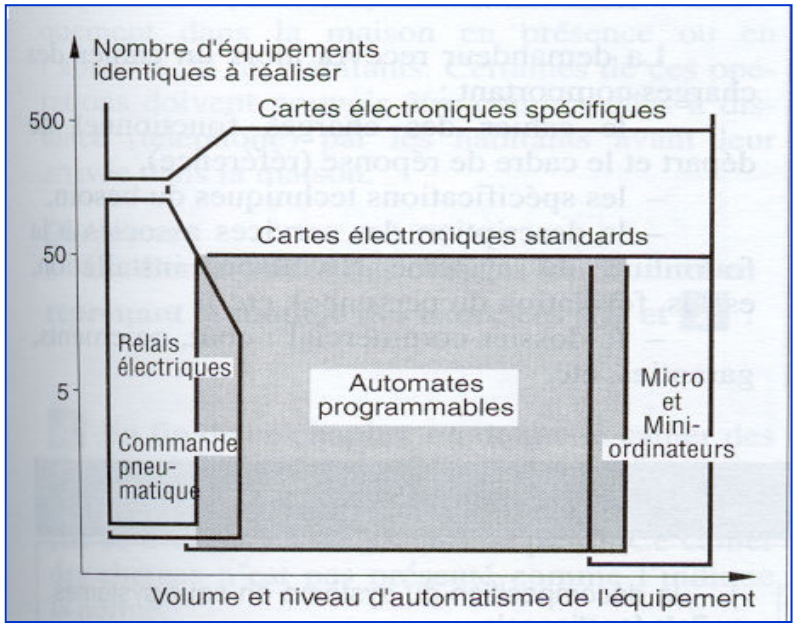
- **Les automates programmables.**

Les automates programmables sont conçus pour réaliser un cycle de fonctionnement à partir d'un programme stocké en mémoire. Il s'agit d'un système de commande informatisé, interfacé pour la commande de processus industriels.

En automatisation de production, les séries importantes sont rares car même chez les constructeurs spécialisés les variantes et adaptations sont courantes. Ils sont donc particulièrement prisés dès que le problème est complexe ou exige flexibilité et évolutivité.

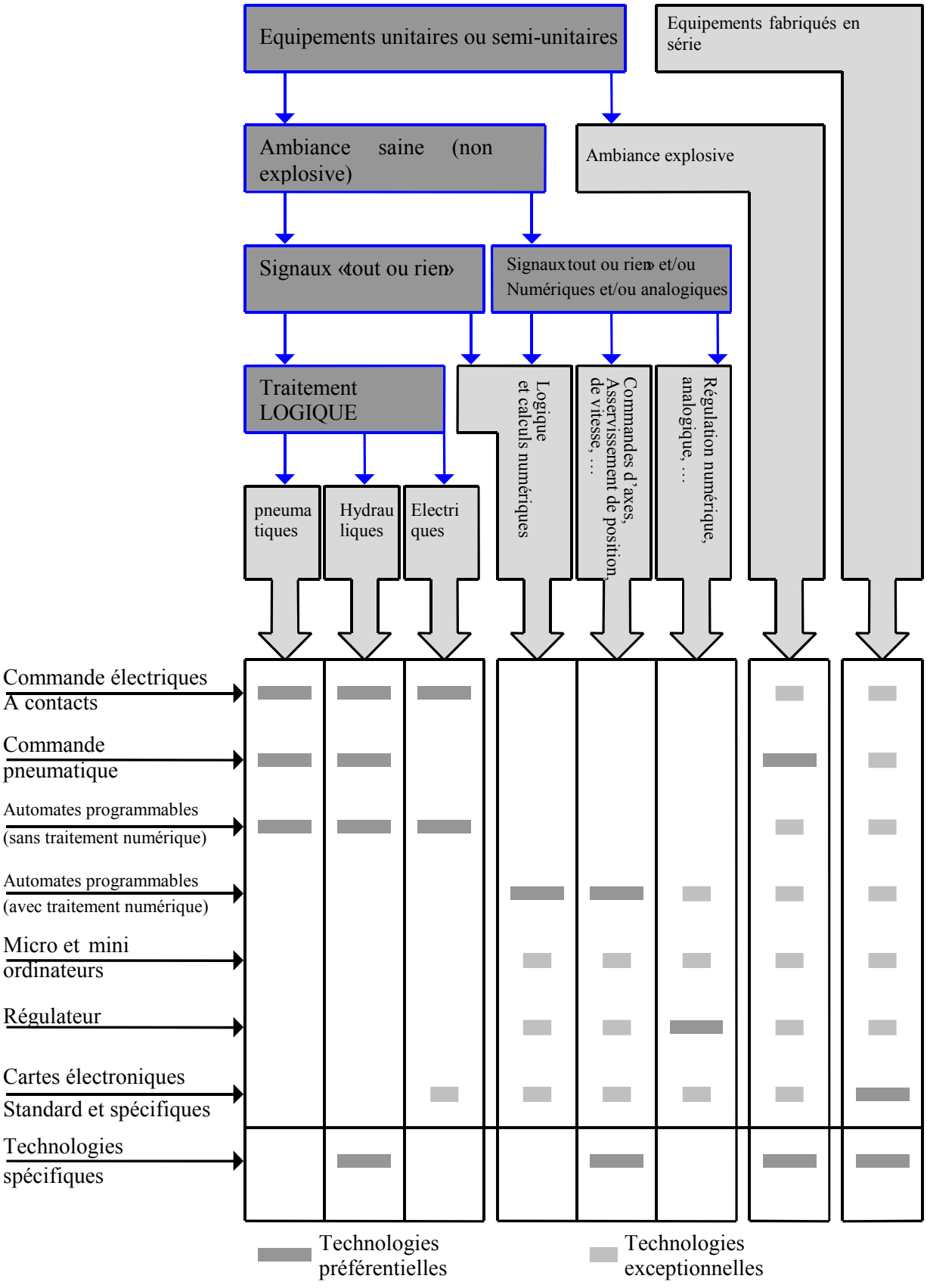
Les avantages des API sont :

- Moins de câblage
- Possibilité de programmation
- Modification et mise au point plus facile
- Extension très aisée
- Performance de calcul requise pour des applications plus complexes (régulation, positionnement, ...)
- Gestion de données (production, défaut…)
- Possibilité de communications …



Le dialogue homme/machine est affiné avec les automates et les processeurs programmés qui affichent et éditent en permanence l'état du système. Les automatismes comportent de plus en plus de fonctions de calcul et de traitements de données. Malgré l'importance des interfaces, le coût des systèmes programmés est faible, et leur capacité d'évolution grande. Une orientation vers la généralisation des systèmes programmés décentralisés et réceptifs à des langages de plus en plus évolués et performants se dessine incontestablement.
Remarquons encore que la conséquence de toutes ces évolutions est de provoquer une multitude de philosophies différentes de conception d'automatismes et une notion de compatibilité quasi inexistantes entre fabricants.

Voici un tableau reprenant les différents critères de choix de la technologie de la partie commande d'un système automatisé [1] :



1 extrait des technoguides de l'ADEPA.

Partie opérative

Pour rappel, dans un système automatisé, différentes fonctions doivent permettre de :

- **Détecter** par des capteurs les informations issues du procédé physique,
- **Actionner** par de l'équipement électrique, électronique, pneumatique et hydraulique …
- **Echanger** des informations avec l'utilisateur au travers de dispositifs de dialogue.

Ces différentes fonctions nécessitent des périphériques adaptés aux rôles qu'ils doivent remplir.

1 Périphériques d'entrées

1.1 Capteurs

Pour exploiter correctement un système automatisé il est nécessaire :

- **de mesurer les variations de certaines grandeurs physiques :**
  - la vitesse du vent pour un store automatisé
  - la pression d'air dans le réseau d'alimentation d'un automatisme pneumatique
  - la température de l'eau dans un lave-linge
  - …

- **de contrôler l'état physique de certains de ses constituants :**
  - la position levée d'une barrière de parking,
  - la présence d'une pièce sur un convoyeur,
  - la présence de pression dans un circuit,
  - la position d'un chariot
  - …

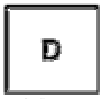
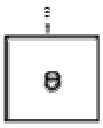
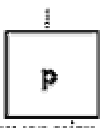
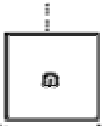
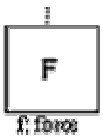
Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.

1.1.1 Nature des Capteurs

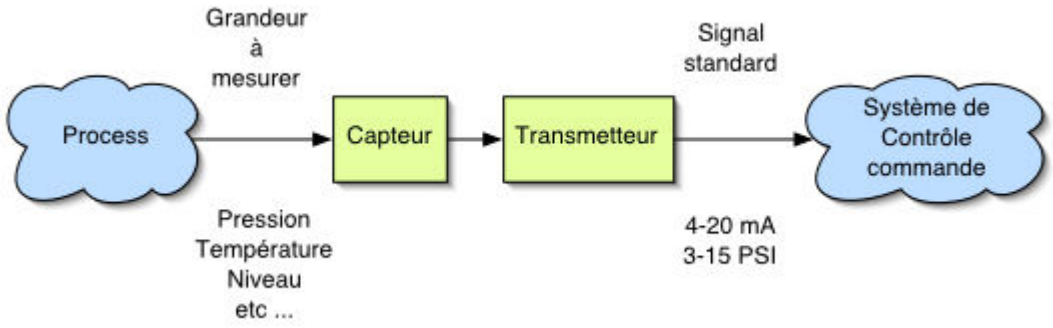
Suivant la nature du signal exploitable, les capteurs se classent en trois catégories:

- Capteurs analogiques, le signal délivré est la traduction exacte de la loi de variation de la grandeur physique mesurée,
- Capteurs logiques, le signal ne présente que deux niveaux, ou deux états, qui s'affichent par rapport au franchissement de deux valeurs; ces capteurs du type tout ou rien sont également désignés par détecteurs,
- Capteurs numériques, le signal est codé au sein même du capteur par une électronique associée; ces capteurs sont également désignés par codeurs et compteurs.

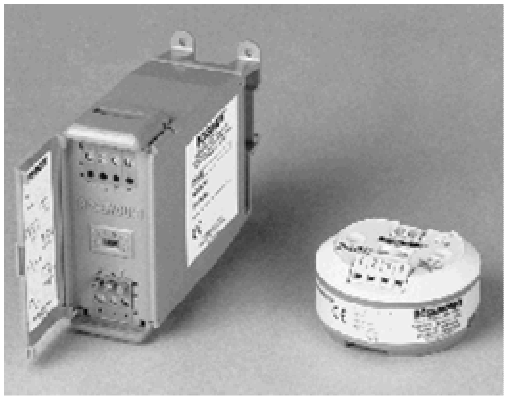
Plusieurs critères de classification des capteurs sont possibles.

INFORMATION	Ex. de CAPTEURS	SYMBOLES	CARACTERISTIQUES	APPLICATIONS
INFORMATION DE POSITION (par contact avec l'objet)	CAPTEUR PNEUMATIQUE DE POSITION	 rappel par ressort	Commande par galet Rappel par ressort Vitesse et attaque maximale :0,1 m/s Force minimale nécessaire pour l'enclenchement à 6 bars: 2daN	Très nombreuses compte tenu: -de la robustesse -de la bonne résistance aux agents extérieurs, huiles, acides, poussières,..
INFORMATION DE DEPLACEMENT (par contact avec l'objet)	CODEUR OPTIQUE INCREMENTAL	 D: déplacement	Tension d'alimentation 5 à 24 V continu Vitesse de rotation de l'arbre: 6000 tr/min. max. Résolution comprise entre 2500 et 7200.	Le positionnement du mobile est entièrement maîtrisé par les systèmes de traitement associés au codeur ce qui convient à des bras de robots, des tables de machines-outils,...
INFORMATION DE TEMPERATURE D'UN FLUIDE	SONDE A THERMISTANCE	 θ: température	Variation de la résistivité d'un semi-conducteur avec la température Grande sensibilité Temps de réponse très court	Grand champ d'applica-tion de -100 à +450° C Utilisé pour la détection de la variation de tempéra-ture.
INFORMATION DE PRESSION DANS UN CIRCUIT	PRESSOSTAT	 p: pression	Nature du fluide Fréquence de fonctionnement Mode de fonctionnement des contacts Endurance électrique	Appareils très robustes ayant un champ d'applica-tion dans les circuits de fluide: eau, air,...
INFORMATION DE VITESSE ANGULAIRE	DYNAMO TACHYMETRIQUE A COURANT CONTINU	 ω: vitesse angulaire	Gamme étendue de mesure Donne le sens de rotation Bonne linéarité Bonne précision	Champ d'application étendu sur une gamme de vitesses pouvant être très élevées (7500 tr/min.).
INFORMATION DE PRESSION DE FORCE ET DE POIDS	JAUGE METALLIQUE EXTENSOMETRIQUE	 f: force	Bonne précision Plage d'utilisation en température étendue (de -30 à 120°C) Haute limite de fatigue Electronique de traitement simple	Le prix de revient de ces capteurs à transduction résistive fait que leurs applications sont nombreuses dans la recherche et dans l'industrie.

1.1.2 Le rôle du transmetteur



C'est un dispositif qui converti le signal de sortie du capteur en un signal de mesure standard. Il fait le lien entre le capteur et le système de contrôle commande.



1.1.2.1 Raccordement électrique

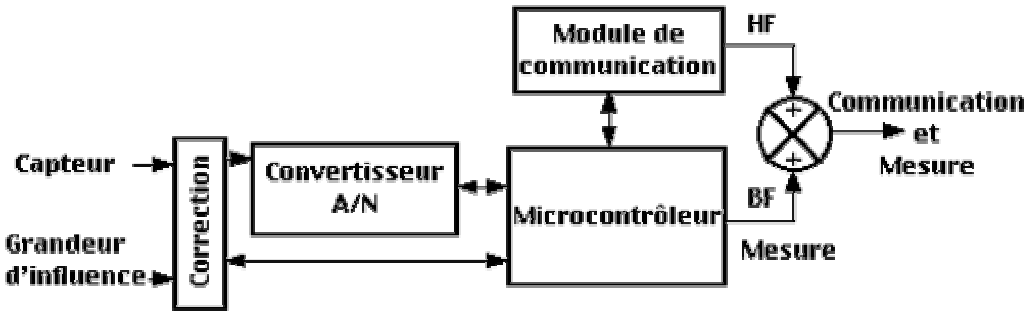
- Les transmetteurs 4 fils (actifs) qui disposent d'une alimentation et qui fournissent le courant I. Leur schéma de câblage est identique à celui des régulateurs :

- Les transmetteurs 3 fils (actifs) sont des transmetteur 4 fils, avec les entrées moins reliées :

- Les transmetteurs 2 fils (passif) qui ne disposent pas d'une alimentation et qui contrôle le courant I fourni par une alimentation externe. Leur schéma de câblage est généralement le suivant :

1.1.2.2 Le transmetteur "intelligent"

Le transmetteur intelligent est un transmetteur muni d'un module de communication et d'un microcontrôleur :



Le module de communication permet :

- De régler le transmetteur à distance ;
- De brancher plusieurs transmetteurs sur la même ligne.

Le microcontrôleur permet :

- De convertir la mesure en une autre grandeur, appelée grandeur secondaire. Par exemple, il peut convertir une mesure de différence de pression en niveau (voir chapitre sur les mesures de niveau).
- De corriger l'influence des grandeurs d'influence sur la mesure.

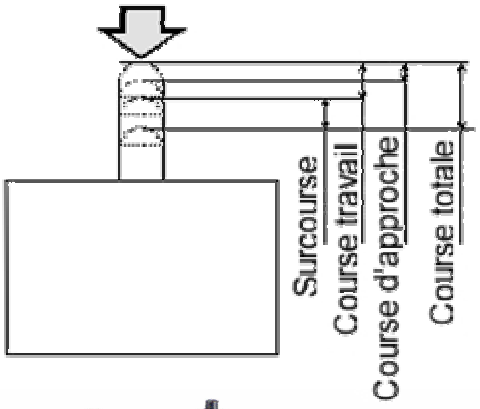
Avantages métrologique du transmetteur "intelligent" :

- Précision. En effet, le transmetteur possède moins de composants analogiques. Les grandeurs d'influences sont compensées. La non linéarité du transducteur peut être corrigé.
- Rangeabilité
- Répétabilité
- Autosurveillance - Position de repli
- Traitement du signal – Filtrage

Avantages à la configuration et maintenance

- Convivialité - Accès à distance
- Standardisation
- Diagnostic - Forçage du signal de sortie

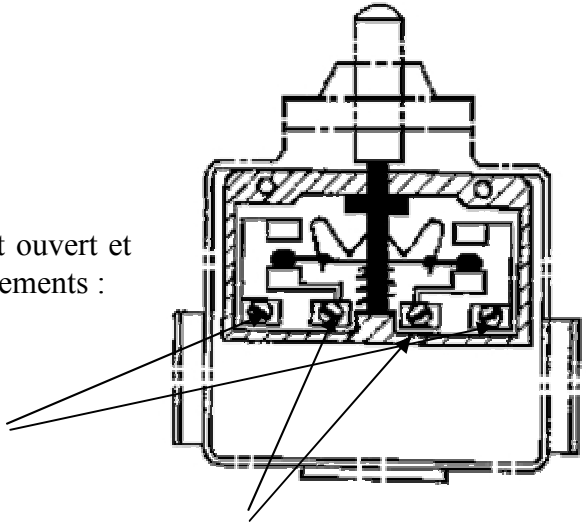
1.2 Les interrupteurs de position ou détecteurs de contact



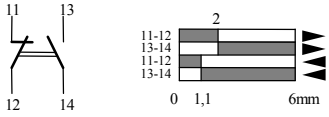
Ces détecteurs de position « tout ou rien » se rencontrent sur de nombreuses machines : robots, machines-outils, machines d'assemblage,...

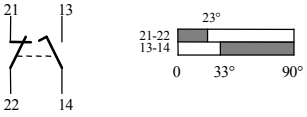
Une multitude de types au point de vue encombrement, fonctionnement mécanique, fonctionnement électrique.

Ils possèdent pour la plupart un contact normalement ouvert et un contact normalement fermé et peuvent être à changements :

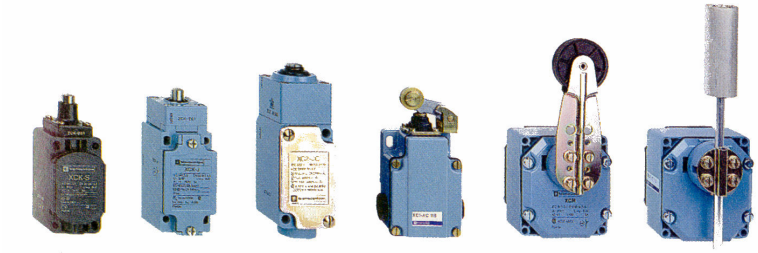


unipolaire (en même temps) ou décalés (le NF s'ouvre avant que le NO ne se ferme et inversement).





Du côté mécanique, ces capteurs réagissent à un mouvement linéaire par action en ou par mouvement angulaire.



Il existe également des interrupteurs de sécurité à clés, à rappel au zéro ou au contraire à positions maintenues.

1.3 Détecteurs de proximité

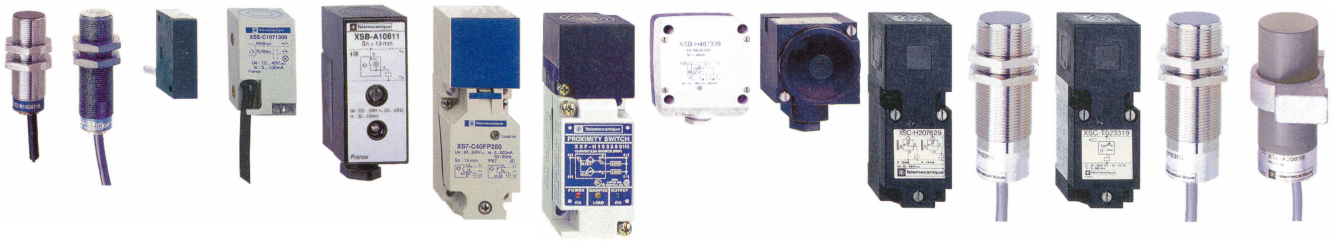
Ces détecteurs opèrent à distance, *sans contact physique* avec l'objet dont ils contrôlent la position (depuis 1 mm à quelques mètres) ⇒ pas d'usure, possibilité de détecter des objets fragiles, cadences de travail élevées ….

Un détecteur de proximité interrompt ou établit un circuit électrique en fonction de la présence ou de la non-présence d'un objet dans sa zone sensible. Dans tous ces détecteurs la présence de l'objet à détecter dans la zone sensible modifie une grandeur physique :

- un champ électromagnétique à haute fréquence dans les détecteurs inductifs;
- la capacité d'un circuit oscillant dans les détecteurs capacitifs,
- le niveau d'éclairement d'un récepteur photosensible dans les détecteurs photoélectriques.

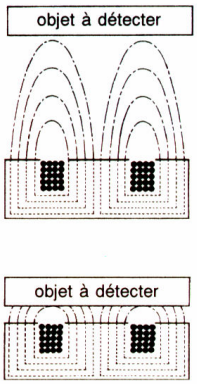
Le choix d'un détecteur de proximité dépend:

- de la nature du matériau constituant l'objet à détecter,
- de la distance de l'objet à détecter,
- des dimensions de l'emplacement disponible pour implanter le détecteur.

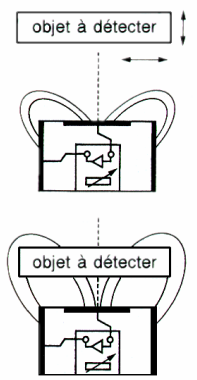


➢ **Détecteurs inductifs :**

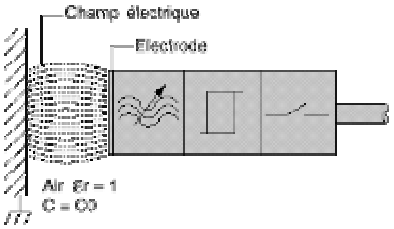
Ce type de détecteur est adapté à la détection *d'objets métalliques*. Un détecteur inductif se compose essentiellement d'un oscillateur dont le bobinage, logé dans un circuit magnétique, engendre un champ magnétique alternatif. Ce champ sort du corps de l'appareil par sa face sensible. Un écran métallique placé dans le champ crée des courants induits et provoquent l'arrêt des oscillations.
Après la mise en forme, un signal de sortie correspondant à un contact à ouverture ou à fermeture est délivré.
Robustes et fiables ces détections ont l'inconvénient d'être sensibles à la présence de poussières métalliques qui peuvent perturber leur fonctionnement en provoquant des détections parasites.
Portée de quelques millimètres à quelques centimètres



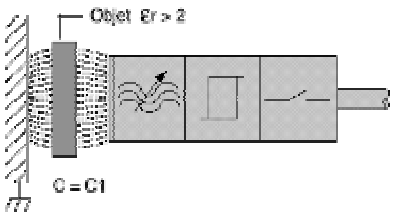
- **Détecteurs capacitifs :**



Dans le cas du détecteur capacitif, l'objet à détecter fait varier par sa position la capacité d'un condensateur formé par la face sensible du détecteur et l'objet lui-même s'il est métallique ou la masse électrique environnante s'il est isolant.

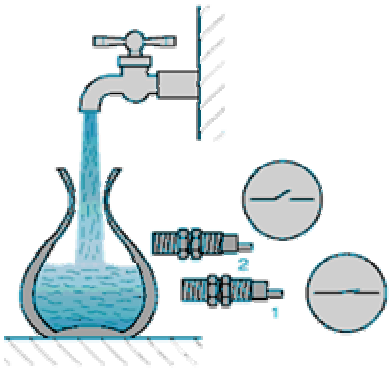


Dans l'air ε_r=1 la capacité de ce condensateur est C0.



Lorsqu'un objet de nature quelconque ε_r=2 se trouve en regard de la face sensible du détecteur, ceci se traduit par une variation du couplage capacitif (C1).
Cette variation de capacité (C1>C0) provoque le démarrage de l'oscillateur.

Ses caractéristiques lui permettent de détecter tout objet même si celui-ci n'est pas métallique. Ce type de détecteur est adapté à la détection *d'objets isolants, liquides, poudres* ... ou à la détection à travers une paroi mince et non-métallique :

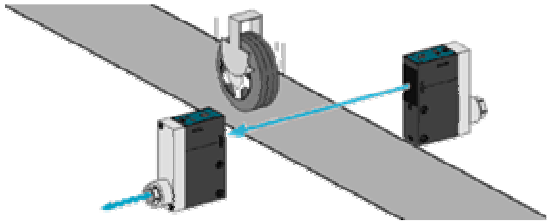


Après la mise en forme, un signal de sortie correspondant à un contact à ouverture ou à fermeture est délivré.
Portée nominale de 15mm (la portée réelle dépend de la matière de l'objet à détecter)

1.4 Détecteurs photoélectriques

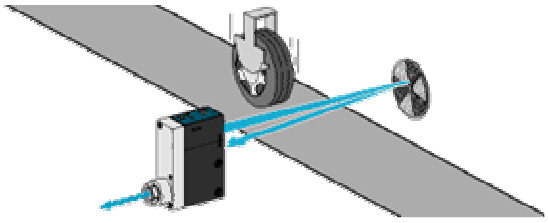
Les détecteurs photoélectriques se composent essentiellement d'un émetteur (diode électroluminescente émettant de manière modulée dans le domaine proche de l'infrarouge ou du visible) associé à un récepteur photosensible. L'objet est détecté lorsqu'il interrompt ou permet d'établir, ou encore fait varier, l'intensité du faisceau lumineux sur le récepteur..

- **Système barrage :** Emetteur et récepteur sont séparés.
  A l'exception des objets transparents qui ne bloquent pas le faisceau lumineux, ce système de détection permet de détecter tout objet (opaque, mat ou réfléchissant) interrompant le faisceau lumineux.



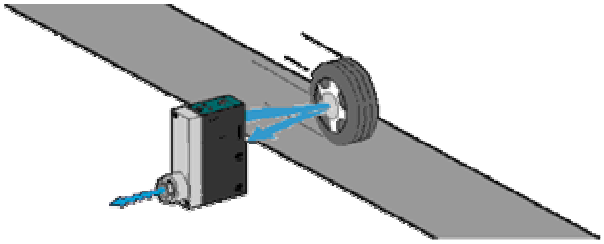
  Il possède une grande portée, une détection précise, est adapté aux environnements pollués (fumée, poussières, emplacements soumis aux intempéries,...) et peut être placé à l'extérieur.
  L'alignement entre émetteur et récepteur doit être réalisé avec soin.
  Portée de 4m à 30m

- **Système réflex :** Emetteur et récepteur sont dans le même boîtier. En l'absence de cible, le retour du faisceau est assuré par un réflecteur (catadioptre) monté en vis à vis.
  La détection est réalisée lorsque la cible bloque le faisceau entre l'émetteur et le réflecteur. C'est donc un système qui n'est pas adapté pour la détection d'objets réfléchissants qui pourraient renvoyer une quantité plus ou moins importante de la lumière sur le récepteur. Son utilisation est un peu plus limitée bien qu'il soit moyennement exploitable dans un environnement pollué. Il bénéficie d'une simplicité de mise en œuvre et tolère jusqu'à 15° d'erreur dans la précision de l'alignement.



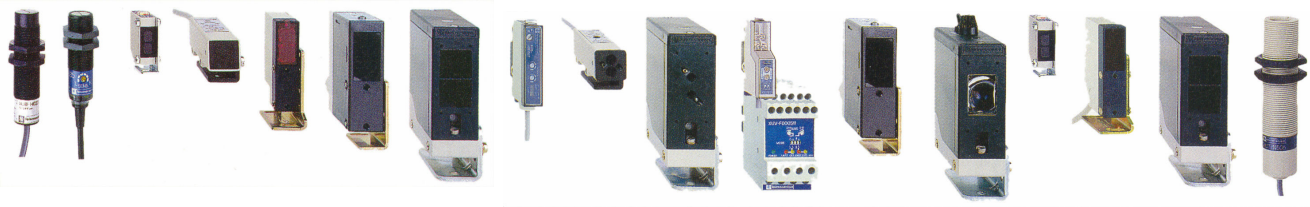
  Portée de 2m à 10m

- **Système proximité :** Emetteur et récepteur sont incorporés dans le même boîtier. Le faisceau est réfléchi en partie vers le récepteur par tout objet se trouvant à proximité.
  Portée de 20mm à 70cm
  La portée d'un système proximité est généralement inférieure à celle d'un système reflex. Pour cette raison, son utilisation en environnement pollué est déconseillée. Cette portée dépend de la couleur de la cible, de son pouvoir réfléchissant et de ses dimensions.

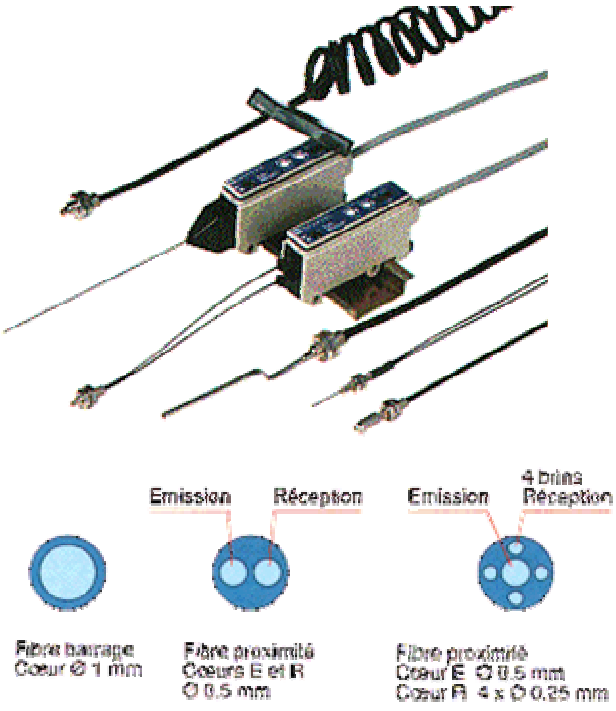


- **Système proximité avec l'effacement de l'arrière plan :** Un système de triangulation permet de mesurer la portée au moment de la détection et de valider cette information en fonction du réglage. Ce système est approprié à la détection d'objets de couleurs ou de réflectivités différentes et élimination d'un arrière plan.

Inconvénient du principe émission-réception d'un faisceau lumineux : la contamination des optiques par l'environnement (poussière, pluie, givre, brouillard, etc.) se traduira par une atténuation du niveau de détection pouvant aller jusqu'à un fonctionnement intermittent dans les cas extrêmes.

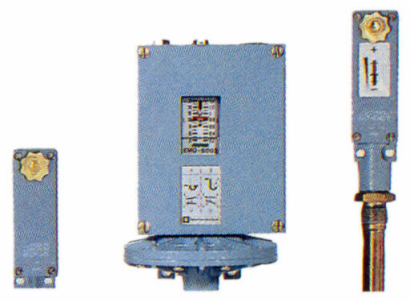


- **Système à fibre optique :** Les détecteurs photoélectriques peuvent utiliser des fibres optiques permettant d'éloigner l'émetteur et le récepteur par rapport au point de détection. La lumière est véhiculée entre ce point et l'amplificateur par des fibres optiques qui, grâce à leurs faibles dimensions, peuvent s'intégrer dans les emplacements les plus exigus. Ces appareils sont également parfaitement adaptés pour la détection de cibles de très petites tailles (vis, rondelles, capsules...).
  Ces détecteurs existent en système barrage et proximité.

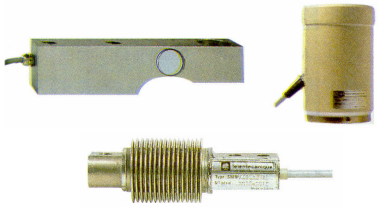


1.5 Pressostats, vacuostats, thermostats :

Détecteurs pour contrôler (surveillance de seuil) ou réguler une pression, une dépression une température ou une charge. Suivant la ou les valeurs atteintes et les valeurs de réglage, le contact électrique associé change d'état.



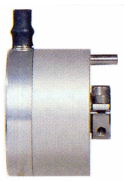
1.6 Les capteurs de pesage



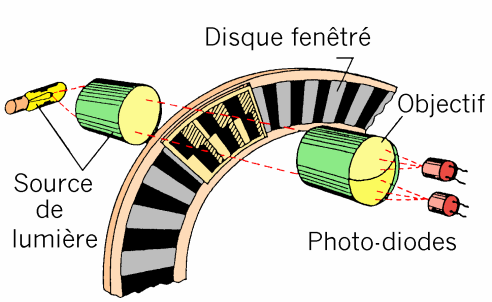
Utilisés pour les opérations de pesage, dosage et conditionnement.
Le détecteur de charge repose sur des *capteurs à jauges de contrainte* pour tous types de récepteurs de charge : plateau, trémie, silo, pont-bascule, pont-roulant, bande transporteuse…qui transforme le poids en signal analogique.
Trois gammes :

- Capteurs de flexion de capacités nominales de 10 à 200kg
- Capteurs de traction-compression de capacités de 500 à 5000kg
- Capteurs de cisaillement de capacités de 500 à 2000kg

1.7 Les codeurs rotatifs

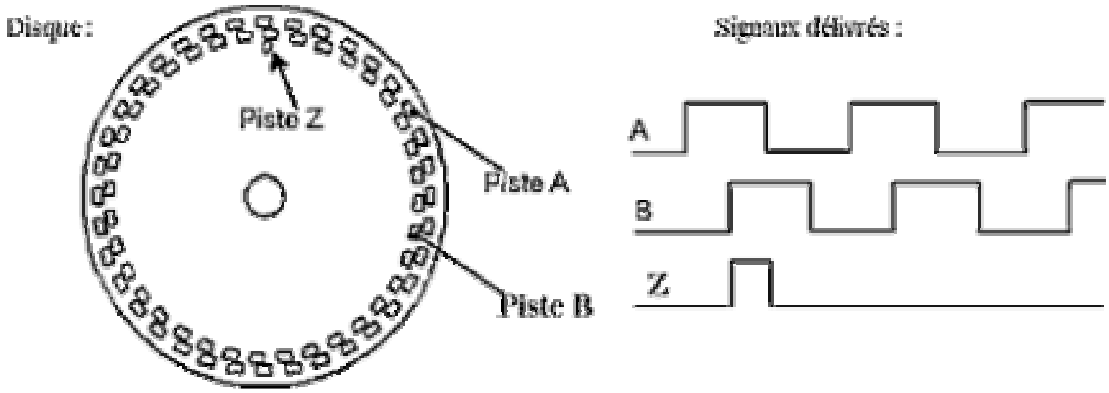


Le codeur rotatif est un *capteur de position ou de vitesse angulaire*. Lié mécaniquement à un arbre qui l'entraîne, son axe fait tourner un disque qui comporte une succession de zones opaques et transparentes.
La lumière émise par des diodes électroluminescentes LED arrive sur des photodiodes chaque fois qu'elle traverse les zones transparentes du disque. Ce signal amplifié puis converti en signal carré est transmis à un système de traitement.

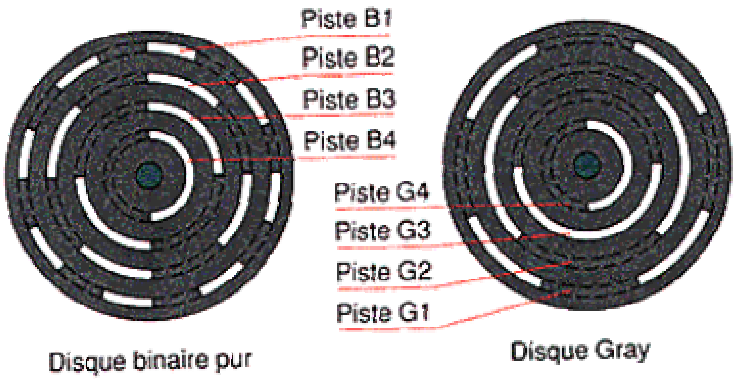


Les codeurs existent sous les types de codeurs incrémentaux (ou générateur d'impulsions) ou codeurs absolus :

- ***Codeur incrémental :*** Le disque d'un codeur incrémental comporte 3 pistes : Deux pistes A et B divisées en « n » intervalles d'angles égaux et alternativement opaques et transparents. « n » permet de définir la résolution ou période. La piste A est décalée de ¼ de période par rapport à B. Le déphasage entre A et B permet de définir le sens de rotation. Deux photodiodes délivrent des signaux carrés pour les pistes A et B chaque fois que le faisceau lumineux traverse une zone transparente. Une piste Z comporte une seule fenêtre transparente. Le signal Z appelé « top zéro » est synchrone avec les signaux A et B. Il définit une position de référence et permet la réinitialisation à chaque tour.



- ***Codeur absolu ou resolver :*** Les codeurs absolus sont destinés à des applications de contrôle de déplacement et de positionnement d'un mobile par codage. Le disque d'un codeur absolu comporte N pistes (jusqu'à 20, selon les modèles). Comme les codeurs incrémentaux les pistes sont alternativement opaques et transparentes. La résolution d'un tel capteur est de 2^N. Deux types de codes sont utilisés : Le code binaire pur et le code Gray

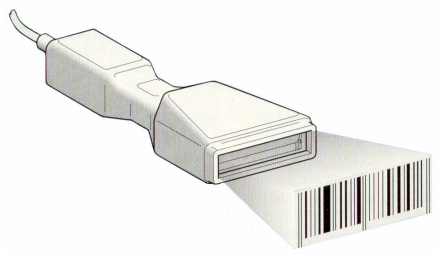


1.8 Les lecteurs de codes à barres

L'identification par codes à barres nécessite deux supports technologiques : le code à barres et le lecteur/décodeur.

Ce type d'identification présente énormément d'avantages :
- Economique : coût limité à celui du support "étiquette"
- Fiable : l'information est présente sur toute la hauteur des barres. Un code même partiellement souillé, détruit ou mal imprimé reste généralement lisible
- Simple à imprimer : les codes peuvent être imprimés sur site utilisateur ou à l'avance
- Souple : grande variété de taille de codes, d'outils d'impression et de lecture.
- Standardisée : il existe pour de nombreux secteurs d'activités une norme de fait qui définit le code à utiliser, sa forme et son contenu.

Les codes à barres sont répartis en deux familles de codes, l'une dites "alimentaires" (code est composé de barres et d'espace dont la largeur peut varier dans un rapport de 1 à 3) et l'autre "industrielle"(barres larges représentent les "1" logiques, et les minces représentent les "0" logiques).



Le lecteur/décodeur se présente sous deux modèles différents :
- le type douchette (compact, léger...)
- le type poste fixe (adapté à la production et à l'automatisation).

2 Périphériques de sortie

Pour rappel, dans un système automatisé, différentes fonctions doivent permettre de :

- **Détecter** par des capteurs les informations issues du procédé physique,
- **Actionner** par de l'équipement électrique, électronique, pneumatique et hydraulique …
- **Echanger** des informations avec l'utilisateur au travers de dispositifs de dialogue.

Les pré-actionneurs sont des interfaces de puissance entre la Partie Commande et la Partie Opérative. Ils permettent d'adapter la nature ou le niveau des énergies de commande et de puissance.
Partie Commande de l'API en très basse tension, 24Volts continu (sécurité) et Partie Opérative 400Volts triphasée (moteur de forte puissance). Pour l'alimentation en énergie des moteurs électriques les pré-actionneurs sont appelés : contacteurs
Partie Commande électrique et Partie Opérative pneumatique (vérins). Pour l'alimentation en énergie des vérins pneumatique ou hydrauliques les pré-actionneurs sont appelés : distributeurs.

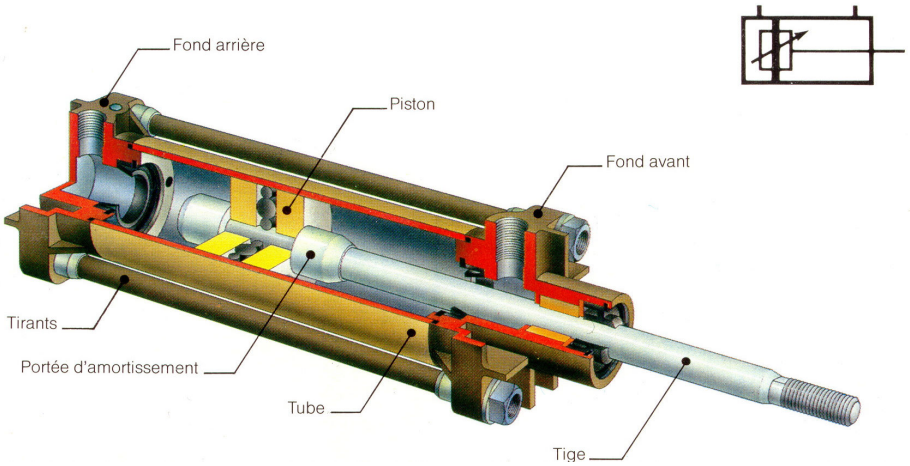
2.1 La commande d'actionneurs pneumatiques :

Avantage des dispositifs pneumatiques :
- Installations peu coûteuses
- Capteurs de précision très pointue et sans contact
- Vitesses de déplacements élevées des vérins.

Inconvénients des dispositifs pneumatiques :
- excellent conditionnement requis : aucune poussière
- compressibilité de l'air ne permet pas des régulations de vitesses régulières
- forces développées restent relativement faibles
- système est bruyant même si les "silencieux" l'atténuent fortement

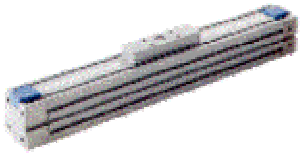
Depuis les années 1980 - 1985, les cadences de production et la flexibilité croissante des installations automatisées favorisent les réalisations électro-pneumatiques.

2.2 Les vérins

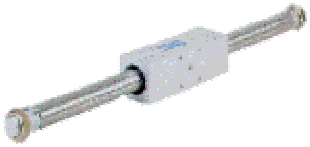


Il n'existe pas un mais plusieurs actionneurs pneumatiques : linéaires ou rotatifs, sans tige ou à double tige, moteurs pneumatiques, …

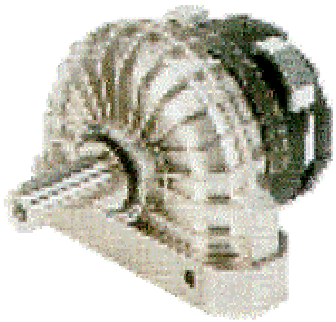
Vérins sans tige



Vérins double tige

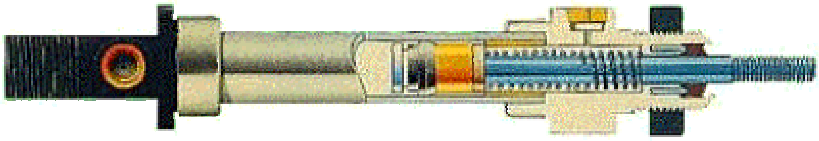


Vérins rotatifs



On les regroupe en deux familles :

- ***Les vérins simple effet***

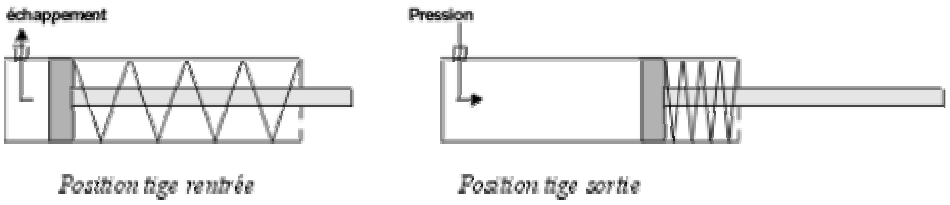


Le vérin simple effet ne peut être alimenté que par une seule arrivée, c'est généralement la chambre arrière.

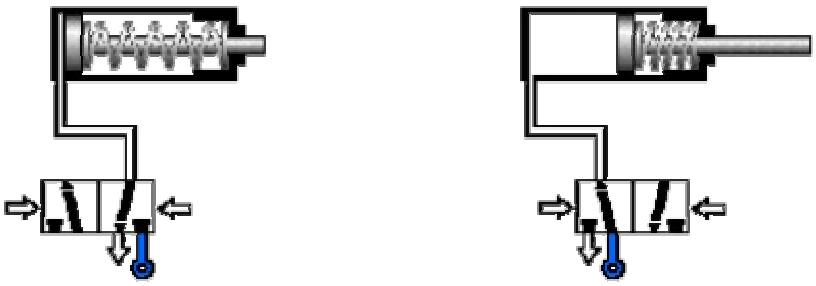
Lorsque l'on cesse d'alimenter en pression cette chambre, le retour s'effectue sous l'action d'un ressort situé dans la chambre opposée.

Celui-ci ne possède donc qu'une seule position stable.

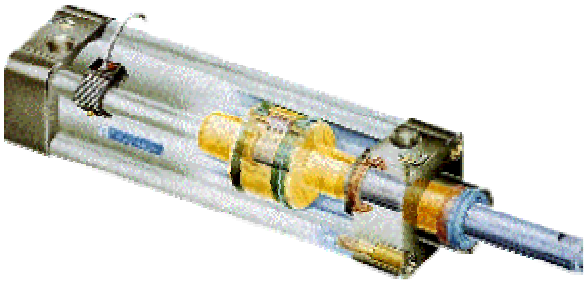
La chambre contenant le ressort est ouverte à l'air libre afin de ne pas contrarier le déplacement du piston.



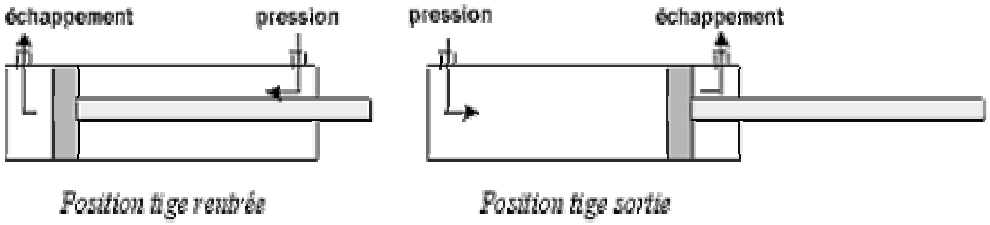
L'alimentation d'un vérin simple effet est obtenue à l'aide d'un distributeur 3/2.



- ***Les vérins double effet***



Le vérin double effet a deux alimentations possibles: soit par la chambre arrière, soit par la chambre avant.



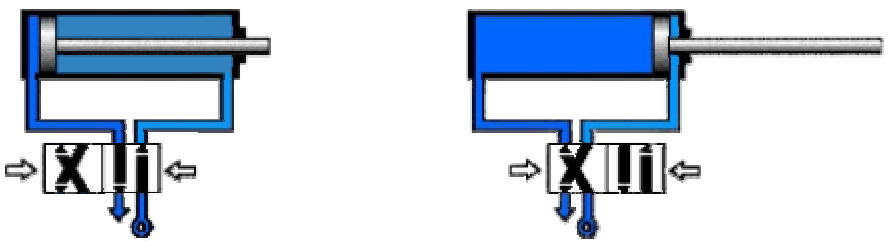
Lors de l'alimentation en pression de la chambre arrière le piston se déplace vers l'avant, celui-ci pousse l'air de la chambre avant (échappement) : la tige du vérin sort A+.

Lors de l'alimentation en pression de la chambre avant le piston se déplace vers l'arrière, celui-ci pousse l'air de la chambre arrière : la tige du vérin rentre A-.

L'air de la chambre à l'échappement doit pouvoir être évacué afin de ne pas s'opposer au déplacement du piston.

Dans un vérin double effet les chambres se trouvent donc alternativement mises à la pression et à l'échappement.

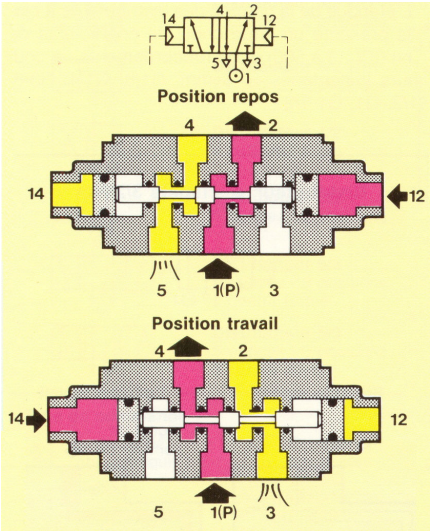
L'alimentation d'un vérin double effet est obtenue à l'aide d'un distributeur 4/2, 5/2 ou 5/3.



2.3 Les distributeurs

Le fonctionnement des vérins impose la possibilité, dans une même chambre et alternativement d'admettre de l'air sous pression et de réaliser une mise à l'air libre. Ce rôle est assuré par des organes spécifiques appelés distributeurs.

La partie mobile du distributeur appelée "tiroir" peut prendre plusieurs positions dont **une seule est active à la fois**. Il est poussé soit à gauche soit à droite suivant une force appliquée aux orifices 12 et 14. Les composants qui fournissent cette force sont appelés pilotes.

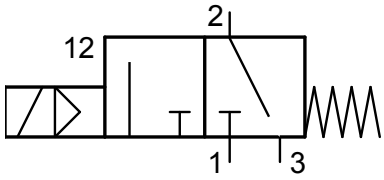


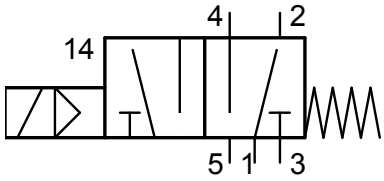
En présence d'un Vérin simple effet (VSE) au minimum 3 orifices suffisent, un pour la chambre du vérin, un pour la pression et un pour l'échappement.

En présence d'un Vérin double effet (VDE) nous avons 2 chambres à alimenter alternativement ce qui impose au minimum 2 positions. Il faut donc disposer d'un distributeur permettant de faire communiquer une chambre avec la pression et l'autre avec l'échappement ; ce qui nécessite au minimum 4 orifices, un pour chaque chambre, un pour la pression et un pour l'échappement.

Les distributeurs sont classés en trois familles :

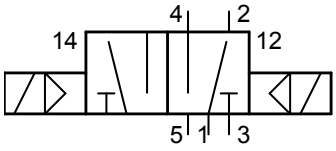
- ***Les distributeurs "monostables"***
  Comme son nom l'indique, il ne possède qu'une position stable. Un ressort de rappel le positionne dans la position de repos. (exemples : 3/2 et 5/2 à commande életro-pneumatique et rappel par ressort)





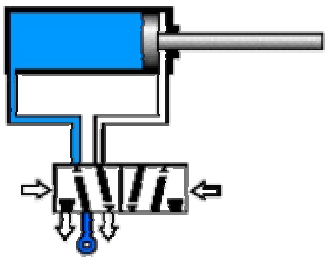
- ***Les distributeurs bistables***

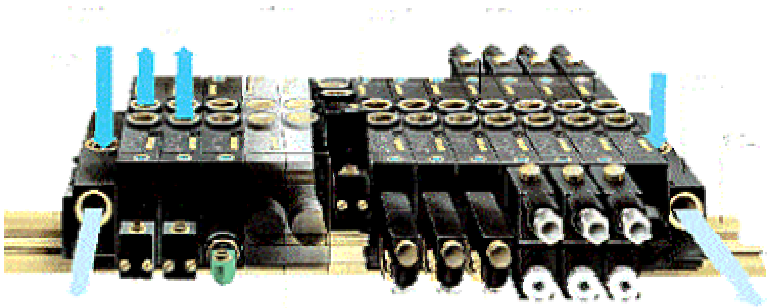
Après la disparition du signal de pilotage sur 12 ou 14, le distributeur (tiroir) conserve la position acquise. Il faut un nouvel ordre pour le faire bouger à nouveau. (Exemple : 5/2 avec pilotes électro-pneumatiques)

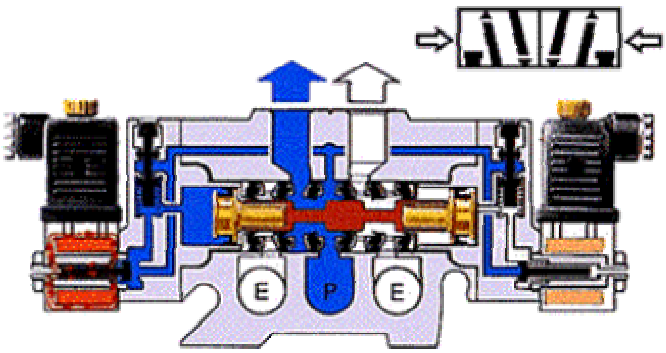


Utilisation d'un distributeur 5/2 (5 orifices / 2 positions)

Le distributeur 5/2 est le plus utilisé actuellement, car son prix est moins important que le 4/2, il est possible d'associer plusieurs distributeurs afin de regrouper les orifices d'alimentation et d'échappement, leur encombrement est plus réduit que le 4/2 pour un diamètre de passage identique.





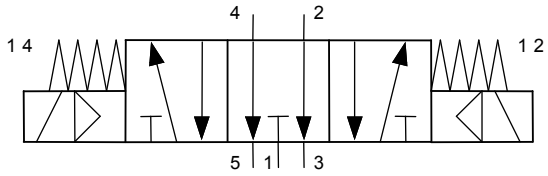


- ***Les distributeurs "multistables"***

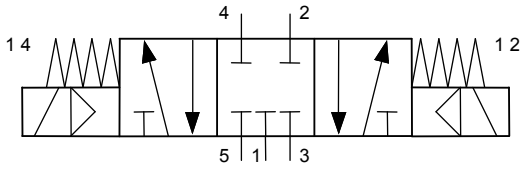
Ces distributeurs généralement à trois positions permettent aux actionneurs d'occuper sur une même trajectoire plus de deux positions.

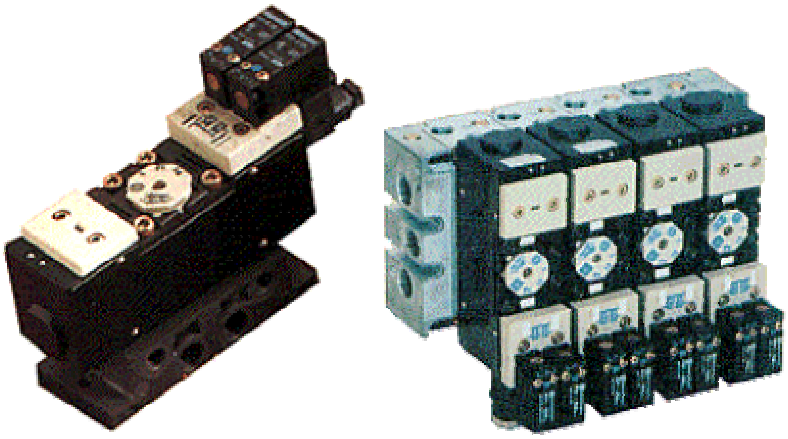
Le distributeur 5/3 est représenté par un symbole proche du 5/2 avec une position centrale supplémentaire :

Symbole 5/3 centre ouvert : Pour des applications particulières on peut utiliser un distributeur **5/3 à centre ouvert** pour alimenter un Vérin double effet. Ce distributeur possède une position centrale repos qui permet de relier les deux chambres du vérin à l'échappement.



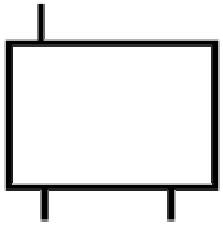
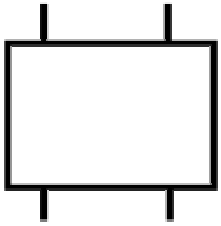
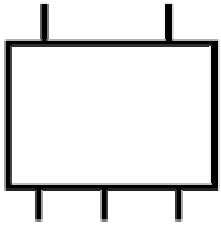
Symbole 5/3 centre fermé : Pour des applications particulières on peut utiliser un distributeur **5/3 à centre fermé** pour alimenter un Vérin double effet. Ce distributeur possède une position centrale repos qui permet de fermer les deux chambres du vérin ce qui a pour effet de bloquer le vérin dans sa position.





2.3.1 Construction du symbole de distributeur

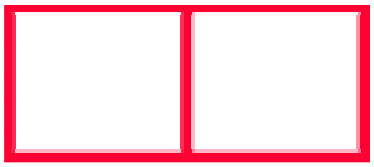
On représente le **corps du distributeur** avec ses orifices à l'aide d'un **rectangle** et de **tirets**.

		
Distributeur à 3 orifices	**Distributeur à 4 orifices**	**Distributeur à 5 orifices**

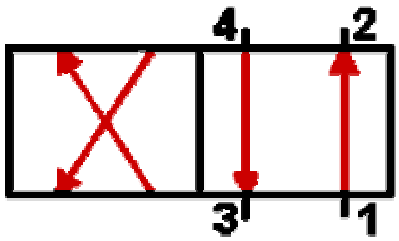
Les orifices d'un distributeur sont repérés par des **chiffres** :

- **1** pour la pression
- **3** pour le premier échappement et **5** pour le second
- **2** et **4** pour les orifices d'utilisation

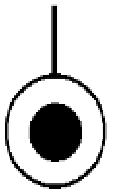
On représente le **nombre de positions** avec des **rectangles** (1 par position)

	
Distributeur à 2 positions	**Distributeur à 3 positions**

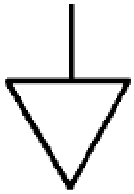
Des flèches viennent compléter ces cases afin de représenter les voies de communication entre les orifices d'alimentation (pression et échappement) vers les orifices d'utilisation (vers vérin).



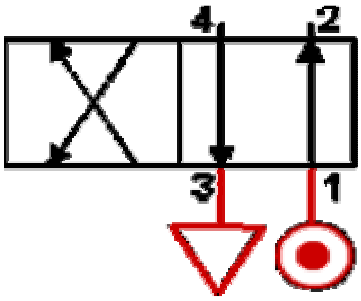
On représente l'alimentation en air comprimé et les échappements de la manière suivante:



Alimentation air comprimé



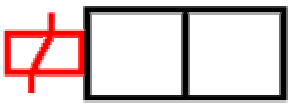
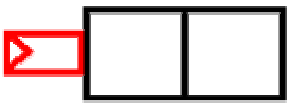
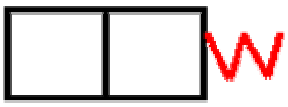
Echappement



Ces deux éléments sont à dessiner du coté de **la position active au repos**

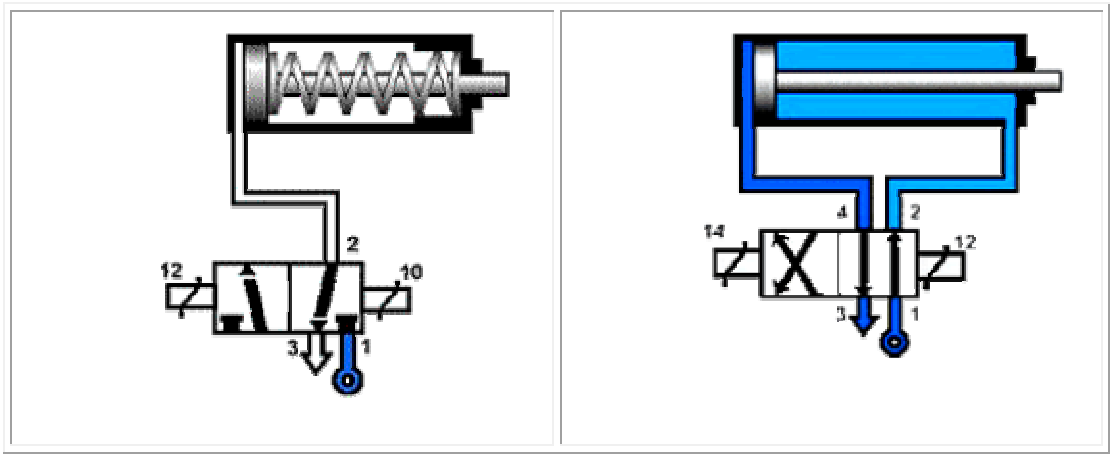
Commande des distributeurs :

L'énergie de commandes des distributeurs permettant le déplacement du tiroir peut être électrique, pneumatique, mécanique, ...combinée.
Les éléments de commande des distributeurs, appelés **pilotes** sont représentés par:

Symbole			
Énergie de commande	Électrique par électro-aimant	Pneumatique	Mécanique (ressort) monostable

Repérage des pilotes :

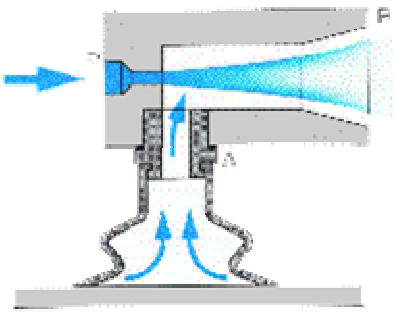
Les pilotes sont repérés par deux chiffres identifiant la communication entre l'orifice d'alimentation 1 et l'orifice d'utilisation.
Le pilote permettant d'alimenter l'orifice 2 est repéré 12 et le pilote permettant d'alimenter l'orifice 4 est lui repéré 14.
Dans le cas d'un distributeur 3/2 bistable une des deux positions interdit la communication de l'alimentation, dans ce cas le pilote correspondant est repéré 10.



Pour dessiner un symbole de distributeur il faut donc connaître: le nombre d'orifices et de positions ainsi que les énergies de commande et de puissance.

2.4 Autres actionneurs pneumatiques

Préhension par le vide : Très utilisée dans la manipulation d'objets. La préhension par le vide, basée sur le principe de l'effet venturi est la plus couramment utilisée. Elle se compose d'un éjecteur pneumatique associé à une ou plusieurs ventouses. Ces ventouses sont des éléments de préhension souples destinés à être utilisés avec un générateur de vide. De matière, de forme et de diamètre différents elles permettent de répondre pratiquement à tous les cas d'application de manutention.



3 Pupitre opérateur

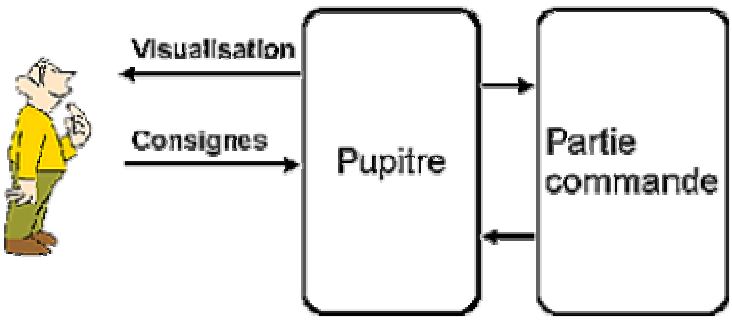
Pour rappel, dans un système automatisé, différents fonctions doivent permettre de :

- **Détecter** par des capteurs les informations issues du procédé physique,
- **Actionner** par de l'équipement électrique, électronique, pneumatique et hydraulique …
- **Echanger** des informations avec l'utilisateur au travers de dispositifs de dialogue.

Un système automatisé a pour vocation de fonctionner sans intervention humaine. Cependant à différents moments de son fonctionnement, il doit pouvoir échanger des informations avec son utilisateur. Le pupitre est un des constituants de la fonction « dialoguer ».

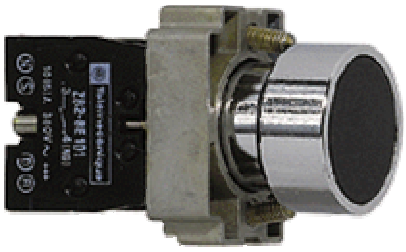
Ces informations sont bidirectionnelles:

- Visualisation d'informations vers l'utilisateur
- Consignes vers le système.



3.1 Emission de consigne

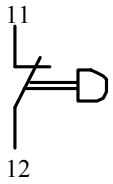
3.1.1 Les boutons poussoirs



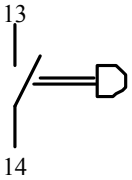
Les boutons poussoirs permettent de transmettre une information binaire.
La couleur doit respecter la norme.

Ils sont de deux types :

- NF : Normalement Fermé :
  enfoncer ce type de BP ouvre le circuit, fermé à l'état de repos
  Conventionnellement, un BP NF est rouge



- NO : Normalement Ouvert :
  enfoncer ce type de BP ferme le circuit, ouvert à l'état de repos
  conventionnellement, un BP NO est noir



3.1.2 Les boutons coup de poing

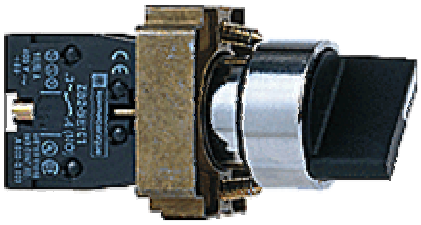
Il existe des boutons poussoirs d'arrêt d'urgence « coup de poing » (norme : jaune sur fond rouge) Lors de leur usage pour Arrêt d'urgence, ils doivent êtres à accrochage, ils peuvent alors êtres déverrouillés à l'aide d'une clef.

3.1.3 Les Switchs

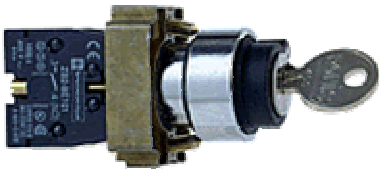
Il s'agit d'un commutateur à deux positions mécaniquement distinctes, chaque position sera connectée électriquement à un niveau de tension relatif soit au niveau logique « 1 », soit au niveau logique « 0 ».

3.1.4 Les Sélecteurs

Ils peuvent compter 2, 3, 4, 5... positions



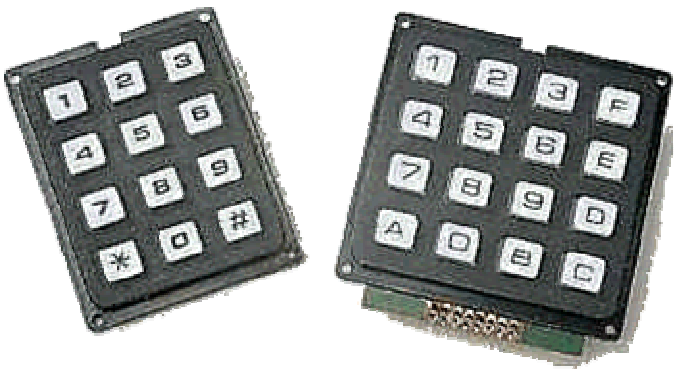
Certains sont munis d'une clé :



3.1.5 Les claviers

- ***Les roues codeuses :*** D'une utilisation limitée elles permettent de communiquer des valeurs numériques limitées (valeur de temporisation, de comptage...).

- ***Les claviers :*** Ils peuvent êtres numériques (12 touches) ou alphanumériques (une centaine de touches)



3.2 Les organes de visualisation

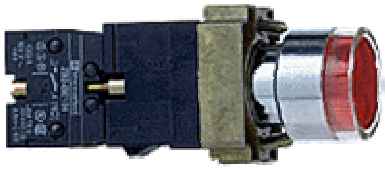
- ***Les voyants :*** L'utilisation de voyants doit respecter la norme des couleurs

- ***Les colonnes ou Verrines*** : Utilisées sur les machines pour visualisation. Un code couleur est défini pour signaler les modes de marche et les défauts.

- ***Les afficheurs :*** Pour les nombres, il existe les afficheurs 7 segments, les afficheurs LCD ainsi que pour les lettres. Pour les graphiques, on retrouve les écrans couleurs ou monochromes.

3.3 Les combinés I/O

- ***Les boutons lumineux :*** Un bouton poussoir et un voyant réunis. Respecter la norme des couleurs.



- ***Les sélecteurs lumineux :*** Un sélecteur et un voyant réunis. Respecter la norme des couleurs.

- ***Les terminaux d'exploitation texte :*** De plus en plus courant sur les machines basiques. Un afficheur texte est complété par un clavier alphanumérique

- ***Les terminaux d'exploitation graphiques :*** Un écran graphique complété d'un clavier alphanumérique. Présent sur les machines complexes (Commandes numériques, supervision de process...)

3.4 La norme des couleurs

(selon NF EN 60204-1 (VDE0013, partie1) : 06.93)

3.4.1 Code de couleur pour organe de commande à BP

Couleur	Signification	Utilisation	Exemple
rouge	URGENCE	Actionnement dans une situation dangereuse ou d'urgence	arrêt d'urgence
jaune	ANOMALIE	Actionnement en cas d'anomalie	intervention permettant d'éliminer un état anormal ou permettant de redémarrer un cycle automatique interrompu
vert	SECURITE	Actionnement dans des conditions sûres ou préparation d'un état normal	Marche ou mise sous tension
bleu	OBLIGATION	Actionnement dans un état nécessitant une action obligatoire	fonction de réarmement
blanc, gris, noir	SANS SIGNIFICATION SPECIFIQUE		

3.4.2 Code de couleur des voyants lumineux de signalisation

Couleur	Signification	Utilisation
rouge	URGENCE	Situation dangereuse
jaune	ANOMALIE	Etat anormal supposant un état critique imminent
vert	NORMALITE	Etat normal
bleu	OBLIGATION	Signalisation d'un état nécessitant une action de l'opérateur
blanc	NEUTRE	Surveillance

Partie commande

4 Les automates programmables industriels : API

L'API [2] est un dispositif électronique de contrôle de processus. Il génère des ordres vers les préactionneurs de la partie opérative à partir de données d'entrées (capteurs) et d'un programme. De par sa souplesse d'utilisation, son encombrement réduit et ses possibilités d'adaptation à tous les milieux industriels, l'automate programmable industriel est devenu l'outil privilégié de la commande des Systèmes Automatisé de Production (SAP). Rapidement, il a pris le dessus sur les registres séquenceurs pneumatiques ou électriques, qui de ce fait ne sont plus utilisés que dans des cas très précis et particuliers. Aujourd'hui, tous les secteurs industriels utilisent des automates.

Citons :

- l'industrie pharmaceutique et alimentaire ;
- l'industrie automobile ;
- le secteur de la transformation des matières plastiques ;
- les systèmes de conditionnement de produits divers

Ce système de commande électronique permet le travail en ambiances sévères [3]. Il offre également la possibilité de faire évoluer le processus si le système automatisé l'impose.

Le secteur de la construction mécanique est aujourd'hui confronté aux exigences nouvelles que sont la production par petits lots et la diversification des produits. Pour assurer la rentabilité, il existe deux parades : accroissement de la flexibilité et diminution des temps de cycle.

De nombreux constructeurs d'entraînement pneumatiques et de systèmes de commande se sont intéressés à cette nouvelle demande. Un grand nombre d'entre-eux propose aux utilisateurs des composants et des systèmes d'automatisation adaptés dans les principaux domaines : actionneurs, capteurs, commandes.

On peut ainsi envisager des réalisations loin dans le futur. Les automates programmables devant assumer de plus en plus de tâches, il est important de mettre l'accent sur la flexibilité. Pour cela, il est indispensable de pouvoir compter sur un guidage opérateur performant qui permettra entre autres de réduire au maximum les temps d'immobilisation et de disposer d'un bon système de diagnostic machine. Les positionnements par moteurs électriques, la communication entre les automates et les liaisons par câbles deux fils font en outre partie intégrante du concept de commande du futur.

---

2 Encore appelé P.L.C. *(Programmable Logic Controler)* ou S.P.S. *(Speicher Programmierbare Steuerung)*

3 L'API est capable de travailler en ambiance industrielle extrêmement sévère: humidité 90%; température 60°C (et même 90°C avec circuit de réfrigération); protection poussière-eau IP65 ; insensible aux chocs et vibrations: variations de tension d'alimentation +20 à -10% du 220V....
Il a surtout une immunité importante aux parasites industriels !

Des nécessités économiques exigent un effort de clarification des méthodologies d'étude et de mise en œuvre. Une méthodologie aboutissant à la réalisation technologique peut être définie par :

- l'établissement du cahier des charges aboutissant au tracé du grafcet fonctionnel «point de vue partie opératoire» (niveau 1)
- l'analyse opérationnelle et technologique aboutissant au tracé du grafcet technologique «point de vue partie commande» (niveau 2)
- l'analyse des modes de marches et d'arrêts (GEMMA)
- le choix de la technologie de commande
- la synthèse de la partie commande (tracé du schéma de principe),
- le câblage de l'armoire de commande.

4.1 La logique programmée

Un automate programmable est conçu autour d'un calculateur logique, ou ordinateur, au jeu d'instructions volontairement réduit, destiné industriellement à la gestion de processus séquentiels ou combinatoires. Les automates programmables qui remplacent progressivement les équipements câblés, permettent, grâce à leur structure, de modifier simplement les séquences d'un processus géré sans contrainte de câblage. Leur emploi est simple, et un personnel familiarisé aux automatismes s'y adapte rapidement.

Il se compose de plusieurs parties et notamment d'une mémoire programmable dans laquelle l'opérateur écrit, dans un langage d'application propre à l'automate, des directives concernant le déroulement du processus à automatiser.

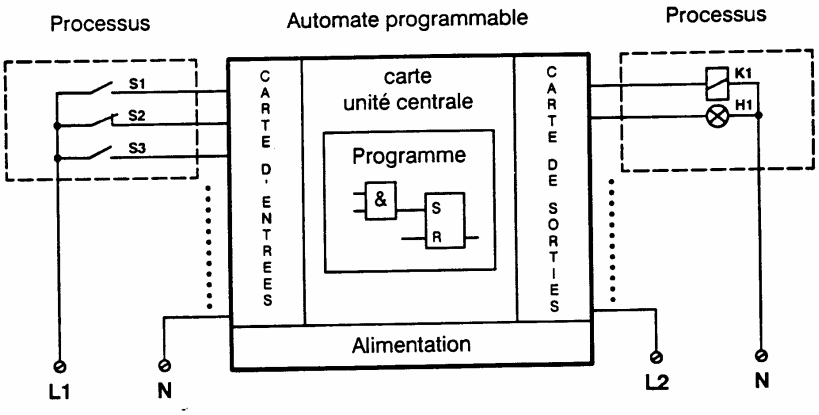
Son rôle consiste donc à fournir des ordres à la partie opérative en vue d'exécuter un travail précis, comme par exemple, la sortie ou la rentrée d'une tige de vérin. Celle-ci, en retour, lui donnera des informations relatives à l'exécution dudit travail.

Les API sont apparus aux Etats-Unis vers 1969 où 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 complexes, les coûts de câblage et de mise au point devenaient trop élevés : dans un automatisme programmé, une modification du système correspond à une modification du programme en mémoire mais pas à une modification de câblage.

4.2 Notions d'architecture

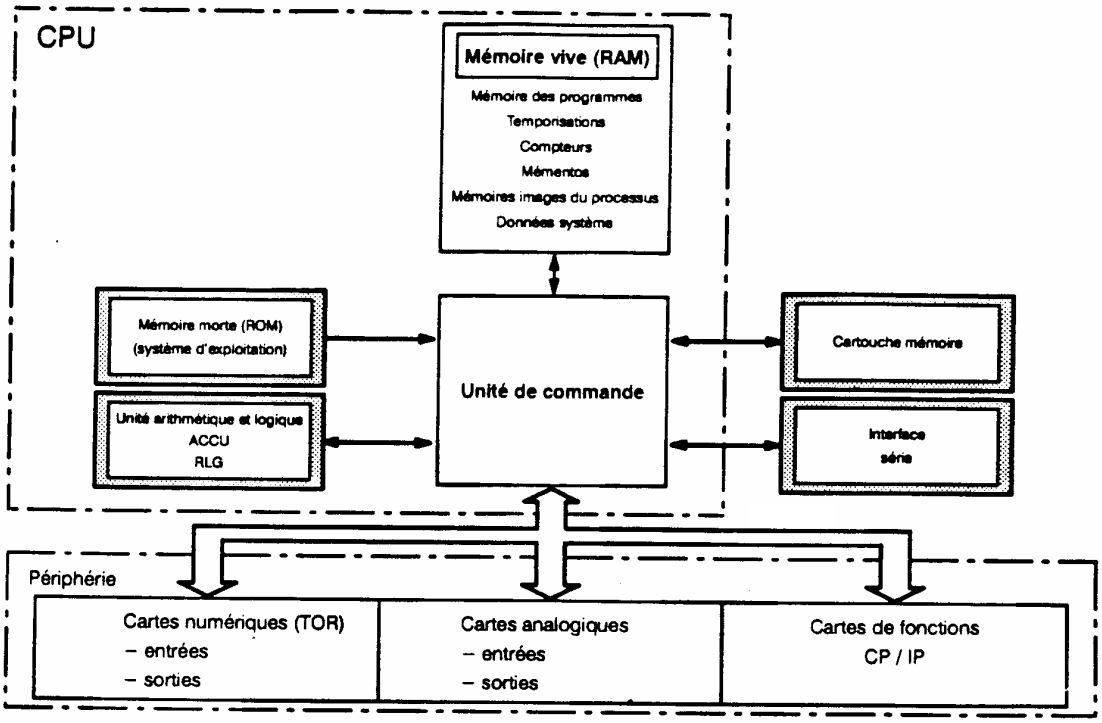
Comment s'effectue le transfert d'information dans l'automate ?



L'automate possède des «interfaces» d'entrée et de sortie.

1. La détection : les signaux électriques en provenance des capteurs arrivent sur les modules d'entrée.
2. Chaque module est repéré par une adresse d'entrée.
3. L'information est traitée par le processeur qui renvoie un signal sur l'interface de sortie.
4. L'interface possède des adresses de sortie, le signal est donc renvoyé vers le préactionneur qui commandera l'actionneur.

Les automates programmables comportent quatre parties principales :



- un processeur (unité centrale)
- une mémoire
- des cartes et des interfaces d'E/S ;
- une alimentation (220 V – 24 V).

Ces quatre parties sont reliées entre elles par des «bus» (ensemble de fils autorisant le passage des informations entre ces quatre secteurs de l'automate). Ces quatre parties réunies forment un ensemble compact appelé «automate».

La plupart des automates actuels se présentent sous forme modulaire.

4.2.1 Une alimentation électrique

Tous les automates actuels utilisent la tension 24 V. Un module d'alimentation 240 VAC fournit une tension réduite de 24 VDC pour les modules d'E/S ainsi que la tension nécessaire pour sa gestion interne. Le module d'alimentation contient une batterie pour garder l'alimentation sur la mémoire RAM.

La tension d'alimentation des capteurs, actionneurs ou autre signalisation (24 à 220 V) doit être fournie par une alimentation externe. Une mise à la terre doit également être prévue.

4.2.2 L'unité centrale : CPU

La CPU est le cœur de l'automate, elle comprend le processeur et les mémoires qui contiennent le logiciel système, le programme utilisateur ainsi que les données.

4.2.2.1 Le processeur

Partie intelligente de l'automate, son rôle consiste d'une part à organiser les différentes relations, entre la zone mémoire et les interfaces d'entrée et de sortie et d'autre part à gérer les instructions du programme. Il lit les états des capteurs sur les entrées, les combine suivant les instructions du programme utilisateur et fournit les tensions aux sorties pour permettre l'activation des actionneurs.

Il contient une unité arithmétique avec des accumulateurs, des registres, des temporisateurs et des compteurs.

Un compteur de programme pointe les instructions successives à exécuter lors du déroulement du programme.

4.2.2.2 La mémoire

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système qui sont :

- le terminal de programmation : introduction du programme utilisateur;
- le processeur qui gère et exécute le programme, l'introduction des informations.

Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates plusieurs types de mémoires qui remplissent des fonctions différentes.

- Système d'exploitation : mémoire ROM.
  Le système d'exploitation contient les programmes « système » nécessaires à l'exécution du programme d'application, à la gestion des E/S, à l'organisation de la mémoire, à la gestion des données, etc. Le système d'exploitation est fixé par le fabricant et ne peut être modifié.
- Sauvegarde des données : mémoire RAM.
  Cette mémoire contient une zone pour la sauvegarde des données du processus, valeurs de temporisation et de comptage et pour la sauvegarde des résultats intermédiaires (flag, registres, mémentos,…). Une autre zone est réservée pour ranger les paramètres système. Et enfin, suivant le type d'automate, une zone de cette mémoire RAM est utilisée pour écrire les états des signaux des modules d'entrées et ceux destinés aux modules de sorties.
- Conception et élaboration du programme.
  - Mémoire RAM : modifiable à volonté, elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).
  - Mémoire (E)EPROM : seule la lecture est possible.
- Conservation du programme pendant l'exécution de celui-ci. Mémoire (E)EPROM. La programmation doit se faire selon un procédé qui n'est pas directement accessible sur l'automate.

4.2.3 Les cartes et les interfaces d'E/S

Les modules d'entrées/sorties permettent de relier l'unité centrale à l'installation à commander. Ces modules servent à la mise en forme des signaux, nécessaires au passage de l'information entre les niveaux de tension existant dans l'installation et la tension interne.

La structure d'un système automatisé fait apparaître 3 parties essentielles qui sont :

- la partie opérative,
- la partie commande,
- l'opérateur.

Ces différents éléments, doivent échanger des informations logiques. Or, la nature des signaux échangés, n'est pas toujours compatible avec la «technologie» de la partie destinataire. La partie commande d'un système automatisé, traite des signaux logiques, électriques ou pneumatiques, de qualités et de niveaux bien définis.

Des interfaces d'entrée transforment les informations en provenance de l'extérieur, en signaux adaptés à la partie commande. Inversement, les signaux en provenance de la partie commande doivent être adaptés au dialogue avec l'opérateur (voyants, afficheurs, etc…) ou à la nature et à la puissance nécessaires à l'alimentation des actionneurs.

Les interfaces transforment ou adaptent les signaux qui transitent entre la partie commande et sa périphérie. [4]

L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon des adresses de sortie. Chaque préactionneur est relié à une de ces adresses.

Le nombre de ces entrées et sorties varie suivant le type d'automate. Des modules analogiques ou digitaux seront utilisés suivant les besoins.

4.2.4 Modules spécialisés.

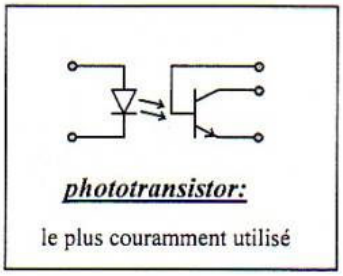
Pour éviter de surcharger l'unité centrale de l'API, des modules spécialisés communiquant avec le CPU sont utilisés pour gérer des tâches bien spécifiques pour lesquelles ils sont optimisés. Ces modules comportent leurs propres CPU ou contrôleurs :

- Cartes de communications
- Modules de régulation
- Modules de comptage rapide
- Modules de positionnement (moteurs pas à pas , DC, AC…)
- Modules de dosage
- Modules de visualisation
- Modules de diagnostic
- …

4.2.5 Le système de bus.

La CPU et les modules d'E/S sont reliés à un bus servant aux échanges de données. Il s'agit souvent d'un bus parallèle permettant l'échange de plusieurs informations simultanément. Il comprend les parties suivantes :

- Bus de données : permet la saisie de l'état des signaux des modules d'entrées ainsi que la distribution des signaux aux modules de sorties.
- Bus d'adresses : transporte les adresses des modules d'entrées et de sorties à traiter.
- Bus de commande : transmet les signaux pour la commande et surveillance du déroulement des fonctions de l'automate.



[4] Les Entrées-Sorties de l'API sont isolées totalement du monde extérieur.

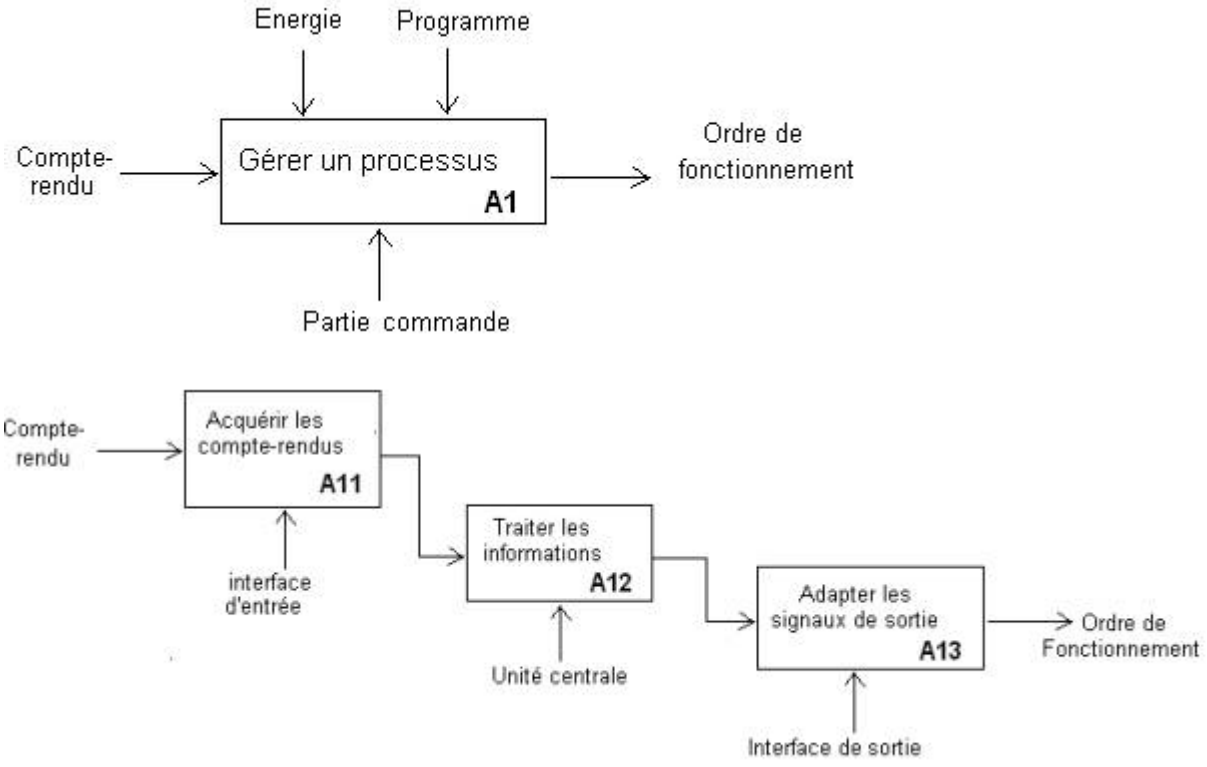
a) L'isolement E-S par opto-coupleur peut atteindre plusieurs kilovolts. Vous vous souvenez *bien entendu* ☺ avoir vu en électronique IG2 qu'un coupleur optique ou Optocoupler (Optoisolator) est constitué :
   - d'un émetteur (LED)
   - d'un récepteur (photodiode, phototransistor, ...)
   - éventuellement de circuits spécifiques dans un même boîtier.

b) L'isolation peut être aussi réalisée à partir de modulateur / démodulateur par transformateur.

c) L'isolation par capacités commutées a tendance à se développer...

Les circuits d'entrée assurent également une mise en forme du signal (dispositifs à seuils, trigger de Schmitt pour les signaux logiques, redressement pour les signaux alternatifs, écrêteurs...). L'immunité aux parasites est assurée par des filtres: le signal doit être présent en entrée au moins pendant 20 ms pour être acceptée.

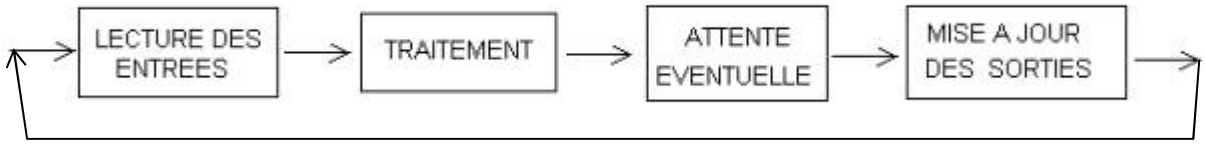
4.3 Mise en œuvre d'un API : Analyse Fonctionnelle



1) Etablir :
   - le schéma à contact ou ladder
   - ou le logigramme ou les équations logiques
   - ou le GRAFCET
2) Ecrire le programme dans la mémoire programme (pupitre de commande) puis le simuler si possible
3) Transférer le programme dans l'unité centrale de l'automate
4) Tester à vide: mise au point finale du programme
5) Raccorder l'automate à la machine

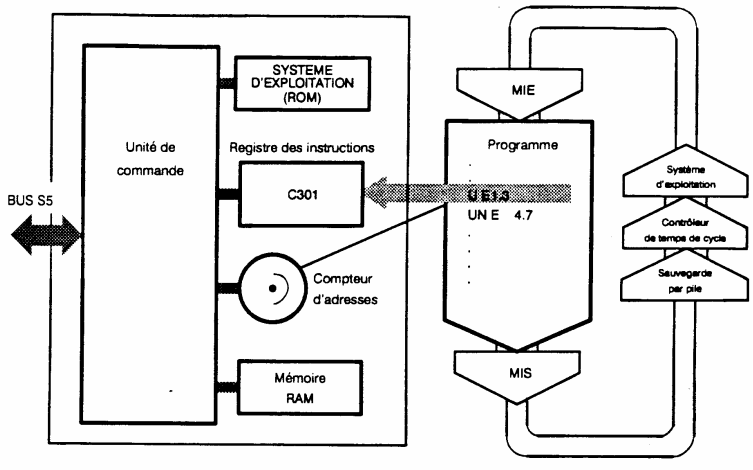
4.4 Fonctionnement : Boucle sans fin

Le fonctionnement d'un API diffère de celui d'un ordinateur. En effet l'API traite le programme en boucle sans fin : fonctionnement cyclique
En début de cycle, le système fait l'acquisition des états logiques des signaux sur l'interface d'entrées. Les entrées sont figées dans la mémoire image des entrées pour tous les calculs d'un même cycle (même si le signal physiquement connecté à l'entrée varie au cours du cycle).
Le processeur exécute alors le programme instruction par instruction en rangeant chaque fois les résultats en mémoire. En fin de cycle les sorties sont affectées d'un état binaire, par mise en communication avec les mémoires correspondantes.
Lorsque le programme est terminé, il est automatiquement recommencé. []



Ce procédé « synchrone par rapport aux entrées et aux sorties » a l'avantage de raccourcir le temps de cycle car les accès en mémoire sont beaucoup plus rapides que les accès aux cartes. De plus, il présente l'avantage de figer l'état d'une entrée pendant tout le traitement d'un cycle de programme (une modification de l'état en cours de cycle pourrait conduire à des défauts de l'installation) et d'empêcher les sorties d'osciller au cours d'un même cycle. Dans ce cas le temps de réponse à une variation d'état d'une entrée peut être compris entre un et deux temps de cycle.

Le processeur de commande exécute le cycle de traitement des informations à une fréquence élevée pour commander efficacement les sorties *en temps réel.* Le temps de cycle est le temps que met l'automate pour exécuter une fois le programme dans son entièreté. Un temps de cycle acceptable pour un système séquentiel classique est de l'ordre de 5 à 150 ms et dépend du CPU.
Un chien de garde « *watchdog* » vérifie la durée du cycle de traitement et génère un message d'erreur si le temps enveloppe est dépassé.



Traitement cyclique avec mémorisation des E/S

4.5 La programmation

Les instructions décrivant le cycle à exécuter par l'automate seront écrites dans un programme sous la forme d'instructions codées. Ce programme doit pouvoir être écrit, transféré à l'automate, testé, corrigé.

4.5.1 Console de programmation

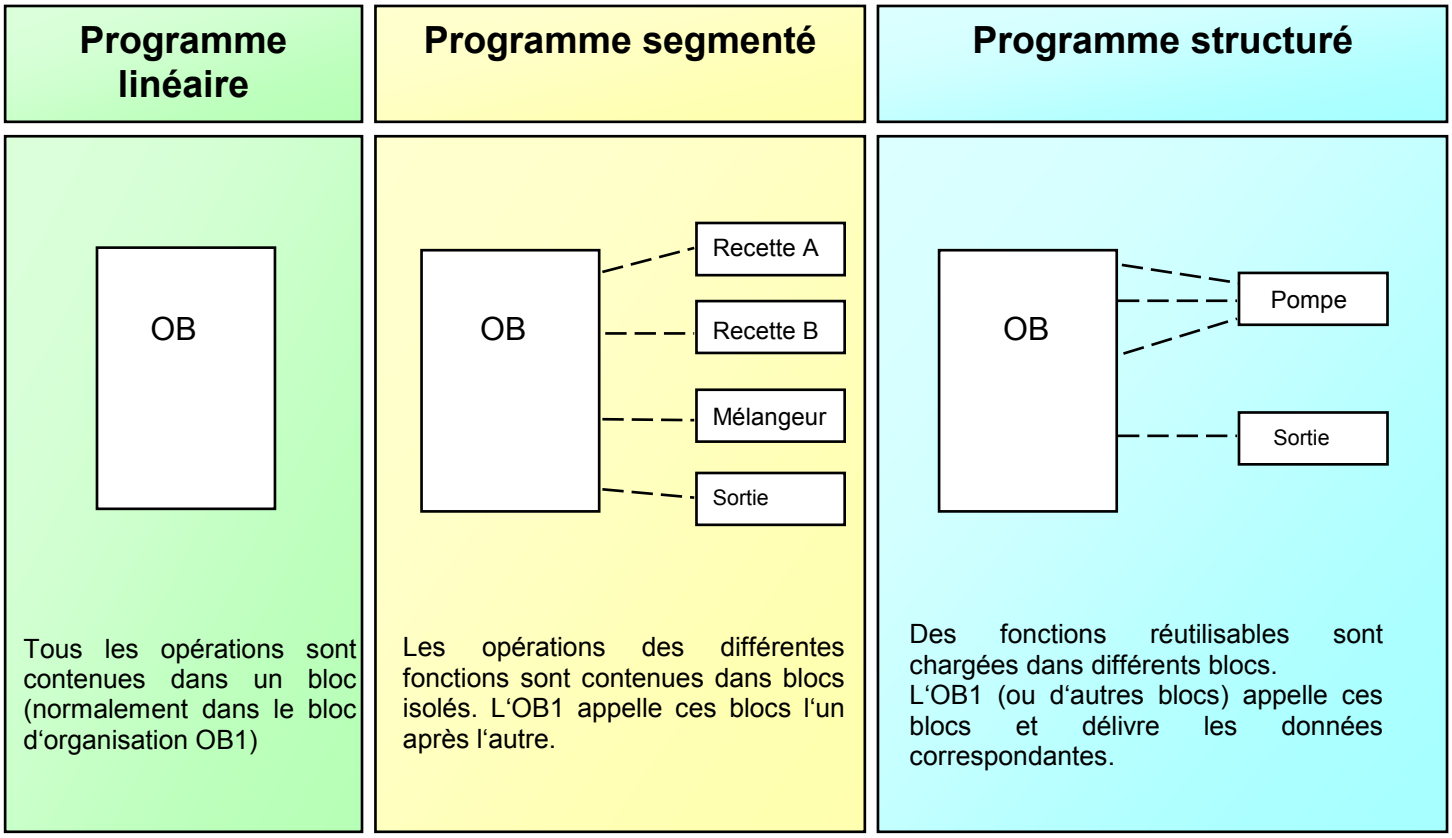
Elle se branche sur l'API par une prise spéciale sur le module CPU.

Son rôle principal consiste à traduire les instructions « utilisateur » du code mnémoniques en instructions machines exécutables par l'automate, mais aussi de permettre le transfert du programme dans l'automate et inversement.

Chaque automate possède son propre langage et donc sa propre console de programmation. Il existe plusieurs types de console de programmation :

- Les consoles portables légères avec clavier et affichage très réduits qui se branchent directement sur l'automate. Le programme est directement écrit dans la mémoire de l'automate (programmation ON-LINE).
- Les consoles avec écran graphique, clavier dédié et disque dur. Ces consoles ont un système d'exploitation qui permet l'exécution du logiciel de programmation de l'automate mais éventuellement aussi d'autres logiciels. Le programme est écrit dans la mémoire de la console (programmation OFF-LINE) et est ensuite transféré dans l'automate. Les programmes peuvent être archivés ce qui présente une certaine sécurité. Ces consoles disposent généralement de l'interface de programmation des modules (E)EPROM qui seront enfichées dans l'automate et qui contiendront les programmes d'application.
- Le *micro-ordinateur de bureau* ou le «portable» : ce moyen de programmation est identique à celui de la console. Le logiciel utilisé, par marque, est le même dans les 2 cas.

4.5.2 Types de programmation



4.5.2.1 Programmation linéaire.

Un programme linéaire est un programme où toutes les instructions sont écrites les unes derrières les autres dans un seul bloc. Il présente l'avantage d'un traitement un peu plus rapide car aucune instruction de saut n'est traitée par le processeur. Il présente par contre très peu de clarté. La programmation linéaire est de plus en plus rarement utilisée.

4.5.2.2 Programmation segmentée

En programmation segmentée, un programme volumineux est divisé en plusieurs parties fonctionnelles bien distinctes permettant de mieux clarifier les fonctions en les séparant. Ces divers modules de programme sont mémorisés sous forme de blocs de programme aussi appelés sous-routines ou sous-programmes. Lors de la mise en service et plus particulièrement du dépannage, cette répartition en blocs présente de nombreux avantages dans la recherche d'erreurs dans le programme.

4.5.2.3 Programmation structurée

La programmation structurée offre la possibilité de paramétrer les blocs. Par exemple, si le processus comporte plusieurs pompes identiques, chaque pompe sera commandée par la même logique mais devra correspondre à son propre adressage de sortie.

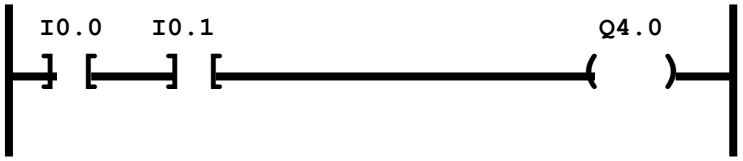
4.5.3 Les langages

La programmation des automates industriels n'est pas, à ce jour, très évidente du fait de divers critères, intervenant au niveau du matériel et du langage de programmation.

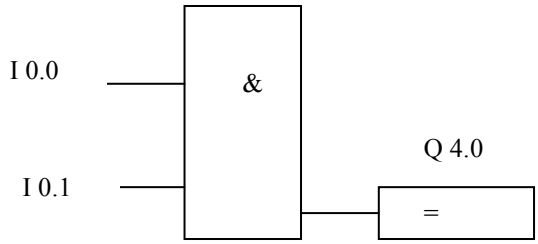
En effet, si les API des grands constructeurs utilisent les mêmes langages, les façons de faire sont différentes. On regroupe les langages suivant le mode de raisonnement suivi pour la programmation. Certains automates proposent l'exploitation de différents langages.

Les langages les plus courants sont :

- le langage à contact ou Ladder diagram : représentation graphique (proche du schéma électrique) utilisant les symboles, usuel aux USA.



- le langage Logique : représentation graphique utilisant un symbole pour chaque fonction. Les entrées sont définies sur la partie gauche du symbole et les sorties sur la droite.



- le langage littéral ou liste d'instruction : langage propre à l'API, instructions sous forme mnémonique qui constitue la traduction « en texte » du programme « en graphique » (et qui permet le codage des instructions « en binaire » dans la mémoire de l'API.

Siemens	Saia	Omron
:A I 0.0	STH I 0	LD 0
:A I 0.1	ANH I 1	AND 1
:= Q 4.0	OUT O 32	OUT 100

Par ailleurs, il existe des outils de programmation « de haut niveau » comme le langage GRAFCET qui offre l'avantage d'être normalisé au niveau de la description, l'implémentation restant propre à l'automate.

4.5.4 Jeu d'instructions

Le processeur peut exécuter un certain nombre d'opérations logiques.

L'ensemble d'instructions booléennes et des instructions complémentaires de gestion de programme (saut, mémorisation, adressage…) constitue un jeu d'instructions.

Sur une ligne de programme, on trouvera systématiquement un code d'instruction, suivi éventuellement de l'adresse de l'opérande (variable) sur laquelle s'applique l'opération.

4.5.4.1 Opérations logiques de base

lecture de l'état d'une variable (load, si, if etc..)

ET logique (quelquefois implicite) (ET, AND, UND, etc…)

OU logique (OU, OR, ODER, +, etc…)

Affectation ou égal (=, SET, OUT, etc…)

PAS ou négation des instructions précédentes (NOT, NON, NICHT, PAS, etc…)

4.5.4.2 Instructions complémentaires

mémorisation (par câblage interne ou programme)

temporisation (par câblage interne ou programme)

comptage (par câblage interne ou programme)

addition, soustraction, multiplication, division

adressage indirect

conversion

saut (avant ou arrière ou les deux)

...

4.5.4.3 Fonctions complémentaires

Fonctions avec paramètres

Traitement des valeurs analogiques

Régulation

Communication

...

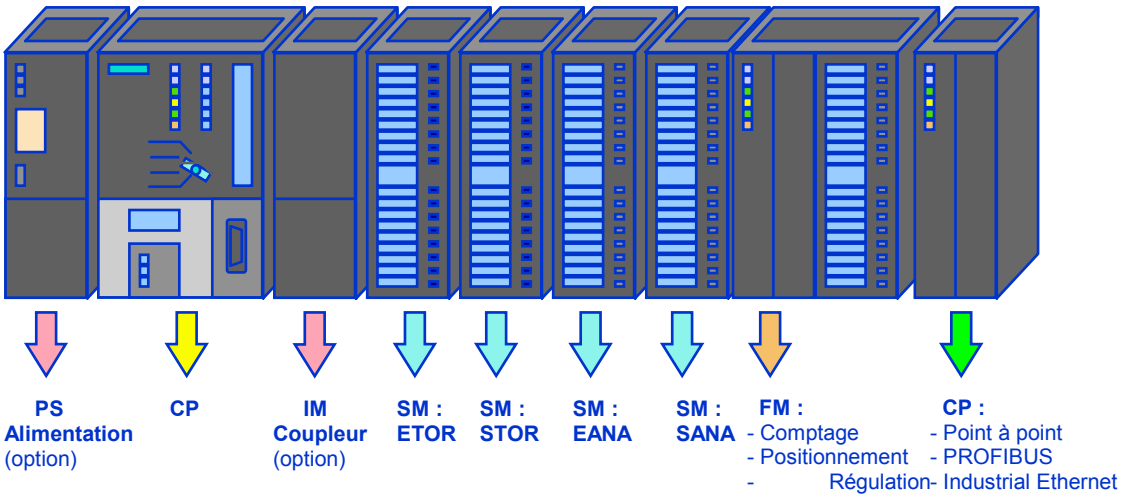
4.6 Principales caractéristiques.

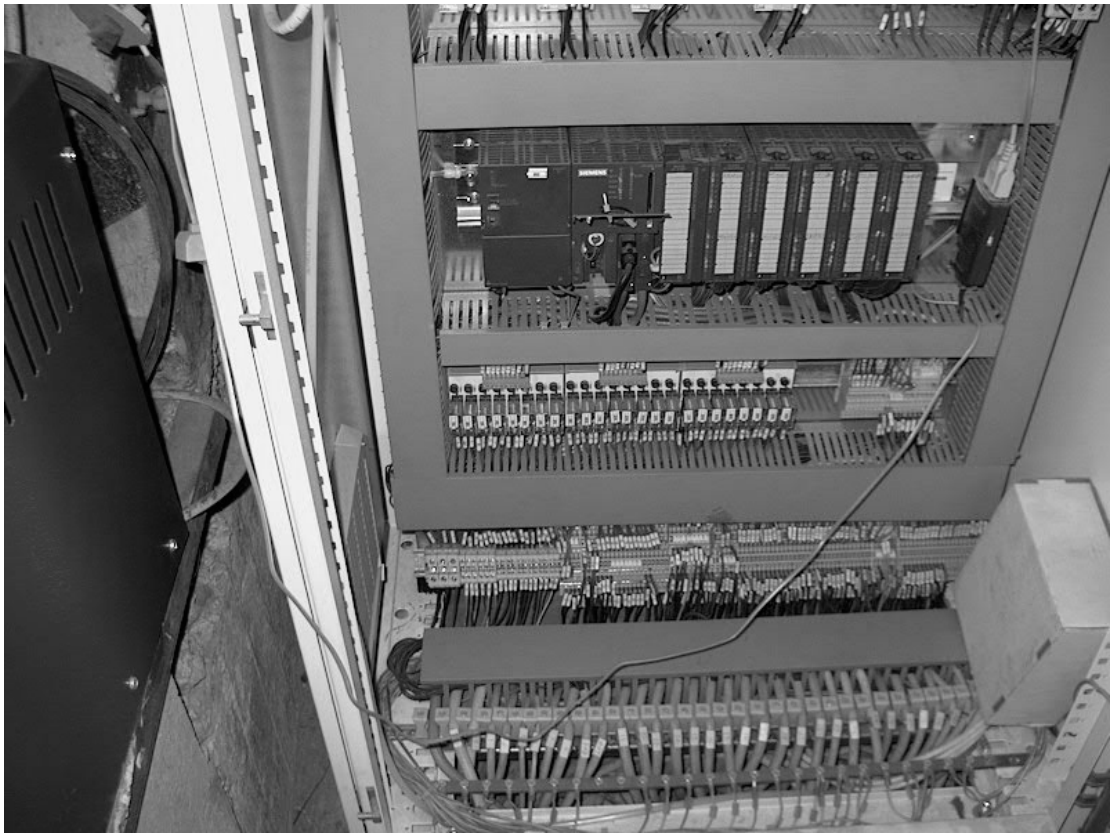
- Grand nombre d'entrées et de sorties parfaitement adaptées au milieu industriel.
- Facilité de programmation, mise en service, dépannage
- Modularité : Investissement progressif
  Extensions possibles si le processus évolue
  Réutilisation possible pour d'autres besoins
  Grand choix de modules spéciaux
- Possibilité de calcul avancé
- Possibilité de communication entre automates, automate – PC, automates – capteurs, actionneurs (bus de terrain)
- Grande gamme de constructeurs et de modèle.

4.7 Critères de choix.

- Nombre d'E/S et leur nature
- Jeu d'instruction
- Vitesse de traitement
- Capacité de traitement
- Capacité mémoire
- Type de µP (8, 16, 32 bits) et sa fréquence d'horloge
- Possibilité de traitement multitâche
- Possibilité de multiprogrammation
- Possibilité de communication

Dans le cadre du laboratoire d'automatique, c'est à l'utilisation des automates programmables **Siemens S7 300** que vous serez confrontés :





5 Programmation des automates Siemens Step 7

5.1 La famille Simatic S7

5.1.1 Le S7-200, le micro-API compact.

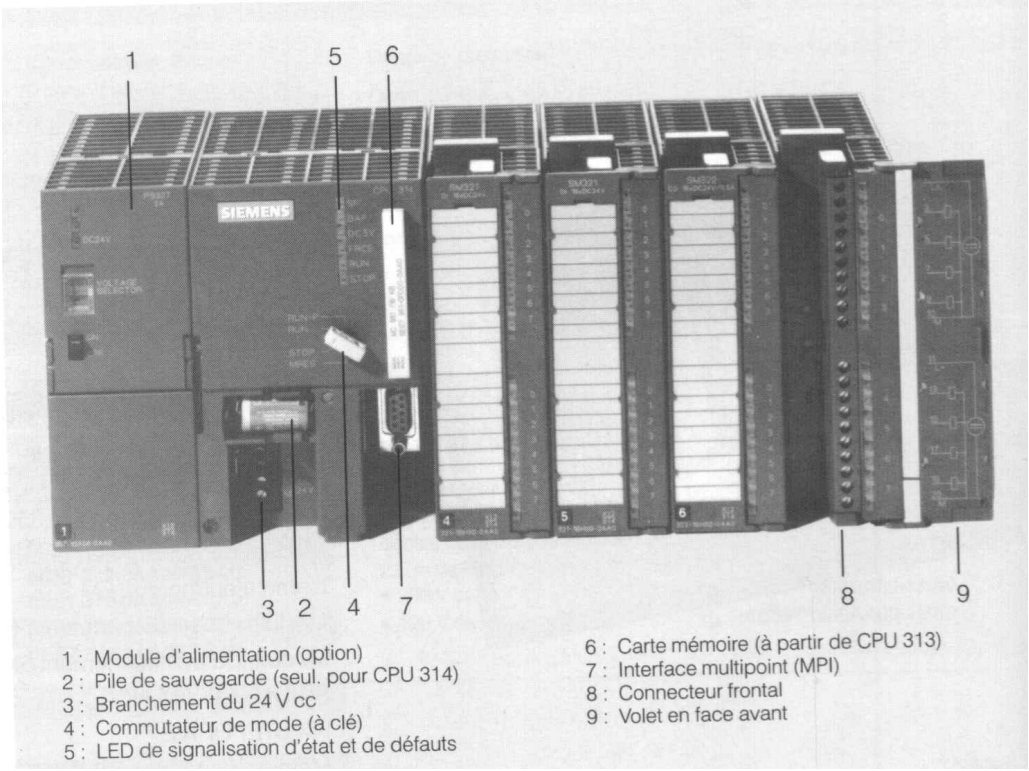
5.1.1.1 Caractéristiques

- Prix avantageux.
- 'Ensemble' regroupant l'alimentation, la CPU, les E/S dans un seul appareil.
- 'Micro-API' à fonctions intégrées.
- Extension modulaire jusqu'à 7 cartes maximum.

5.1.1.2 Fonctions.

- Alimentation électrique intégrée des capteurs.
- Forçage (commande) des entrées et sorties.
- Accès direct aux entrées et sorties.
- Horloge temps réel intégrée (pour la CPU 214).
- Deux potentiomètres analogiques pour CPU 214, un pour CPU 212.
- Deux sorties d'impulsion intégrées(pour la CPU 214).
- Entrées d'interruption.
- Compteurs rapides intégrés.

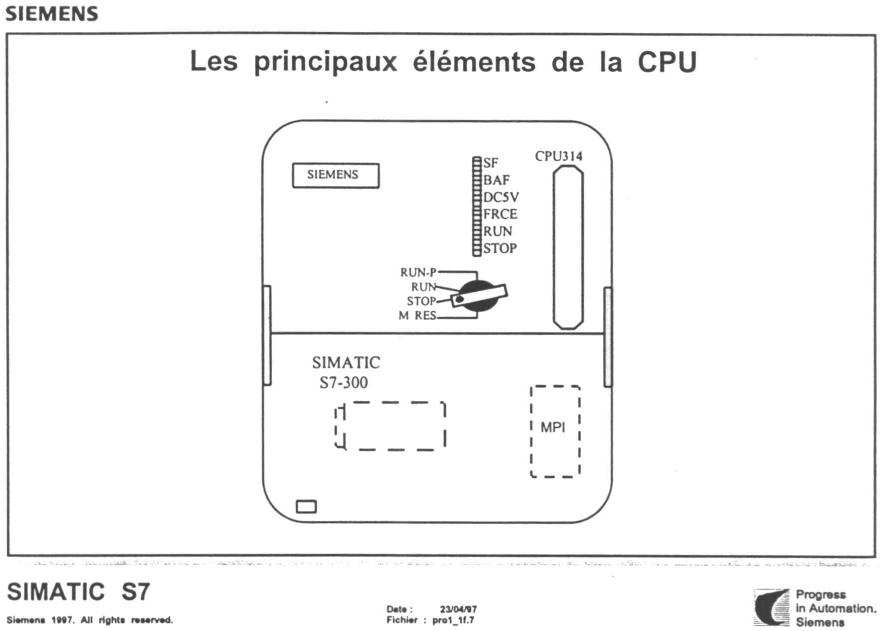
5.1.2 Le S7-300



5.1.2.1 Caractéristiques.

- Gamme échelonnée de CPU.
- Gamme étendue de cartes.
- Extension modulaire jusqu'à 32 cartes.
- CPU avec différentes classes de performance.
- Bus de fond de panier intégré aux cartes.
- Peut être mis en réseau avec l'interface multipoint, PROFIBUS et Industrial Ethernet.
- Raccordement central de la console de programmation avec accès à toutes les cartes (FM, CP).
- Pas de règles à respecter pour l'emplacement des cartes.
- Configuration et réglage des paramètres à l'aide de l'outil Configuration matérielle.

5.1.2.2 Principaux éléments de la CPU.



5.1.2.2.1 Interrupteur à clé :

Cet interrupteur permet le réglage manuel de la CPU. Il comporte 4 positions différentes :

- MRES = effacement général de la RAM.
- STOP = mode Stop ; le programme n'est pas exécuté même lorsque la CPU est en ligne. Accès à toutes les fonctions via la console.
- RUN-P = mode Run ; la CPU exécute le programme et permet la modification de celui-ci par la console.
- RUN = le programme est traité mais il ne peut être que lu (pas de correction possible).

5.1.2.2.2 Visualisation d'état par témoins lumineux.

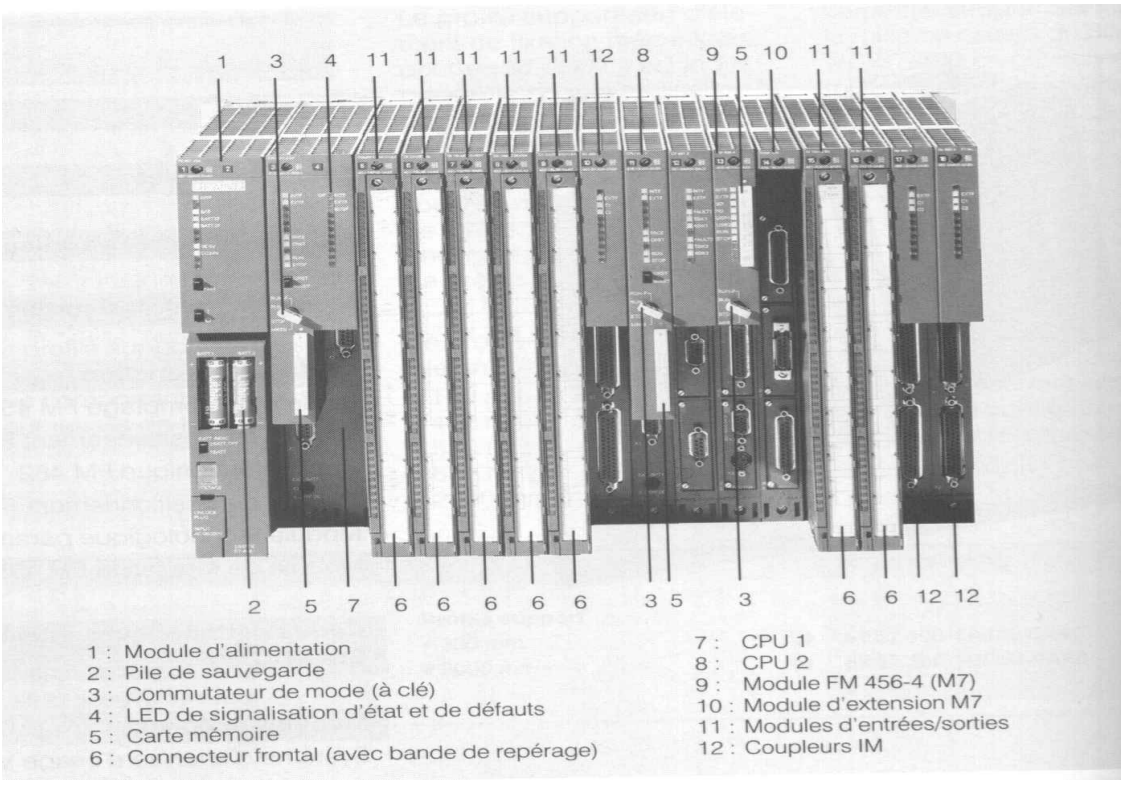
La CPU S7-300 possède 6 LED sur la face avant permettant le contrôle du fonctionnement du programme.

- SF = erreurs groupées ; erreurs internes à la CPU ou à la carte dotée d'une fonction de diagnostic.
- BAF = erreur de pile ; la pile est à plat ou non disponible.
- DC5V = visualisation de la tension d'alimentation interne de 5V.
- FRCE = forçage ; indication qu'au moins une entrée ou sortie est forcée.
- RUN = mode Run ; clignote lors de la mise en route de la CPU, est allumée en permanence en mode Run.
- STOP = mode Stop ; clignote lorsqu'un effacement général est nécessaire, allumée en permanence en mode Stop.

5.1.2.3 Gamme des cartes

- Cartes de signaux : reçoivent les signaux de l'installation et les convertissent aux différents niveaux de tensions internes au S7-300. (Entrées/Sorties TOR ou analogiques).
- Cartes d'interface : autorisent des configurations à plusieurs racks.
- Cartes de fonctions : offrent des « fonctions spéciales » comme le comptage, le positionnement et la régulation.
- Cartes de communication : communication du type Point à Point, PROFIBUS ou Industrial Ethernet.

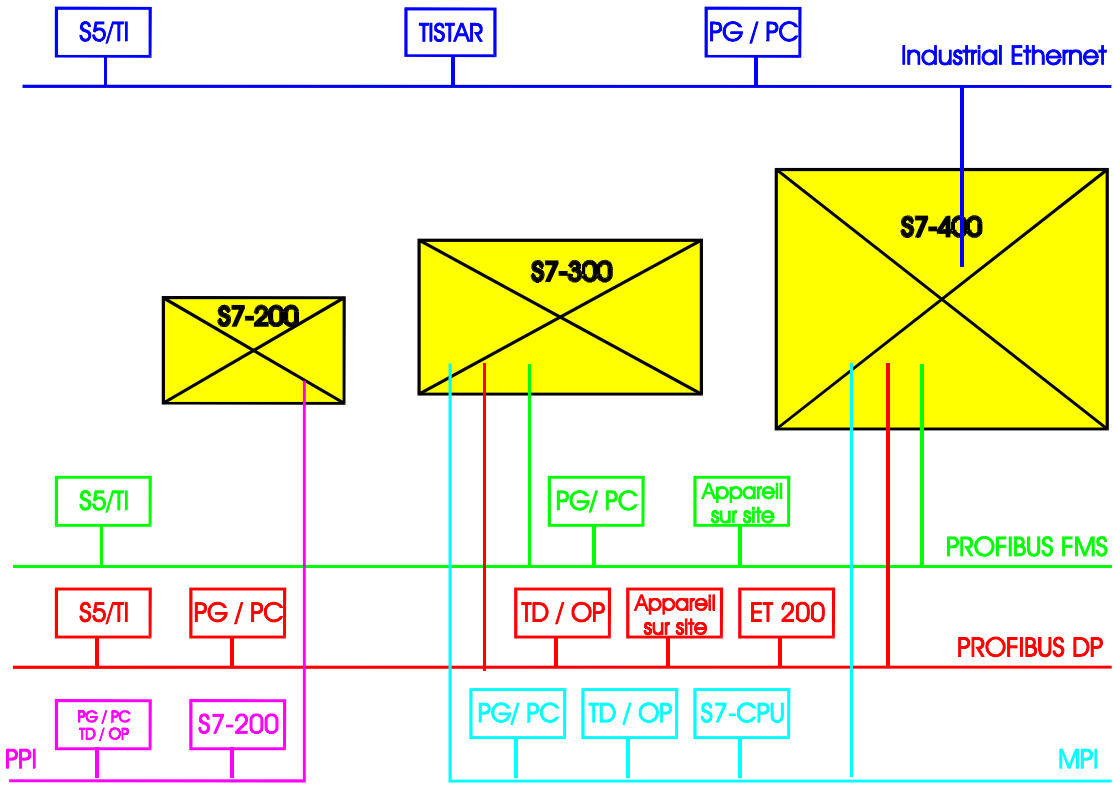
5.1.3 Le S7-400 (pour des applications de milieu et haut de gamme).



La constitution modulaire, compacte et sans ventilateur, l'évolutivité, la robustesse, la facilité de réalisation d'architectures décentralisées et la simplicité d'emploi font du SIMATIC S7-400 la solution idéale pour les tâches les plus diverses dans les moyennes et grandes applications très contraignantes.

5.1.4 Possibilités de mise en réseau

L'illustration ci-dessous montre les différentes possibilités de mise en réseau.



Possibilités de mise en réseau

Les différentes abréviations utilisées sont définies ci-dessous :

- S5/TI : Automates SIMATIC S5 et SIMATIC TI.
- TISTAR : SCADA = (Supervisory Control And Data Acquisition). Interface de commande du système pilote.
- PG/PC :Console de programmation (PG Siemens) ou ordinateur personnel.
- Industrial Ethernet : Réseau local.
- TD/OP : Afficheur de textes et pupitre opérateur.
- Appareils de terrain : Matériel d'entrée/sortie.
- PPI : Interface point à point.
- MPI : Interface multipoint.
- PROFIBUS DP et PROFIBUS FMS : Bus de terrain.

5.2 Fonctionnement de l'automate

5.2.1 Unité de commande

L'automate programmable est un poste central qui commande et surveille le processus de votre installation. Les signaux des capteurs sont enregistrés par les **cartes d'entrées**. La carte unité centrale combine les signaux des capteurs en suivant les instructions du programme. Selon les résultats logiques formés pendant l'exécution du programme, les actionneurs sont commandés par les **cartes de sorties**.

Le cœur de **l'unité centrale CPU** est l'unité arithmétique et logique ALU . Cette unité exécute successivement les instructions de programme en fonction des signaux d'entrées des capteurs pour fournir les résultats logiques de commande des sorties. Afin de réaliser des opérations de calcul, elle comporte plusieurs registres de 16 ou 32 bits, appelés accumulateurs. Des opérations sur mots ou sur octets y sont réalisées. Pour les opérations sur bits, il existe un accumulateur binaire (RLG = résultat logique).

5.2.2 Zones mémoires

L'unité centrale de l'automate dispose de plusieurs types de zones mémoires :

- La **mémoire morte ROM**, qui peut uniquement être lue, contient le système d'exploitation qui gère le fonctionnement de l'automate.
- La **mémoire vive RAM** présente l'avantage de pouvoir être programmée et modifiée rapidement par l'utilisateur mais perd ses données en absence de tension. Une pile de sauvegarde a donc été prévue.
  Cette mémoire RAM contient :
  - le programme transféré à partir d'une console de programmation
  - les temporisateurs, les compteurs et la zone des mémentos (mémorisations des résultats intermédiaires).
  - les mémoires images des entrées et de sorties **MIE MIS** dans lesquelles sont mémorisés les états des signaux d'entrées et les états à affecter aux sorties.
  - une zone réservée aux données système utilisée pour l'organisation du déroulement du programme.

5.2.3 Adressage

Dans un automate programmable, les modules d'entrées et de sorties permettent de relier le processus au CPU en adaptant les niveaux de tensions externes à un niveau convenant au CPU. Les modules d'entrées et de sorties sont reliés à l'unité centrale par bus parallèles (formés de plusieurs liaisons parallèles) : le bus de données, le bus d'adresses et le bus de commande.

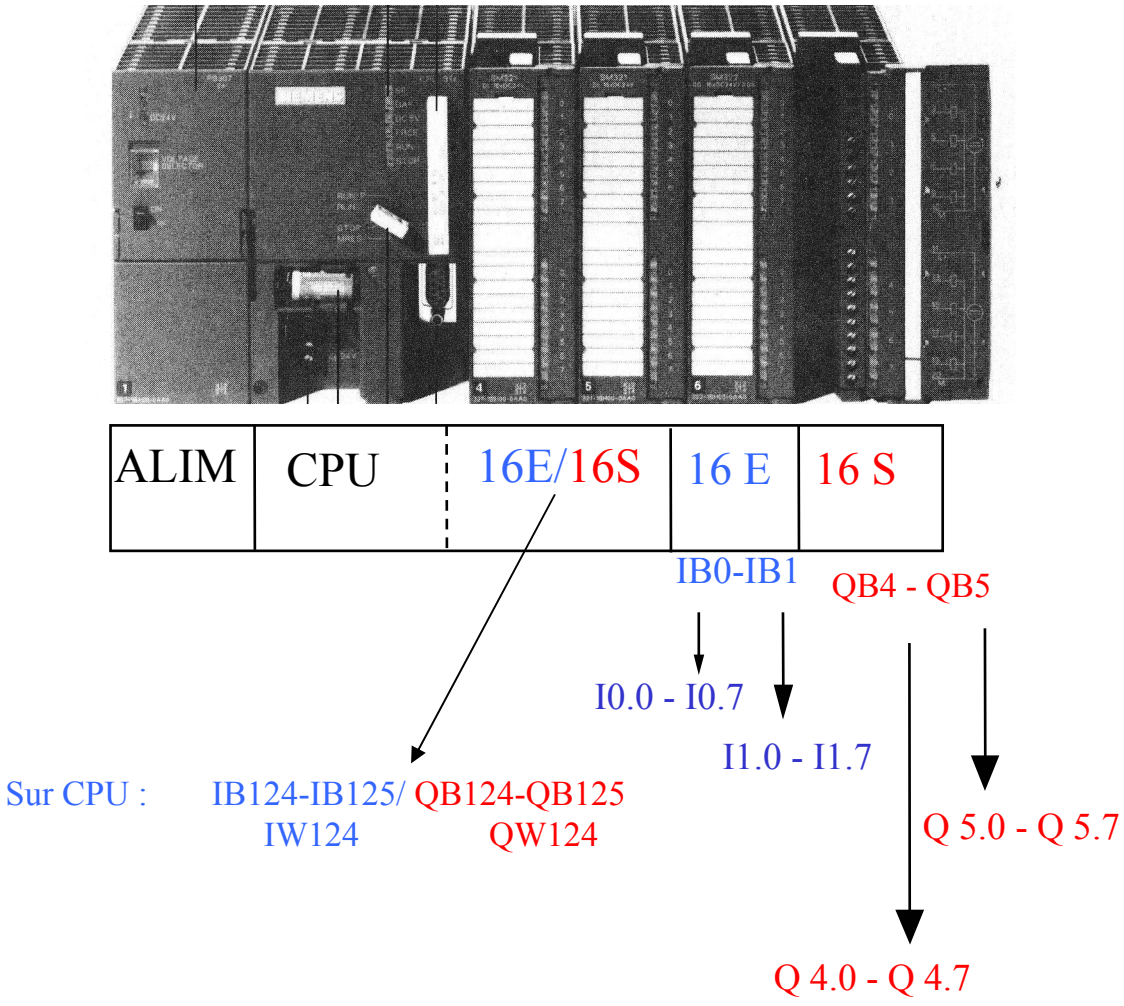
L'accès aux entrées et sorties se réalise grâce à une adresse. Chaque canal d'une carte TOR est désigné par un numéro de bit de 0 à 7. Les bits sont regroupés en octets : 1 octet = 8 bits.

Chaque octet reçoit un numéro d'octet. Une adresse numérique se compose d'une adresse octet et d'une adresse bit.

Ex : le bit 3 de l'octet 0 est répertorié : 0.3
le bit 0 de l'octet 1 est répertorié : 1.0
l'octet 0 comprend les bits allant de 0.0 à 0.7

Les entrées sont précédées de la lettre **I**. Les sorties sont précédées de la lettre **Q**.

L'adresse du module dépend de sa position par rapport au CPU. Un module peut comporter jusqu'à 4 octets de signaux ainsi chaque emplacement de module utilise 4 octets d'adresses. Le module enfiché au premier emplacement utilise les adresses octet de 0 à 3. Le module suivant aura les adresses 4 à 7 et ainsi de suite. Les adresses peuvent être modifiées dans la configuration matérielle.



5.2.4 Traitement cyclique

Dans les automates SIEMENS , le déroulement du programme est coordonné par le logiciel système de l'automate qui exécute les programmes de manière cyclique. Le logiciel système commande au processeur de traiter le bloc d'organisation spécifique appelé OB1 dès que celui-ci se trouve dans la mémoire de l'A.P.

Le programme peut être entièrement écrit dans ce bloc OB1 (programmation linéaire) ou alors le programme peut être réparti dans plusieurs autres blocs (programmation structurée) : les blocs contenant le programme doivent se trouver dans la mémoire de l'A.P. et seront appelés par le bloc OB1. Chaque bloc de programme à traiter doit être appelé par une instruction de saut. Ces instructions demandent au processeur de "sauter" dans le bloc afin de traiter le programme qu'il contient.

Quand l'OB1 contient plusieurs instructions de saut, le processeur les traite successivement. Il saute à chaque fois dans le bloc appelé pour traiter le programme que contient le bloc. A la fin du bloc, il revient alors dans l'OB1 pour traiter l'instruction de saut suivante.

Lorsque le bloc d'organisation est terminé, le processeur quitte ce bloc pour reprendre le traitement des instructions du logiciel système. Lorsque le traitement des routines du logiciel système est terminé, le processeur reprend le traitement du programme de l'OB1 dès son début. Cette succession de traitements est appelée ***cycle***. Le processeur effectue ainsi un ***traitement cyclique*** du programme. Le temps mis par le processeur pour parcourir une fois tout le programme est appelé temps de cycle.

5.2.5 Déroulement du processus de commande

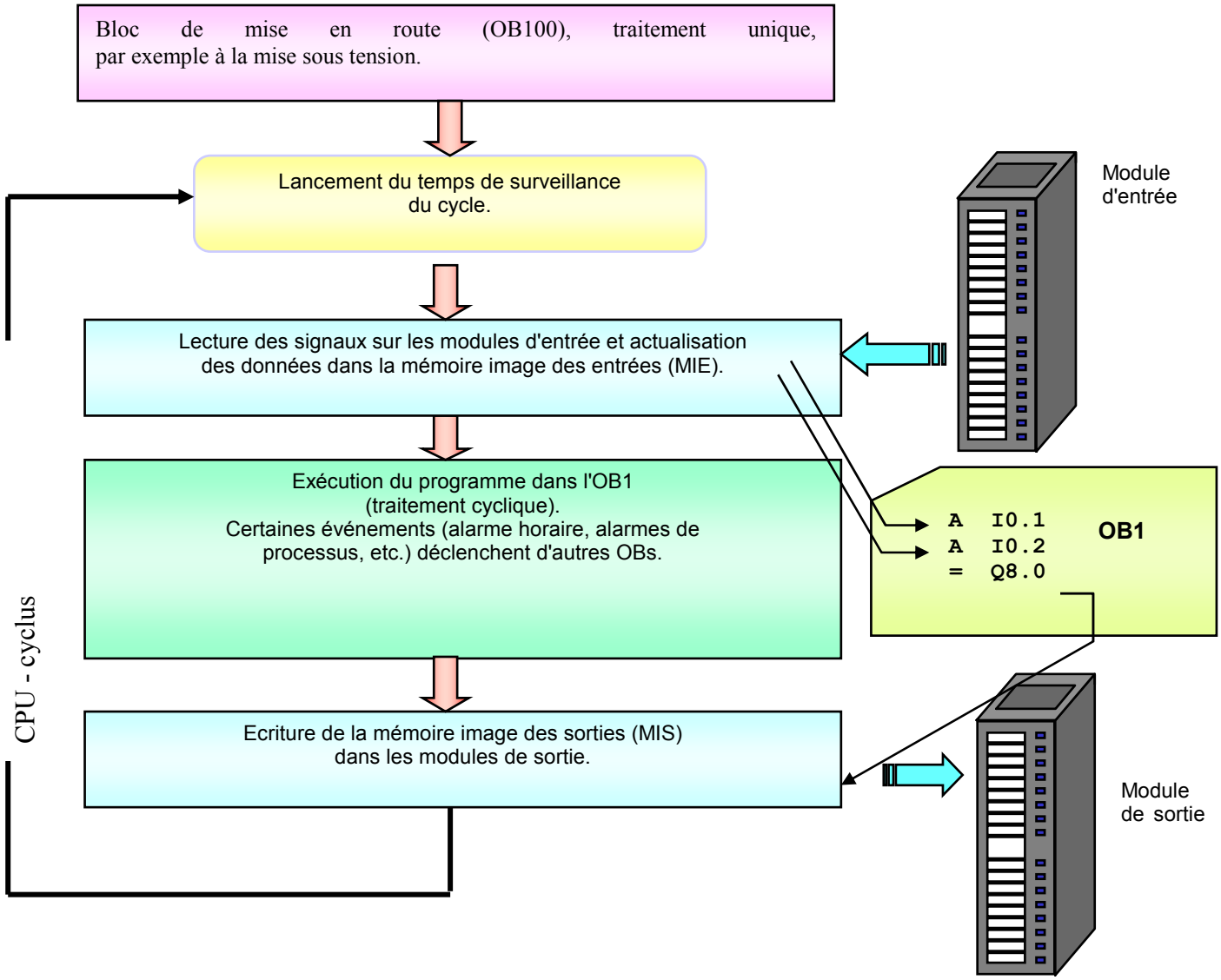
Avant le début d'un cycle de programme, le processeur exécute une routine du système et démarre la surveillance du temps de cycle. Si le traitement du programme amène un dépassement de ce délai de surveillance, l'automate se met en STOP.

Le processeur mot de l'automate copie les états des signaux des cartes d'entrées dans une mémoire image des entrées **MIE**. Le programme est exécuté suivant l'ordre dans lequel il est écrit et il puise ses informations dans la MIE. Si une entrée change d'état, l'automate n'en tiendra compte qu'au cycle suivant.

Les résultats sont stockés au fur et à mesure de leur traitement dans la mémoire image des sorties **MIS**. A la fin de l'exécution du programme c-à-d à la fin de l'OB1, la MIS est transmise aux cartes de sorties.

Un nouveau cycle recommence assurant le fonctionnement cyclique de l'automate.
Le traitement cyclique n'est assuré que par l'indispensable OB1.

Le fait de charger un bloc dans l'automate le place en mémoire mais n'en assure pas l'exécution. En effet, l'exécution d'un bloc ne sera active que s'il y a appel cyclique, donc référence à l'OB1 (ou périodique .).



5.2.6 La programmation structurée : Blocs du programme utilisateur

La programmation structurée permet de fragmenter les programmes en BLOCS, un bloc ayant une tâche bien définie dans le contrôle d'une machine ou processus ou d'une partie de machine ou processus.
La mise en service et le dépannage d'un programme bien structuré s'en trouvera grandement facilitée. Les blocs peuvent être testés séparément et un défaut peut être localisé beaucoup plus facilement.

Il existe 4 types de blocs disponibles pour l'utilisateur : les blocs d'organisation (OB), les blocs fonctionnels (FB), les fonctions (FC) et les blocs de données (DB).

5.2.6.1 Les blocs d'organisation (OB)

Les OB ont chacun leur fonction propre et sont appelés directement par le système d'exploitation de l'automate. Lorsqu'un événement précis apparaît, la CPU appelle l'OB correspondant.
L'utilisateur peut programmer ces différents blocs d'organisation et ainsi réagir aux différents événements pouvant survenir lors de l'exécution d'un programme.
L'OB1 est exécuté cycliquement. C'est ce bloc d'organisation qui comporte la structure du programme. En effet, tout programme est scindé en plusieurs blocs et c'est via l'OB1 que ceux-ci sont appelés lorsqu'ils sont nécessaires.
En plus de l'OB1, le système d'exploitation peut appeler d'autres OB pour réagir à des événements, tels que : Interruptions d'horloge, interruptions de diagnostic, interruptions de traitement d'erreur, interruptions temporisées, interruptions matérielles, mise en route du matériel.

5.2.6.2 Fonctions (FC)

Une fonction est un bloc logique ou une séquence d'instructions. Les données temporaires sont stockées dans la pile L jusqu'à ce que la fonction soit close.
Il existe deux types de FC : avec ou sans paramètres. Les FC sans paramètre sont considérées comme des sous-programmes du bloc de code appelant. Si le bloc appelant n'a délivré aucun paramètre, la FC exécute son code avec ses propres valeurs. Les valeurs sont mémorisées dans la pile L, une mémoire locale affectée de façon dynamique. A la fin de l'exécution de la FC, les valeurs contenues dans la pile L sont effacées.
Par contre, les FC avec paramètres permettent au bloc appelant d'avoir accès aux noms symboliques des paramètres déclarés dans la FC.

5.2.6.3 Les blocs fonctionnels (FB)

Un bloc fonctionnel est un bloc ou une séquence d'instructions auquel est affectée une mémoire permettant de stocker des variables. Un FB a besoin de cette mémoire supplémentaire sous forme de 'bloc de données d'instance' (DB). Les paramètres sont déclarés dans le FB et certaines données locales sont mémorisées dans le bloc de données d'instance. D'autres données temporaires sont stockées dans la pile locale (L).
Les données qui sont mémorisées dans le bloc de données d'instance sont conservées lorsque le bloc fonctionnel est clos. Les données qui sont mémorisées dans la pile L ne sont pas rémanentes.

Une application intéressante pour les FB est le modèle multi-instance. En fait, comme écrit ci-dessus, chaque FB a besoin d'un DB d'instance. Cependant, STEP7 autorise une autre solution. En effet, un FB appelant (gestionnaire d'instances) gère les données des FB de niveau inférieur avec son propre DB d'instance. Pour ce faire, il faut veiller à ce que le FB de niveau inférieur soit défini dans le FB appelant sous forme de « variable statique » du type Fbxy.
Le niveau d'imbrication maximal est de 8 FB.

5.2.6.4 Blocs de données (DB)

Un bloc de données est une zone affectée de manière fixe dans laquelle sont mémorisées des données ou des informations qui ont été collectées par une autre fonction. Les blocs de données sont des zones de stockage de valeurs qui peuvent être chargées avec les blocs de programme dans la CPU.
Il existe deux types de données : les blocs de données globaux et les blocs de données d'instance.
Les DB globaux peuvent être employés par tous les blocs logiques dans le programme. Tous les FB, FC ou OB peuvent écrire ou lire des données dans un DB global. Pour accéder aux données, il suffit d'ouvrir le bloc. Les données restent mémorisées dans le bloc de données, même après sa fermeture.
Un DB d'instance est associé à un bloc fonctionnel défini. Les données mémorisées dans ces blocs de données sont lues ou écrites par le bloc fonctionnel associé.
La création d'un DB d'instance se fait généralement en parallèle avec celle d'un FB à un tel point que le système exécute lui-même l'opération. Le DB aura la même structure que celle indiquée dans la partie déclarative des variables du FB.
La CPU dispose de deux registres de blocs de données, les registres DB et DI. Ainsi, deux blocs de données peuvent être ouverts simultanément. L'ouverture d'un bloc de données dans un registre déjà affecté provoque la fermeture du bloc de données ouvert précédemment.

5.2.6.5 Appel de blocs

Lors de l'appel d'un bloc FC ou FB, il faut spécifier à cet instant les différents paramètres que l'on désire traiter par ces blocs. Ceux-ci doivent apparaître dans l'ordre approprié fixé dans la table de déclaration des variables du bloc appelé.
En plus de ces paramètres, on remarque, en langage CONT ou LOG, deux entrées supplémentaires : EN(=Enable Input) et ENO(=Enable Output).

Si l'entrée EN est activée (état logique 1), il y a une exécution de l'opération et inversement.
Si la sortie ENO est activée, (état logique 1), l'opération a été exécutée sans erreur. Par contre, si elle n'est pas activée, cela signifie que soit l'opération n'a pas été appelée pour son exécution ou qu'une erreur est survenue lors de son exécution.

5.2.7 Blocs système

Il existe 3 types de blocs système : les fonctions système (SFC), les blocs fonctionnels système (SFB), les blocs de données système (SDB).

5.2.7.1 Les fonctions système (SFC)

Une fonction système est une fonction préprogrammée testée, intégrée dans la CPU S7, comme par exemple le réglage des paramètres pour des modules, le transfert de données, des fonctions de copies, etc. Les SFC peuvent être appelées à partir du programme via le catalogue de fonctions disponible dans l'éditeur de blocs de programme. Il n'est pas nécessaire d'affecter un bloc de données (DB) aux SFC.

5.2.7.2 Les blocs fonctionnels système (SFB)

Un bloc fonctionnel système est une partie intégrée de la CPU S7. On peut appeler un SFB à partir du programme car les SFB font partie du via le catalogue de fonctions disponible dans l'éditeur de blocs de programme. Il faut affecter aux SFB un DB d'instance qui doit être généré dans le projet puis chargé dans la CPU en tant que partie du programme.

5.2.7.3 Les blocs de données système (SDB)

Un bloc de données système est une zone mémoire de programme créée par différents outils STEP 7 et dans laquelle sont stockées des données nécessaires à l'exploitation de la commande. On mémorise dans les SDB des informations telles que les données de configuration, les critères de communication et les paramètres des cartes de périphérie.

5.3 Le logiciel STEP 7.

5.3.1 Introduction

5.3.1.1 Présentation du produit

STEP 7 est le logiciel permettant de configurer et de programmer les automates programmables SIMATIC S7-300/400 et les systèmes d'automatisation SIMATIC M7-300/400, ainsi que les systèmes intégrés compacts SIMATIC C7.
Le système d'automatisation M7 consiste en un calculateur industriel qui peut être mis en œuvre dans des cas d'automatisation qui nécessitent des impératifs technologiques spéciaux, des asservissements rapides ou des tâches spécifiques de communication.
Un système intégré compact C7 consiste en l'intégration d'un automate programmable et d'un pupitre opérateur qui permet de réaliser des commandes de machine d'une extrême compacité.
STEP7 est constitué d'un logiciel de base et de logiciels optionnels.

5.3.1.2 Logiciel de base

Le logiciel de base est composé de plusieurs éléments dont :
* Création et gestion de projets
* Configuration et paramétrage du matériel et de la communication
* Gestion des mnémoniques
  La création de programmes pour système cible S7 (il existe des logiciels optionnels pour créer des programmes pour les systèmes cibles M7).
* Chargement de programmes dans des systèmes cibles.
* Test de l'installation d'automatisation.
* Diagnostic lors de perturbations de l'installation.

Le logiciel de base supporte 3 langages de programmation différents :

a) Le schéma à contact (CONT)
b) Les listes d'instructions (LIST)
c) Le logigramme (LOG)

5.3.1.3 Logiciels optionnels

- Configuration de réseaux
- Téléservice : permet les mises à jours de programmes via le réseau Internet par exemple.
- DOCPRO : logiciel mettant à disposition des ressources pour l'établissement et la gestion de la documentation de projets.
- Simulation : logiciel permettant la visualisation lors de simulation de programme.
- Langages de programmation (GRAPH, HiGraph, SCL et CFC)

5.3.2 Installation de STEP 7

5.3.2.1 Installation du logiciel STEP 7

STEP 7 contient un programme d'installation automatique « SETUP.EXE ».
Donc le logiciel s'installe comme tout software de type Microsoft ce qui facilite énormément son installation.
De plus, le programme SETUP guide l'utilisateur pas à pas tout au long de la procédure d'installation.

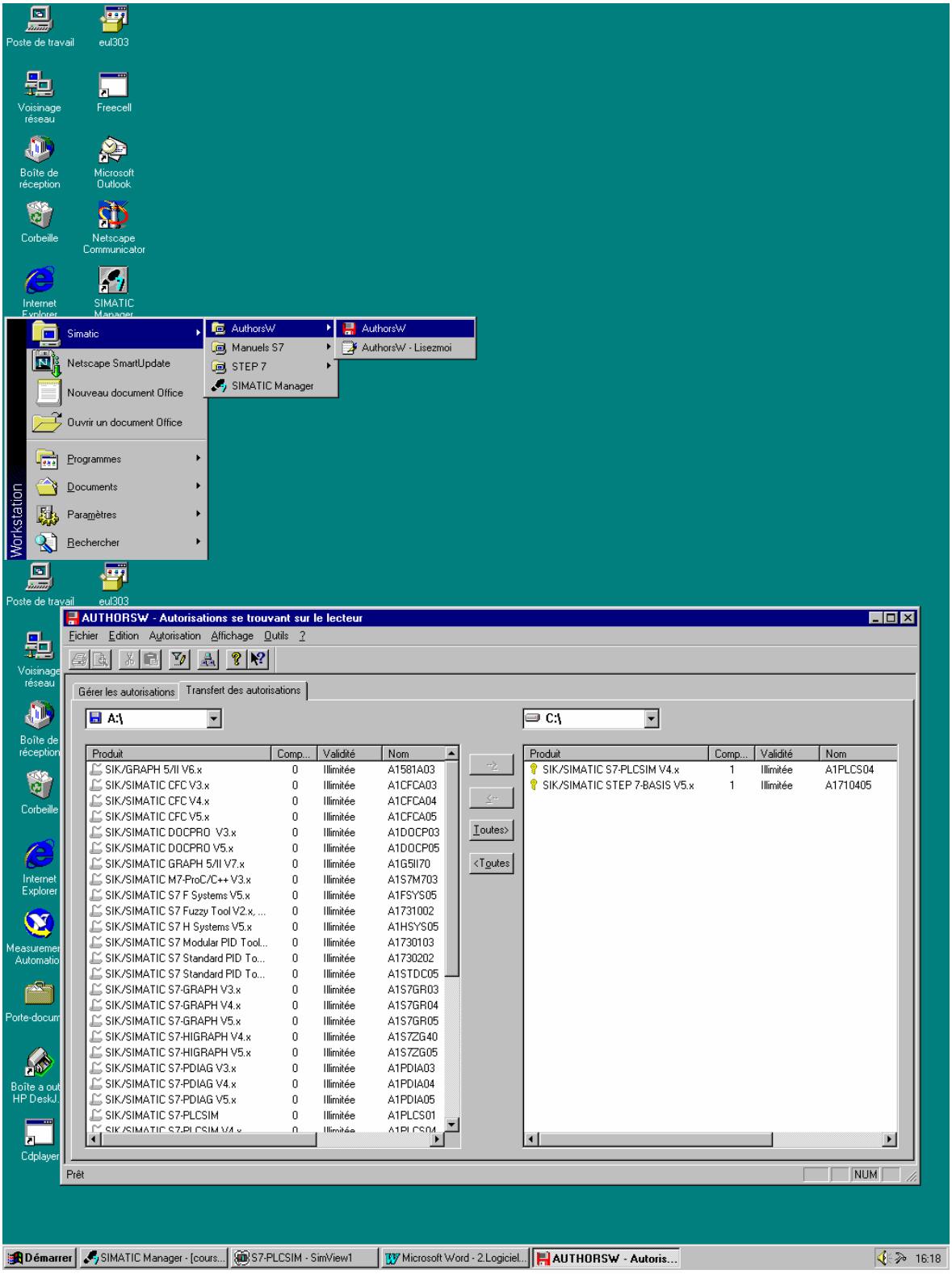
5.3.2.2 Autorisation, licence d'utilisation.

Le logiciel STEP 7 est fourni avec une licence d'utilisation software. Celle-ci se présente sous la forme d'une disquette ou clé USB contenant un fichier permettant l'utilisation et l'exploitation du logiciel STEP7.
L'autorisation s'installe pendant ou après l'installation du STEP 7.

Pour installer l'autorisation après l'installation du STEP7 :

Placer la disquette d'autorisation dans le lecteur puis dans le menu Démarrer, sélectionner SIMATIC => Authorsw =>

La fenêtre montre les autorisations se trouvant sur la disquette et celle se trouvant sur le lecteur C :
Il faut ensuite sélectionner l'autorisation voulue et la transférer à l'aide de la flèche. L'autorisation doit être installée sur le disque C :


Poste de travail
eul303
Voisinage réseau
Freecell
Boîte de réception
Microsoft Outlook
Corbeille
Netscape Communicator
Internet Explorer
SIMATIC Manager
Simatic
Netscape SmartUpdate
Nouveau document Office
Ouvrir un document Office
Programmes
Documents
Paramètres
Rechercher
Workstation
AuthorsW
Manuels S7
STEP 7
SIMATIC Manager
AuthorsW
AuthorsW - Lisezmoi
AUTHORSW - Autorisations se trouvant sur le lecteur
Fichier Edition Autorisation Affichage Outils ?
Gérer les autorisations
Transfert des autorisations
A:\
C:\
Produit Comp... Validité Nom
SIK/GRAPH 5/II V6.x 0 Illimitée A1581A03
SIK/SIMATIC CFC V3.x 0 Illimitée A1CFCA03
SIK/SIMATIC CFC V4.x 0 Illimitée A1CFCA04
SIK/SIMATIC CFC V5.x 0 Illimitée A1CFCA05
SIK/SIMATIC DOCPRO V3.x 0 Illimitée A1DOCP03
SIK/SIMATIC DOCPRO V5.x 0 Illimitée A1DOCP05
SIK/SIMATIC GRAPH 5/II V7.x 0 Illimitée A1G5II70
SIK/SIMATIC M7-ProC/C++ V3.x 0 Illimitée A1S7M703
SIK/SIMATIC S7 F Systems V5.x 0 Illimitée A1FSYS05
SIK/SIMATIC S7 Fuzzy Tool V2.x, ... 0 Illimitée A1731002
SIK/SIMATIC S7 H Systems V5.x 0 Illimitée A1HSYS05
SIK/SIMATIC S7 Modular PID Tool... 0 Illimitée A1730103
SIK/SIMATIC S7 Standard PID To... 0 Illimitée A1730202
SIK/SIMATIC S7 Standard PID To... 0 Illimitée A1STDC05
SIK/SIMATIC S7-GRAPH V3.x 0 Illimitée A1S7GR03
SIK/SIMATIC S7-GRAPH V4.x 0 Illimitée A1S7GR04
SIK/SIMATIC S7-GRAPH V5.x 0 Illimitée A1S7GR05
SIK/SIMATIC S7-HIGRAPH V4.x 0 Illimitée A1S7ZG40
SIK/SIMATIC S7-HIGRAPH V5.x 0 Illimitée A1S7ZG05
SIK/SIMATIC S7-PDIAG V3.x 0 Illimitée A1PDIA03
SIK/SIMATIC S7-PDIAG V4.x 0 Illimitée A1PDIA04
SIK/SIMATIC S7-PDIAG V5.x 0 Illimitée A1PDIA05
SIK/SIMATIC S7-PLCSIM 0 Illimitée A1PLCS01
Toutes>
<Toutes
SIK/SIMATIC S7-PLCSIM V4.x 1 Illimitée A1PLCS04
SIK/SIMATIC STEP 7-BASIS V5.x 1 Illimitée A1710405
Prêt
NUM
Démarrer
SIMATIC Manager - [cours...
S7-PLCSIM - SimView1
Microsoft Word - 2.Logiciel...
AUTHORSW - Autoris...
16:18

5.3.2.3 Outils STEP 7.

Les outils STEP 7 sont les différents programmes installés utiles à la programmation des systèmes d'automatisation S7-300/400.

Paramétrage des cartes à mémoire. Cet outil permet de paramétrer une carte mémoire.

Paramétrage de l'interface PG-PC. L'interface PG est paramétrée avec cet outil (par exemple, Interrupt ou adresse ES).

Conversion de fichiers S5-S7. Le convertisseur S5/S7 vous permet de transposer des programmes ou des blocs STEP 5 en programmes ou blocs STEP 7.

Programmations de blocs S7. Cet outil vous permet d'écrire votre programme avec l'un des langages de programmation STEP 7 : schéma à contacts (CONT) et liste d'instructions (LIST).

Didacticiel S7. Programme sur la console de programmation ou le PC permettant de se familiariser avec le SIMATIC S7.

SIMATIC Manager. Il s'agit du programme principal, qui apparaît d'ailleurs directement sur le bureau de WINDOWS 95. Il permet de gérer les projets, d'établir une liaison en ligne, de lire l'état des modules ou de programmer des cartes mémoire.

5.3.3 Installation du matériel.

La communication entre l'automate SIMATIC S7 et le PC ou PG se fait par l'intermédiaire d'une carte MPI (Multi Point Interface) ou grâce à un câble de conversion RS232/MPI.

5.3.3.1 Installation de la carte de communication.

La carte MPI permettant la communication entre le PC et la CPU requiert un slot ISA de 16 bits libre. Il faut manipuler la carte avec précaution. Pour l'installation dans le PC, tout d'abord, couper l'alimentation du PC, enlever le capot de protection du PC, décharger son corps de l'électricité statique (en touchant une table métallique par exemple), insérer la carte dans le slot libre sans toucher les autres cartes. Vérifier que la carte est correctement enfoncée, refermer le capot et allumer le PC.
Attention, il semblerait que la carte pose des problèmes si le PC comporte d'autres cartes de communication comme des cartes réseau Ethernet, des cartes PROFIBUS.

5.3.3.2 Paramétrage de l'interface PG/PC.

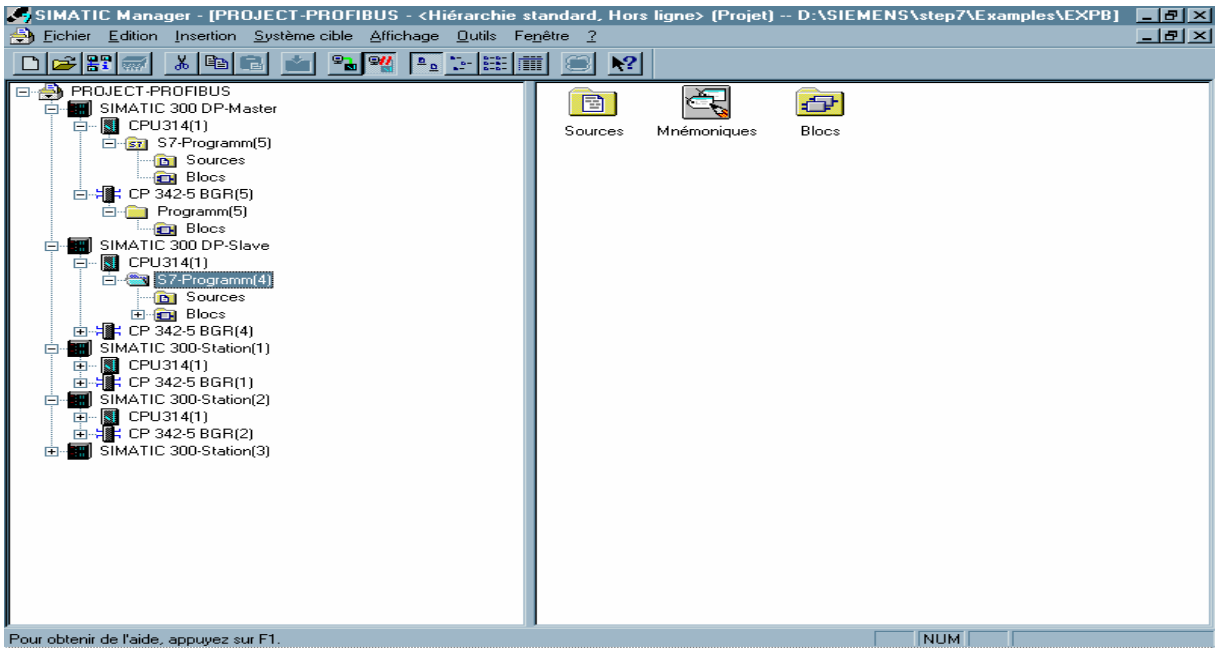
Le paramétrage qu'il faut réaliser permet de définir la communication entre PG/PC et le système d'automatisation.
Lors de l'utilisation d'un PC avec une carte MPI, il faut vérifier dans le « Panneau de configuration » de Windows l'absence de conflits d'interruption ou de recoupement de plages d'adresses. Celle-ci étant directement configurable sur la carte à l'aide de Jumpers.

5.3.3.3 Montage de l'automate.

Le montage de l'automate ne requiert aucune compétence particulière.
Il faut cependant veiller à respecter la convention d'emplacement des différents modules. En effet, il faut dans l'ordre, installer le module d'alimentation, la CPU, la carte IM, les cartes d'entrées et de sorties et la CP. L'emplacement du module IM (le numéro 3) doit être libre si l'on ne dispose pas de cet élément.

5.3.4 Lancement STEP 7

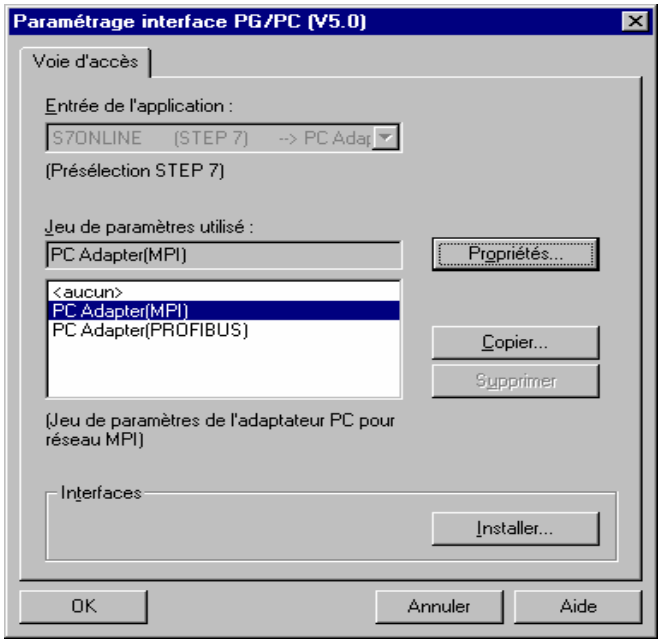
Le SIMATIC Manager se présente sous la forme d'une icône sur le bureau de WINDOWS .
L'outil SIMATIC Manager fonctionne de manière similaire au gestionnaire de fichiers de Windows, il ne tient cependant compte que des fichiers et structures significatives pour STEP 7, il vous permet de voir les fichiers qui sont directement liés aux applications STEP 7.
L'option Fenêtre/Disposition permet d'agencer plusieurs fenêtres en cascade ou en mosaïque.



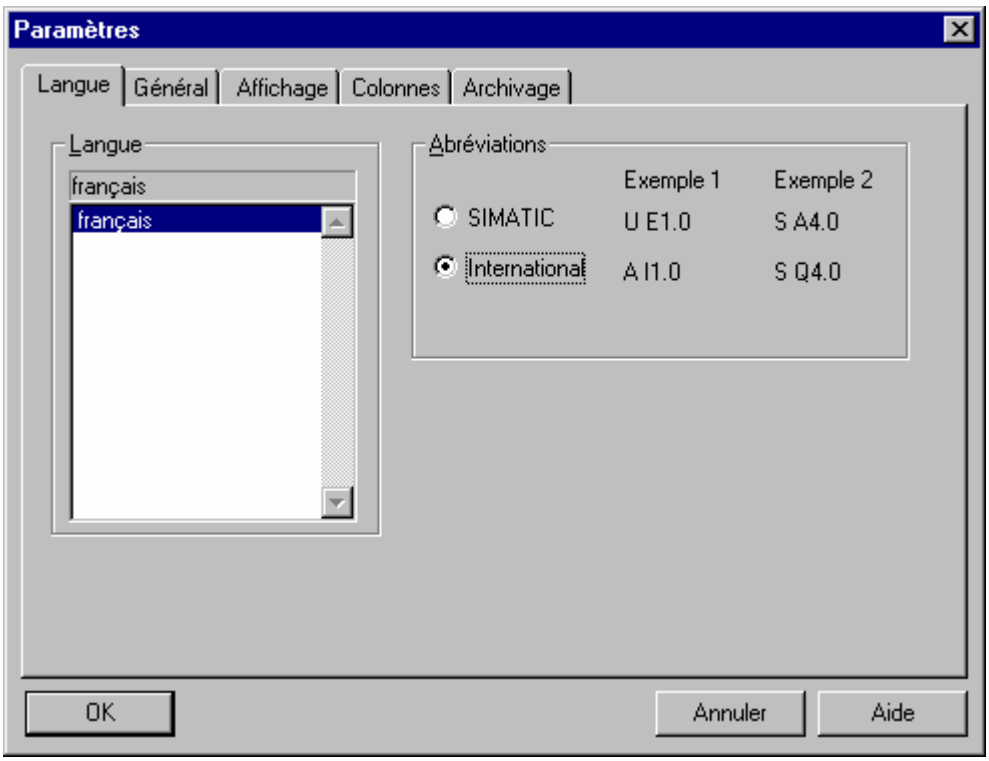
5.3.4.1 Paramétrage du SIMATIC MANAGER

Il y a deux paramétrages importants : le paramétrage de l'interface de communication et le paramétrage du projet.

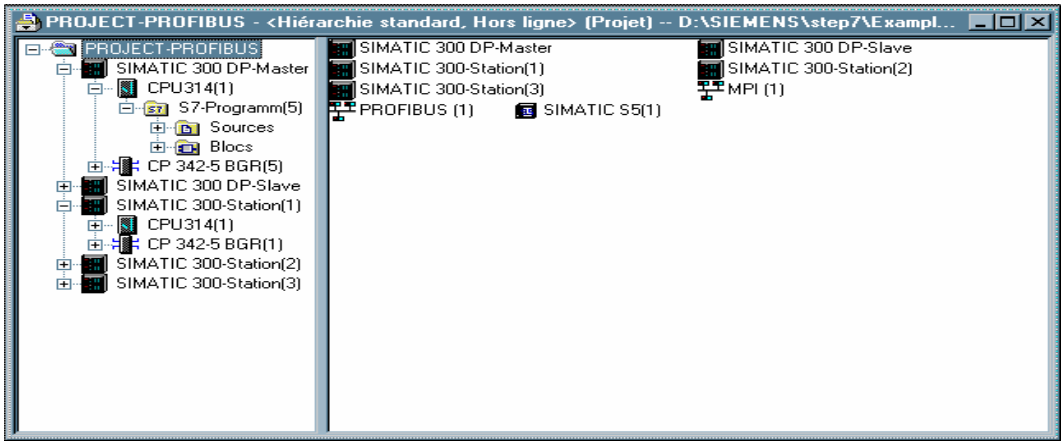
1. Le paramétrage de l'interface permet de choisir le mode de liaison avec l'automate et ses propriétés telles l'adresse et la vitesse de transmission.
   SIMATIC Manager => outils => paramétrage interface PG/PC



2. Le paramétrage du projet permet de choisir la langue de travail, le langage de programmation, le répertoire projet, la disposition dans les fenêtres, le logiciel d'archivage, …
   SIMATIC Manager => outils => paramètres

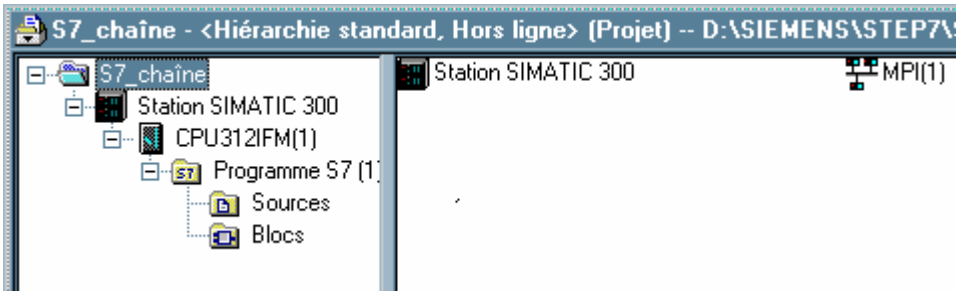


5.3.4.2 Structure des fichiers des projets STEP 7



Un projet est composé de 2 fenêtres distinctes.
Dans la fenêtre de gauche, le projet correspond à un répertoire du disque dur. SIMATIC Manager comprend un répertoire Projet. L'utilisateur peut y définir ses propres projets. Un projet est l'ensemble de toutes les parties d'une application : il peut contenir le programme automate, le programme d'un pupitre opérateur, la programmation du réseau, … avec le même fichier symbolique.
Le niveau inférieur des programmes S7 contient le programme CPU (programme utilisateur) avec les blocs du programme utilisateur et un dossier Sources. Dans ce dernier, on peut programmer des blocs avec un éditeur orienté source en recourant à la syntaxe LIST, langage de programmation nécessaire à la réalisation d'une tâche d'automatisation dans la CPU S7.
La fenêtre de droite montre le contenu de l'élément sélectionné dans la fenêtre de gauche.

5.3.4.3 Structure d'un projet dans SIMATIC



Projets	L'icône Projet se situe au premier niveau.
Station	On peut raccorder plusieurs stations S7-300 et S7-400. On peut obtenir ici les informations relatives à la structure matérielle de la station et à ses données de configuration.
Programmes S7	Les programmes S7 contiennent aussi des objets pour les différentes applications ou méthodes de programmation.
Réseaux	Les icônes de réseaux fournissent des informations sur le réseau affiché.
Sources	Une source (fichier de texte) est une partie de programme créée dans un éditeur graphique (Graph S7, Higraph S7) ou textuel (SCL, LIST, éditeur ASCII quelconque). La compilation de la source fournit le programme utilisateur S7 exécutable

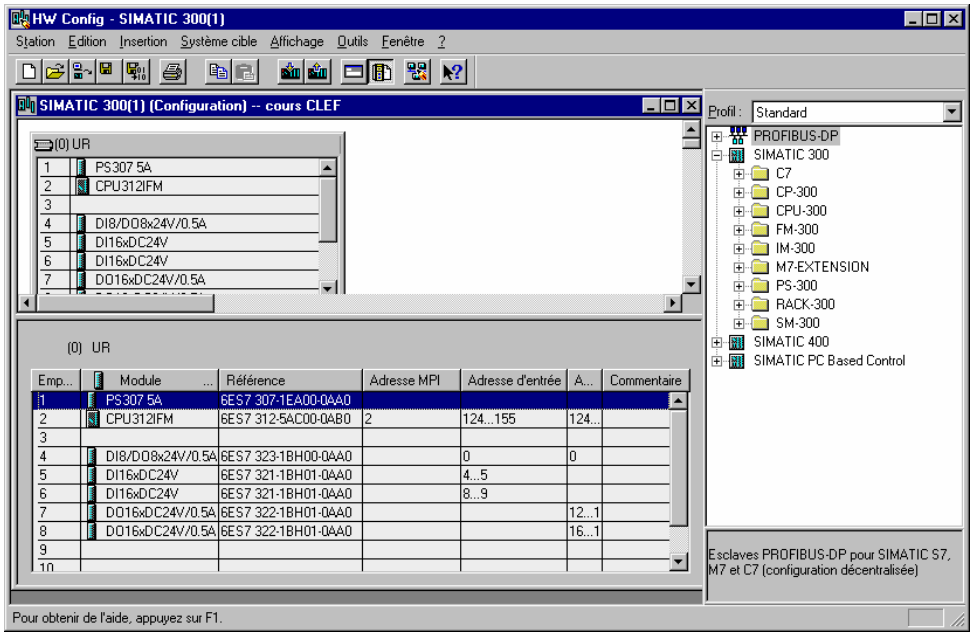
5.4 Configuration matérielle

5.4.1 Configuration effective

Lors du démarrage, la CPU crée une configuration effective, c'est-à-dire qu'elle mémorise la disposition des cartes et affecte les adresses selon un algorithme fixe. Si aucun paramétrage n'a été effectué, le système utilise les paramètres par défauts définis en usine.
Le système mémorise cette configuration effective dans un bloc de données système. Avec le logiciel STEP7, il est possible de lire la configuration effective et de l'utiliser en préréglage pour des structures identiques ou de la modifier avec l'outil Configuration matérielle. Cet outil permet aussi le paramétrage de la CPU et des cartes de périphérie.
La configuration que vous définissez est appelée Configuration personnalisée.

5.4.2 Configuration personnalisée

L'outil Configuration matérielle est appelé via le SIMATIC Manager en sélectionnant la station matérielle, en cliquant avec le bouton droit de la souris et en activant l'option de menu « ouvrir un objet » ***ou*** en cliquant deux fois sur l'icône correspondante
L'utilisateur peut créer une configuration conforme à ses souhaits.
Pour ce faire, il sélectionne des cartes dans le catalogue électronique et les place aux emplacements souhaités. Le système attribue automatiquement les adresses aux cartes.
Une fois la configuration terminée, l'utilisateur peut paramétrer une carte en cliquant deux fois dessus.



A partir de la CPU 315-2 et avec le S7-400, les adresses des cartes peuvent être choisies librement.

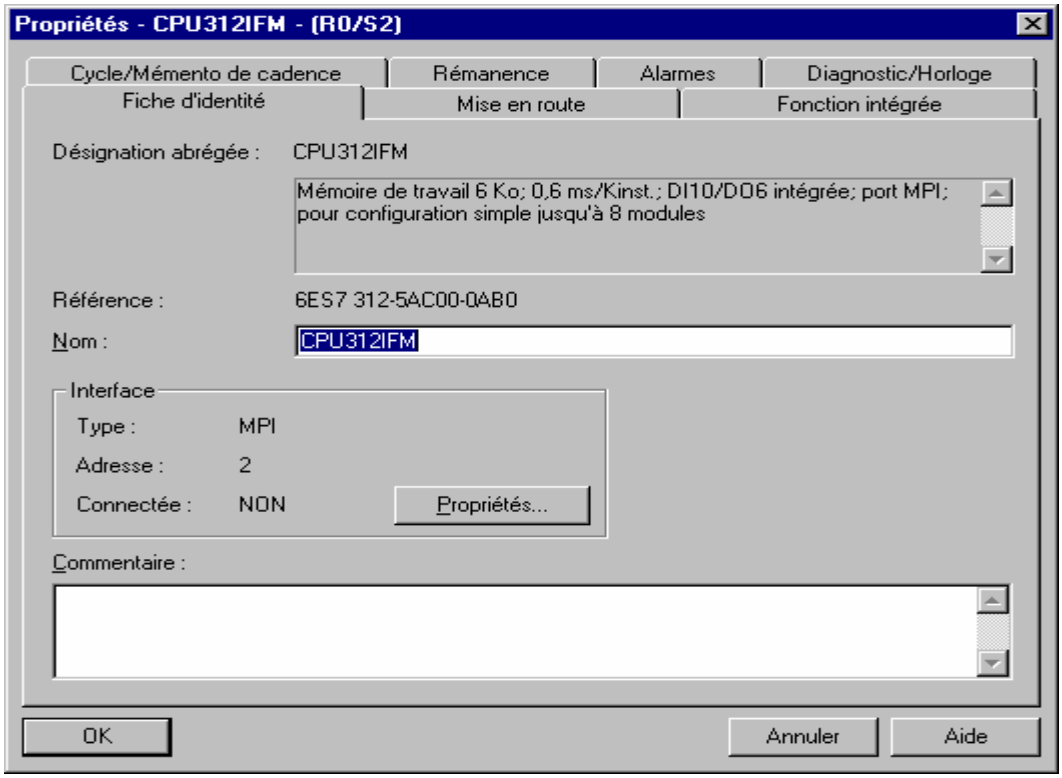
5.4.3 Paramétrage de la CPU.

Les paramètres suivants peuvent être définis sur la CPU :

- Adresse de l'interface MPI.
- Mise en route/ cycle : temps de cycle maximal, charge maximale du cycle pour assurer les fonctions de communications …
- Alarme cyclique
- Zones de rémanence : mémentos, temporisations, compteurs, blocs de données
- Mémentos de cadence : octet de mémento avec différents rythmes de clignotement
- Diagnostic du système

Si l'utilisateur ne procède à aucun paramétrage, le système fonctionne avec les réglages par défaut de la CPU.

Après le paramétrage, transférez les nouvelles valeurs avec l'option de menu Système cible => Charger ou en cliquant sur l'icône correspondante.



5.5 La programmation STEP 7.

5.5.1 Langages de programmation

5.5.1.1 List

La liste d'instructions est un langage de programmation textuel, orienté machine. LIST est le langage assembleur de STEP 5 et STEP 7. Dans un programme créé en LIST, chaque instruction correspond à une étape de traitement du programme par la CPU.

5.5.1.2 Cont

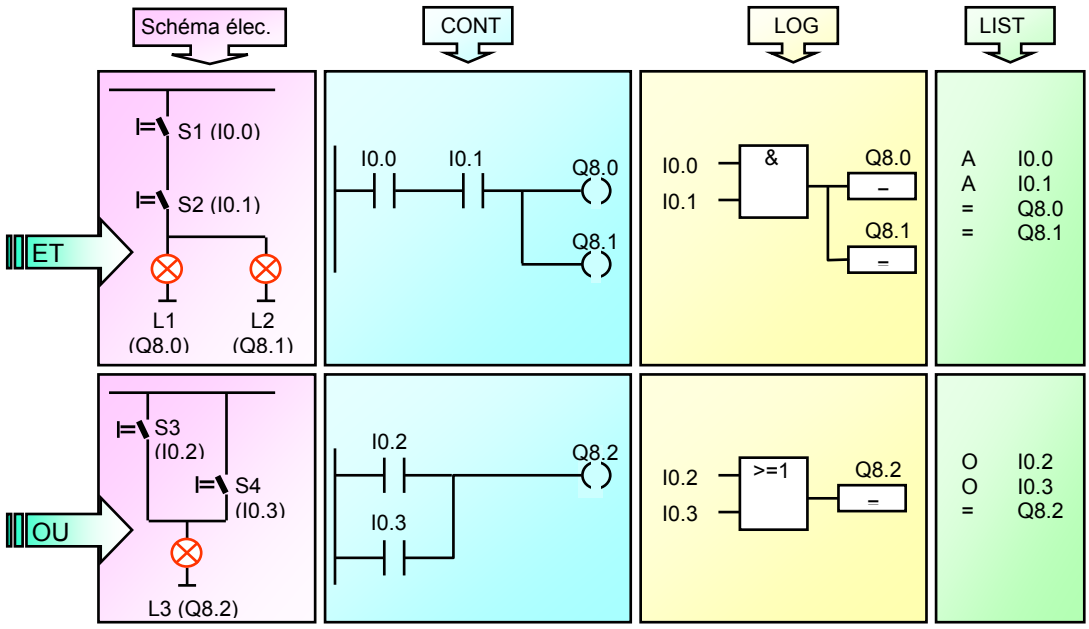
Le schéma à contacts est un langage de programmation graphique. Il s'agit de l'un des langages de programmation dans STEP 5 et STEP7. La représentation du schéma à contacts, conforme à la norme DIN 19239, correspondant à la représentation d'un schéma à relais. Contrairement à la liste d'instructions (LIST), le schéma à contacts ne permet pas de représenter l'entièreté du jeu d'opérations. Les éléments d'un tel schéma, comme par exemple les contacts à ouverture ou les contacts à fermeture sont reliés pour former des réseaux. Un ou plusieurs de ces réseaux forment la section d'instructions complète d'un bloc de code.

5.5.1.3 Log

Le langage de programmation LOG utilise les portes logiques bien connues en algèbre booléenne pour la représentation logique. Il permet en outre de représenter des fonctions complexes, telles que les fonctions mathématiques en les mettant directement en liaison avec ces portes logiques. La compilation d'un logigramme vers un autre langage de programmation (par exemple schéma à contact) est possible.
Le langage de programmation LOG fait partie du logiciel de base STEP 7 et répond à la norme CEI 1131-3.

5.5.2 Les opérations binaires.

5.5.2.1 Les fonctions combinatoires



5.5.2.2 Contact NO-NF

Contacts NF et contacts NO -
Capteurs et symboles d'interrogation

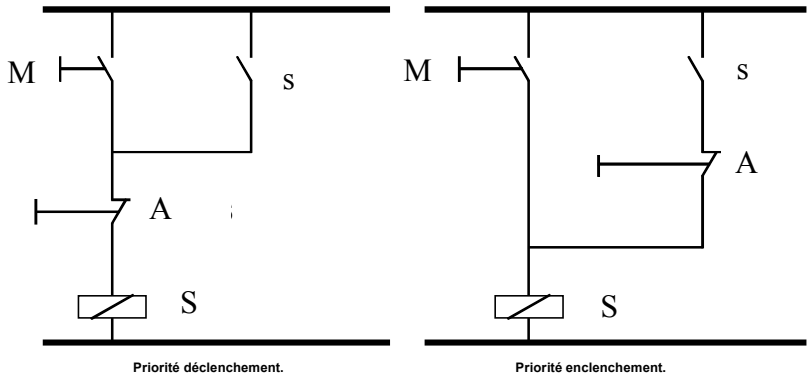
Processus			Evaluation du programme dans l'automate				
Le capteur est un ...	Le capteur est ...	La tension à l'entrée est ...	Etat du signal à l'entrée	Interrogation à "1"		Interrogation à "0"	
				Symbole / opération	Résult. logique	Symbole / opération	Résult. logique
contact NO	activé	présente	1	*CONT:* "contact NO"	"Oui" 1	*CONT:* "contact NF"	"Non" 0
	inactivé	absente	0	*LOG:* &	"Non" 0	*LOG:* &	"Oui" 1
contact NF	activé	absente	0		"Non" 0		"Oui" 1
	inactivé	présente	1	*LIST:* A I x.y	"Oui" 1	*LIST:* AN I x.y	"Non" 0

5.5.2.3 Les fonctions mémoires

5.5.2.3.1 Exemple : commande marche-arrêt d'un moteur :

Un bras manipulateur (S) est mis en marche et arrêté par deux boutons poussoirs séparés : M met en marche la machine qui peut seulement être arrêtée par l'action de A (le relâchement de M ne le peut pas).

Fonction mémoire avec auto-maintien



La représentation conventionnelle d'une fonction mémoire est l'auto-maintien. Un **contact auxiliaire** du relais est placé en parallèle du poussoir d'enclenchement M afin de maintenir le relais activé après le relâchement de ce poussoir. Par opposition aux entrées-sorties qui sont des variables externes du système, on voit ici la nécessité que le système lui-même élabore une variable secondaire pour construire la sortie (S).

Une action sur le poussoir de déclenchement A (contact d'ouverture) interrompt le maintien et le relais retombe. Il faut toutefois prendre garde que (même lors de l'emploi d'un automate programmable), le déclenchement doit s'effectuer par des contacts d'ouverture, pour d'évidentes raisons de sécurité : un câble ou un fin de course arraché doit provoquer un état de sécurité stable.

Pour le déclenchement du relais, il existe deux variantes selon lesquelles soit l'enclenchement soit le déclenchement est prioritaire [5] :

- Dans un déclenchement prioritaire, les contacts de déclenchement se placent en série avec la combinaison parallèle d'enclenchement.
- Dans un enclenchement prioritaire, les contacts de déclenchement se mettent en série avec l'auto-maintien, dans la branche parallèle de contact d'enclenchement.

Pour des raisons de sécurité, la priorité sera choisie au déclenchement pour des installations de mise en mouvement. Toute information d'arrêt (fin de course, arrêt d'urgence, défaut moteur,…) interrompt l'alimentation en énergie ; de même, en cas de coupure d'énergie, le relais n'étant plus alimenté, le contact d'auto-maintien (s) retombe. Il faut alors réarmer le système si l'on désire le remettre en fonctionnement (on utilise souvent le terme BP de réarment au lieu de mise en marche pour insister sur la priorité à l'arrêt).
Inversement, on choisira une priorité à l'enclenchement pour des dispositifs d'alarme, par exemple.

Ces systèmes auto-maintenus à arrêt prioritaire occupent une place prépondérante dans les systèmes automatisés : on les trouve systématiquement en tête de tout système d'alimentation en énergie des actionneurs (électriques, pneumatiques, …) des équipements, imposés par la normalisation européenne en matière de sécurité (normes EN292, EN954-1, EN60204-1).

5.5.2.4 Bascule RS de l'automate

Le bloc fonction qui réalise la mémorisation dans un système automatisé est la bascule RS. Elle possède deux états stables. Les deux entrées sont conventionnellement appelées :

- S pour set : entrée d'activation ou de mise à 1
- R pour Reset : entrée de désactivation ou de mise à zéro

Un signal 1 sur l'entrée Set met à "1" la fonction mémoire, alors qu'un signal 1 sur l'entrée Reset met à "0" la fonction mémoire. Un signal "0" aux entrées S R n'a pas d'influence.

[5] On parle de gestion de priorité lorsque les deux BP de mise à 1 et de mise à 0 sont enclenchées simultanément ! Dans certaine application, c'est de manière matérielle que l'on s'assurera que cette situation ne peut se produire (exemple : un switch 2 positions pour M et A).

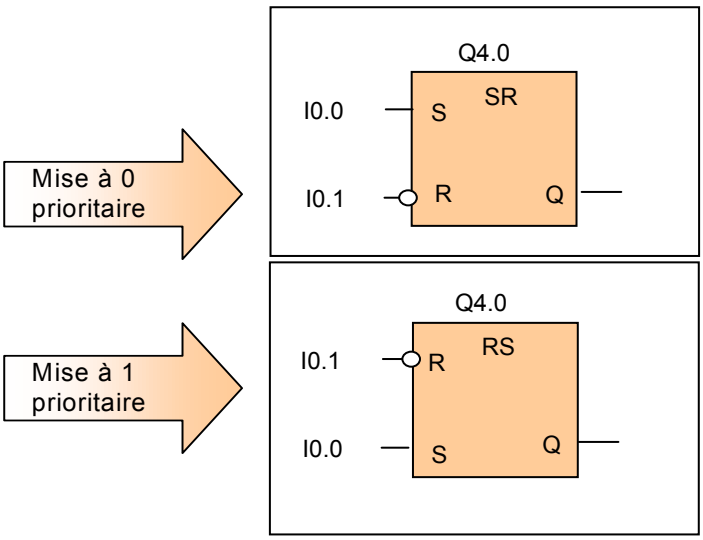
La bascule RS sera principalement utilisée lorsque l'on voudra maintenir l'effet d'un signal impulsionnel.

Il existe différentes variantes de cette bascule : lorsque les résultats logiques sur R,S sont simultanément à 1, il peut y avoir priorité à la mise à 0 ou à la mise à 1.

Dans le cas des automates Siemens :

Dans le réseau 1, l'opération de mise à 1(SET) est effectuée sur la sortie, suivie de l'opération de mise à 0 (RESET). L'activation de la sortie ne s'effectue que dans les mémoires images des sorties (en interne). En effet, ce n'est qu'à la fin du traitement cyclique que les mémoires images sont transférées aux sorties. La sortie sera donc mise dans l'état imposé par la dernière instruction qui l'a activée : la mise à 0.
Le câblage d'un contact d'ouverture sur pour le bouton de déclenchement impose une interrogation à 0 (ou inverseur) sur la mise à 0.

Par analogie, le réseau 2 montre une mise à 1 comme résultat final sur la sortie.

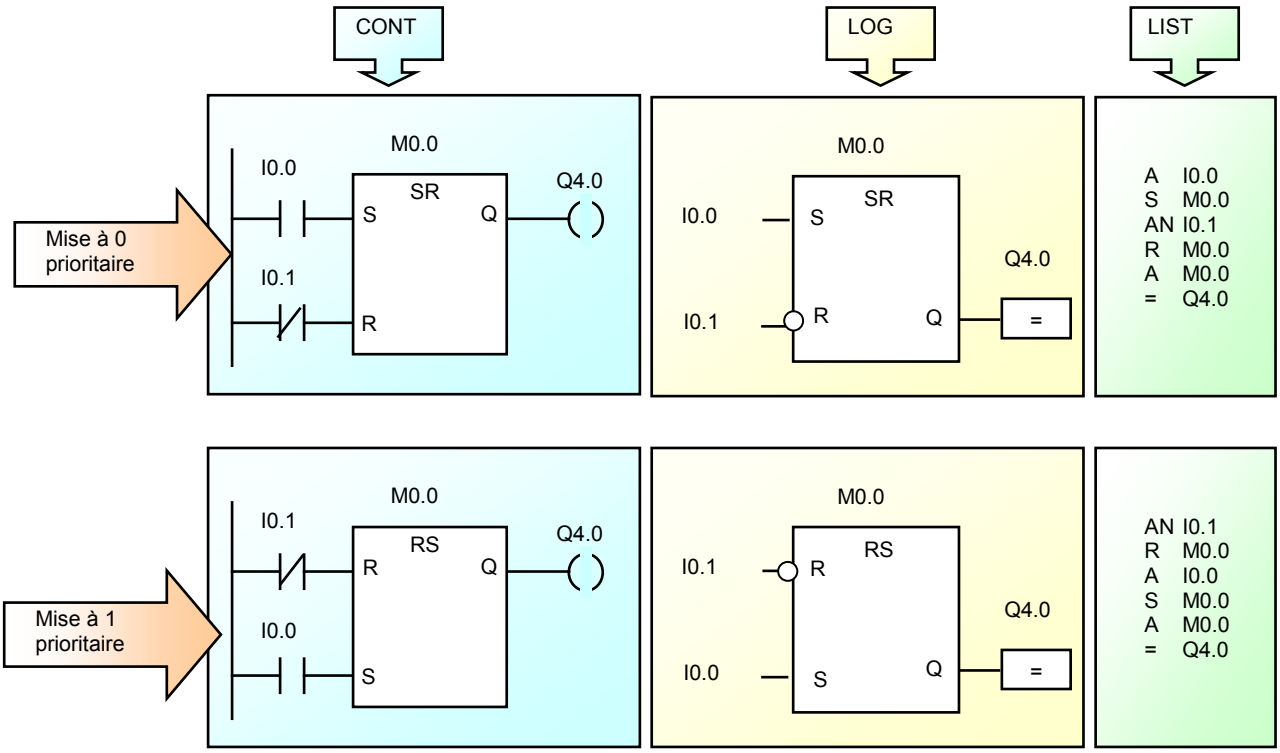


Une partie de la mémoire de données des constituants programmables est organisée pour le traitement logique. Chaque unité de mémoire, adressable séparément, peut être lue, écrite (mise à 1) ou effacée (mise à 0). On distingue deux types de mémoire :

- les mémoires « monostables », remises à 0 à chaque réinitialisation du système
- les mémoires « bistables » qui conservent leur état même en cas de coupure d'énergie.

Dans le cas des automates Siemens :
Les emplacements mémoires sont nommés mémentos et selon leur adresse, ils appartiennent à l'une ou l'autre famille.
On distingue les mémentos rémanents (bistables) et non rémanents qui sont remis à 0 à chaque phase d'initialisation.
Dans la configuration matérielle de la CPU, on peut configurer le nombre de mémentos rémanents et non rémanents. Par défaut, les 16 premiers bytes sont rémanents.

L'utilisation d'un mémento rémanent sur la fonction de mémorisation permettra donc de garder la sortie enclenchée en cas de rupture d'alimentation.
L'utilisation d'un mémento non-rémanent peut être utile si la fonction fait partie d'une combinaison logique plus complexe et n'enclenche pas directement une sortie.



5.5.2.5 Réponse aux fronts – RLG

Jusqu'à présent seule a été considérée la valeur (vraie ou fausse, pour une certaine durée) d'une proposition logique. Des propositions logiques comme « dcy enfoncé » ou « vérin en mouvement », il est possible de dériver d'autres propositions logiques comme « on vient d'enclencher le BP dcy » ou « le vérin arrête de se déplacer vers la droite » qui présentent la propriété de n'être vraie qu'à un instant bien précis (signal impulsionnel). Ces propositions traduisent des évènements : le début ou la fin d'un état logique, juste au moment du changement d'état, on détectera donc ce front ou flanc du signal.

On parle d'un front de RLG lorsque le résultat d'un signal ou d'une opération change.

A toute variable logique a, on peut associer deux variables logiques :

- a↑ : qui traduit le front montant de a
- a↓ qui traduit le front descendant de a

Dans le cas des automates Siemens :

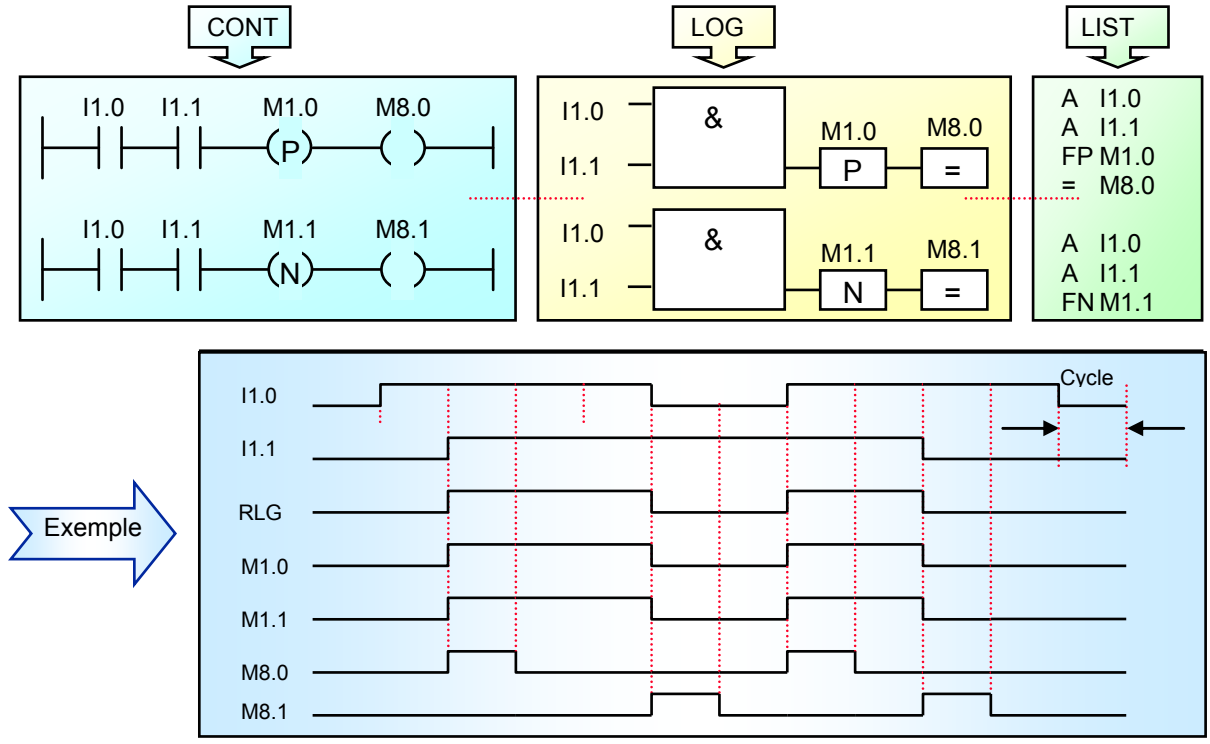
Le front montant est obtenu par la porte P :

- Lorsque le RLG passe de 0 à 1, l'interrogation « P » délivre un état logique à 1(dans l'exemple M8.0) pendant la durée d'un cycle.
  Pour que le système puisse détecter ce changement d'état, il faut que le RLG soit mémorisé dans un mémento (dans exemple M1.0)

et le front descendant par N :

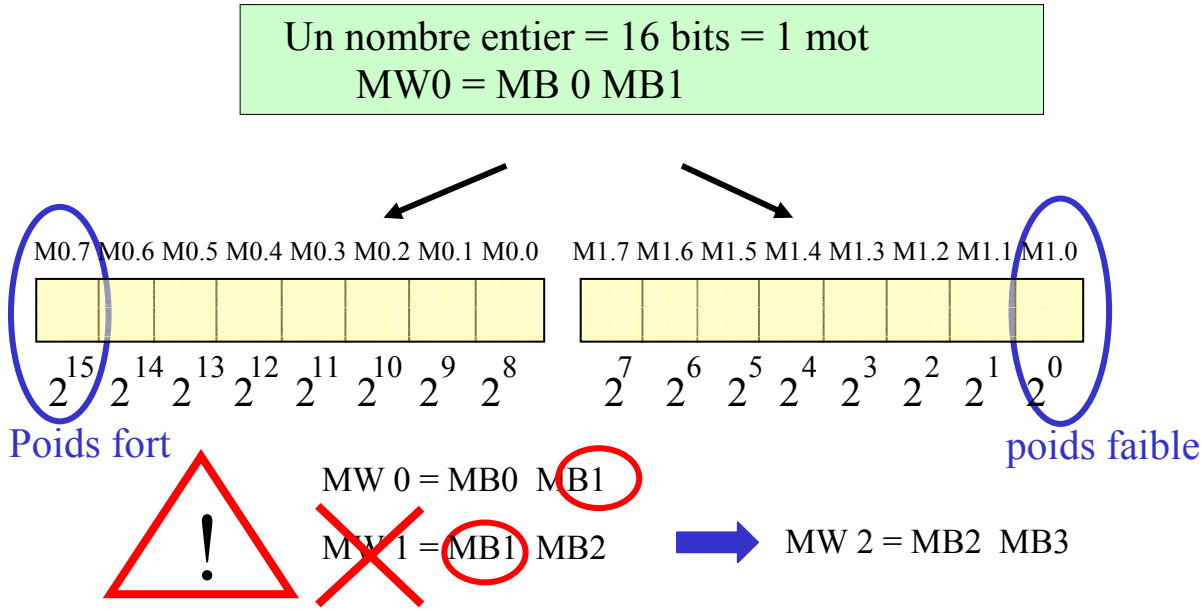
- Lorsque le RLG passe de 1 à 0, l'interrogation «N » délivre un état logique à 1(dans l'exemple M8.1) pendant la durée d'un cycle.
  Pour que le système puisse détecter ce changement d'état, il faut que le RLG soit mémorisé dans un mémento (dans exemple M1.1)

La détection de front est utilisée pour transformer un signal qui « dure » en un signal « impulsionnel ».



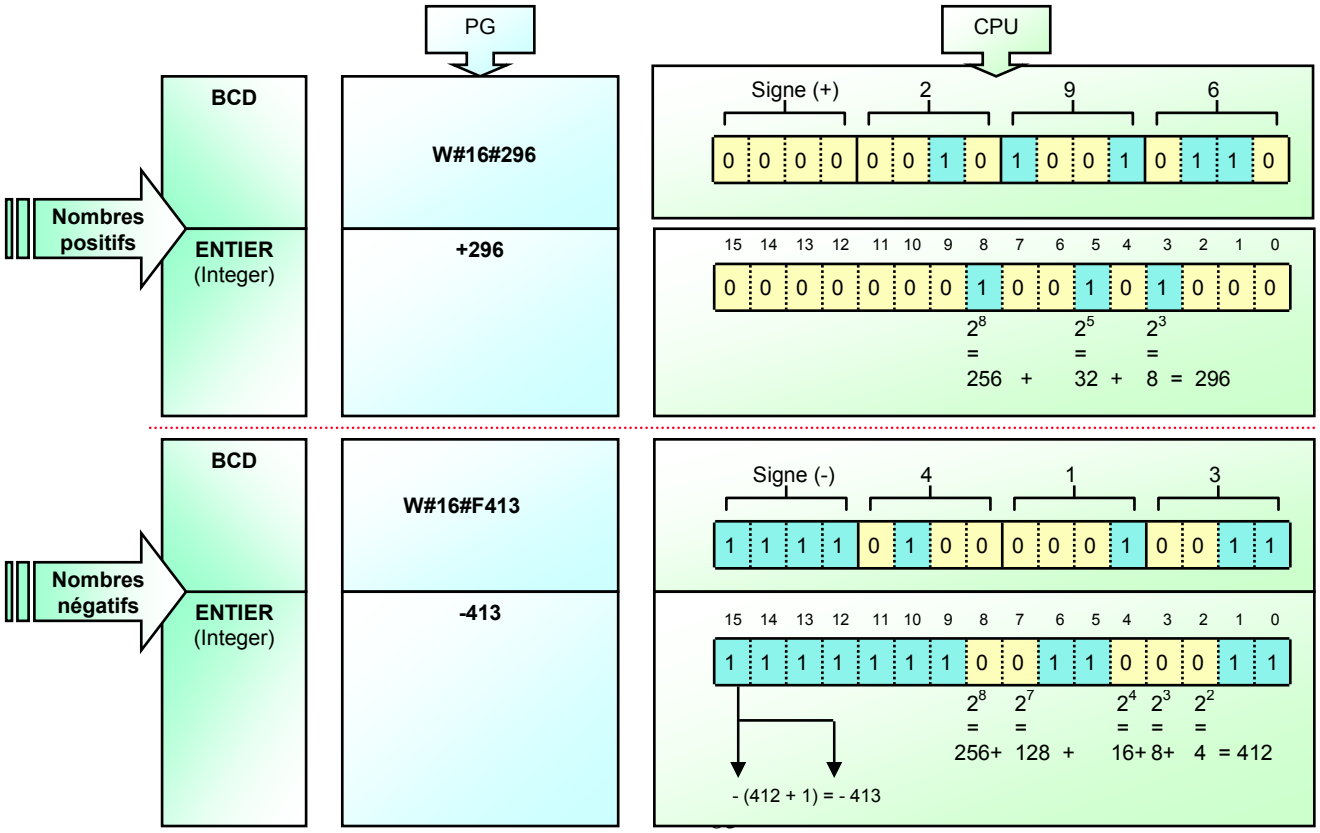
5.5.3 Les opérations numériques

5.5.3.1 L'adressage des nombres

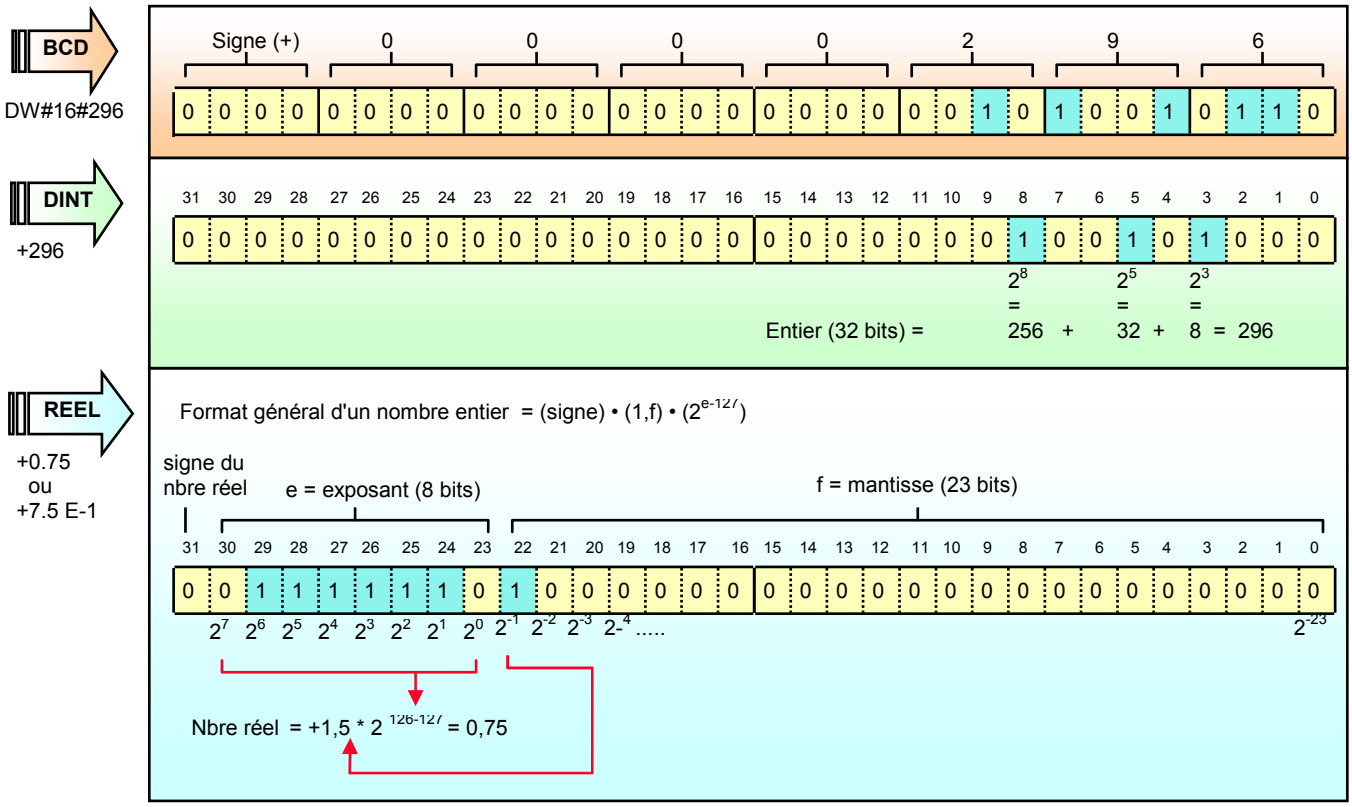


Double mot ou réel codé sur 32 bits
MD0 = MB0 MB1 MB2 MB3

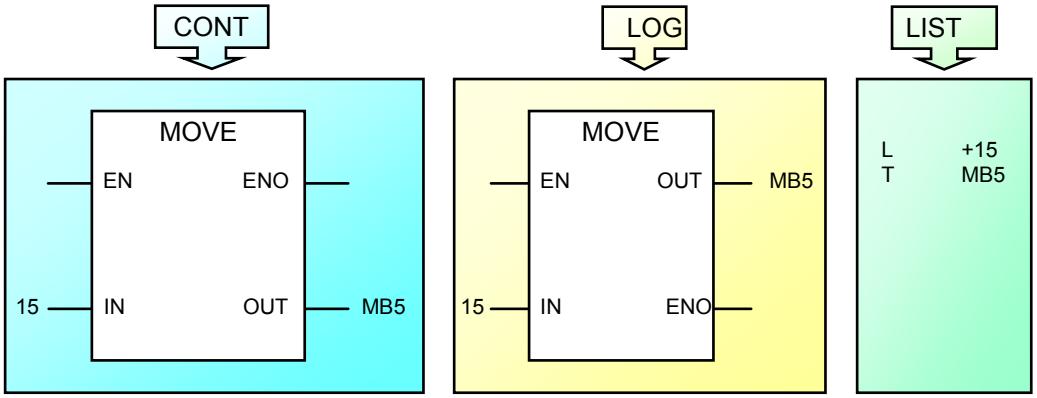
5.5.3.2 Les formats des nombres 16 bits



5.5.3.3 Les formats des nombres 32 bits



5.5.3.4 Chargement et transfert de données



Exemples de chargement

```
L   +5                          // Constante 16 bits (Integer)
L   L#523123                    // Constante 32 bits (Long Integer)
L   B#16#EF                     // Octet hexadécimal
L   2#0010 0110 1110 0011       // Valeur binaire 16 bits
L   3.14                        // Constante 32 bits (Real)
```

5.5.3.5 Les opérations de comptages

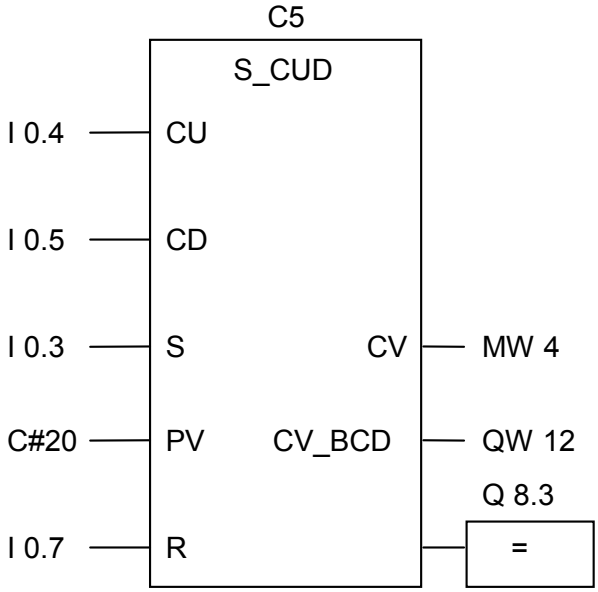
La fonction comptage peut être obtenue par un dispositif « matériel » (électronique ou autre) ou plus simplement, dans l'automate, par une fonction « logicielle » qui permet l'incrémentation d'un emplacement mémoire n→n+1.

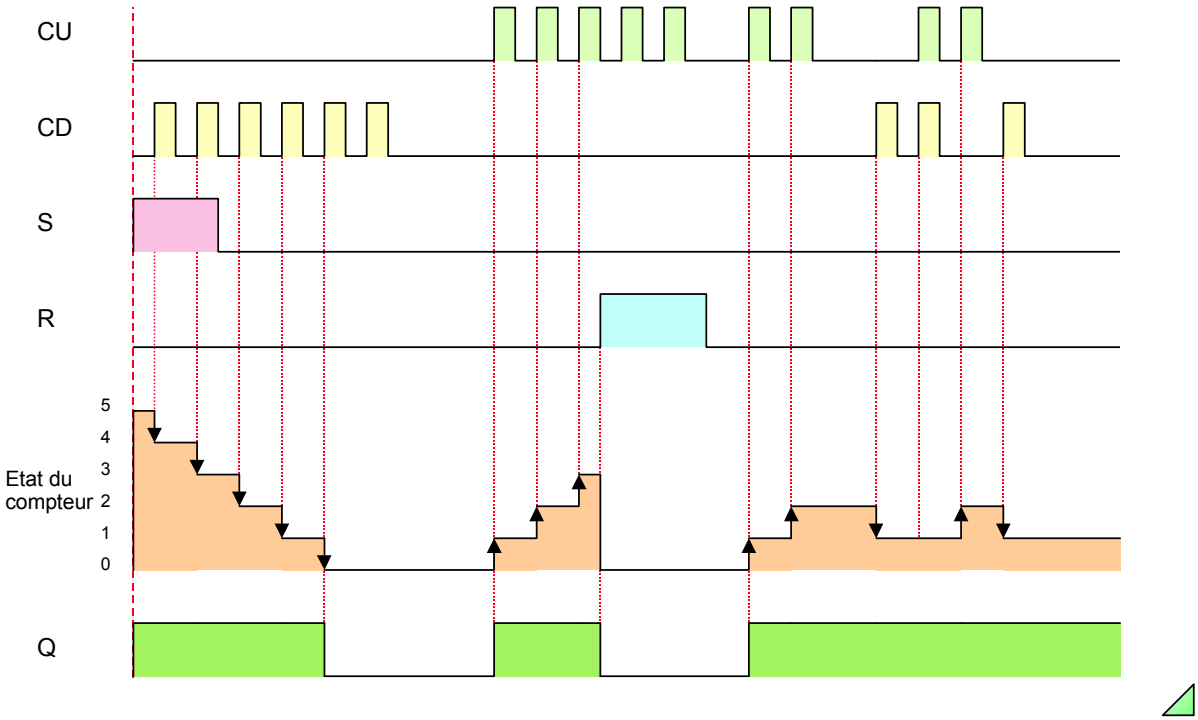
Le comptage/décomptage est une fonction utilisée par les automates pour convertir une suite d'occurrence logique (changements de valeurs de IN) en :

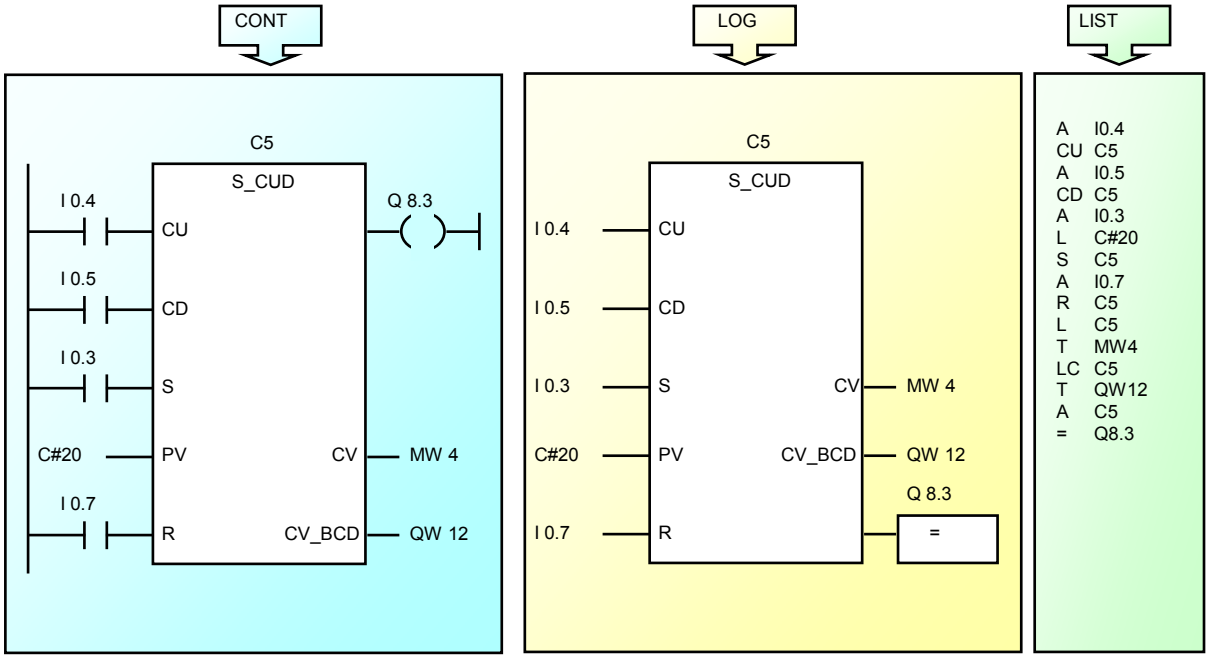
- Une information numérique : codage du nombre de fois que IN aura rencontré une transition montante (par exemple sur 3 sorties logiques dans l'exemple ci-dessous)
- Une information logique qui indique si le compteur est >0 ou à 0..

Comme dans le bloc ci-dessous (Siemens S7), les compteurs disposent bien souvent :

- D'une entrée de comptage (**c**ount **u**p) : l'opération +1 se réalise au flanc montant du signal d'entrée CU
- D'une entrée de décomptage (count **d**own) : l'opération –1 se réalise au flanc montant de CD
- D'une entrée de remise à 0 (clear ou reset)
- D'une entrée d'initialisation (S) à une valeur fixée par mot d'entrée PV
- D'un mot de sortie qui donne un code binaire (base la mieux adaptée au comptage/décomptage et à l'exploitation de comparateurs ou autres opérateurs arithmétiques)
- D'un mot de sortie qui donne le code en BCD (base la mieux adaptée à l'affichage)
- D'une sortie logique qui signale l'état du compteur (0 si le compteur est à 0, 1 si le compteur est à une valeur >0)




CU
CD
S
R
Etat du compteur
5
4
3
2
1
0
Q


CONT
LOG
LIST
C5
S_CUD
I 0.4
CU
Q 8.3
I 0.5
CD
I 0.3
S
C#20
PV
CV
MW 4
I 0.7
R
CV_BCD
QW 12
Q 8.3
=
A I0.4
CU C5
A I0.5
CD C5
A I0.3
L C#20
S C5
A I0.7
R C5
L C5
T MW4
LC C5
T QW12
A C5
= Q8.3

5.5.3.6 Les opérations de comparaisons

CMP : Les opérations de comparaison permettent de comparer les valeurs numériques suivantes:

I deux nombres entiers (de 16 bits chacun à virgule fixe)

D deux nombres entiers (de 32 bits chacun à virgule fixe)

R deux nombres à virgule flottante (nombres réels de 32 bits = nombres à virgule flottante IEEE).

Si la comparaison est "vraie", le RLG est égal à 1, dans le cas contraire, il est égal à "0".

La comparaison porte sur les entrées IN1 et IN2, selon le type d'opération sélectionné:

== IN1 est égale à IN2

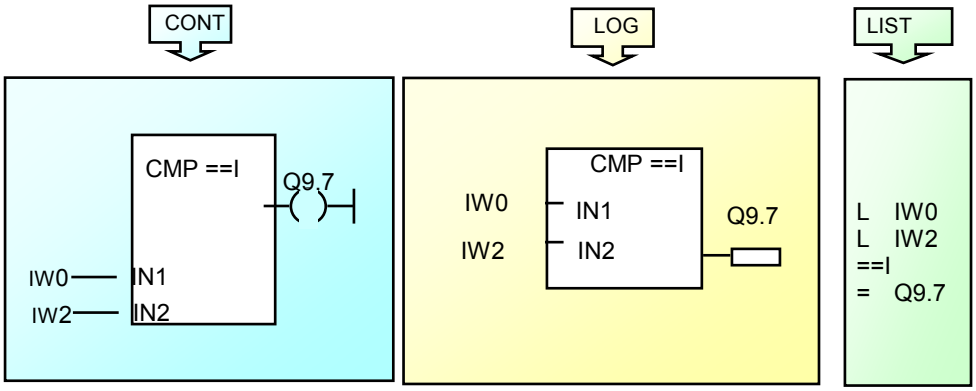
<> IN1 est différente de IN2

> IN1 est supérieure à IN2

< IN1 est inférieure à IN2

>= IN1 est supérieure ou égale à IN2

<= IN1 est inférieure ou égale à IN2

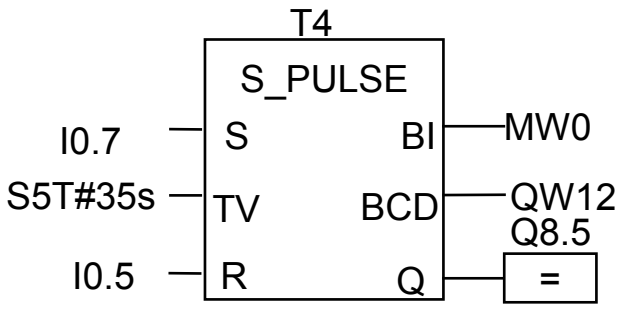


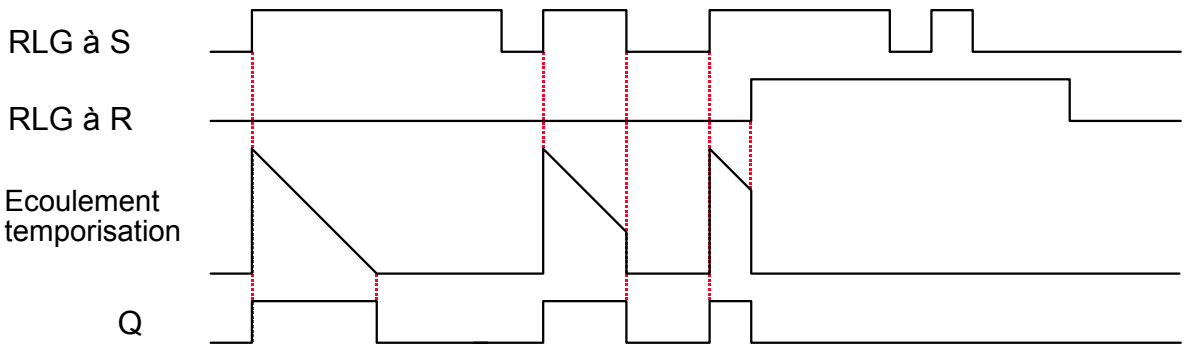
5.5.3.7 Les temporisations

Une temporisation est une fonction de comptage un peu particulière : ce que l'on compte, ce sont des périodes de temps, on parle de *timer* en anglais. Le signal d'entrée est donc une base de temps bien précise (interne à l'automate) et non un signal externe.

Il existe différentes variantes des temporisations (retard à l'enclenchement, au déclenchement, calibrage, …) mais, de manière générale, les tempos disposent bien souvent :

- D'une entrée d'activation : la temporisation démarre lorsque ce signal d'entrée (S) start rencontre le front auquel elle est sensible
- D'une entrée (TV) de chargement de la durée (mot) et de spécification de la base de temps employée
- D'une entrée de remise à 0 (clear ou reset)
- D'un mot de sortie qui donne un code binaire
- D'un mot de sortie qui donne le code en BCD
- D'une sortie logique, dont la valeur dépend du type de tempo et du temps (complètement écoulé ou non)



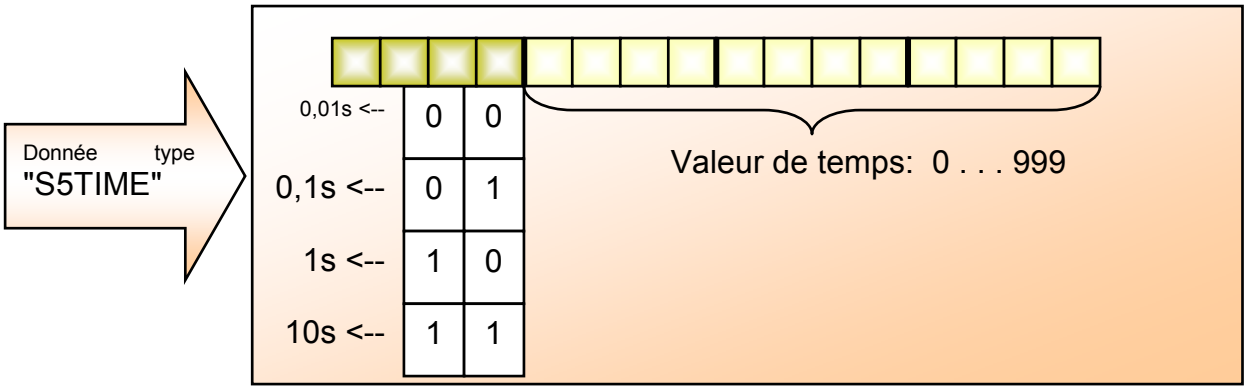


Par exemple dans le cas des automates Siemens :

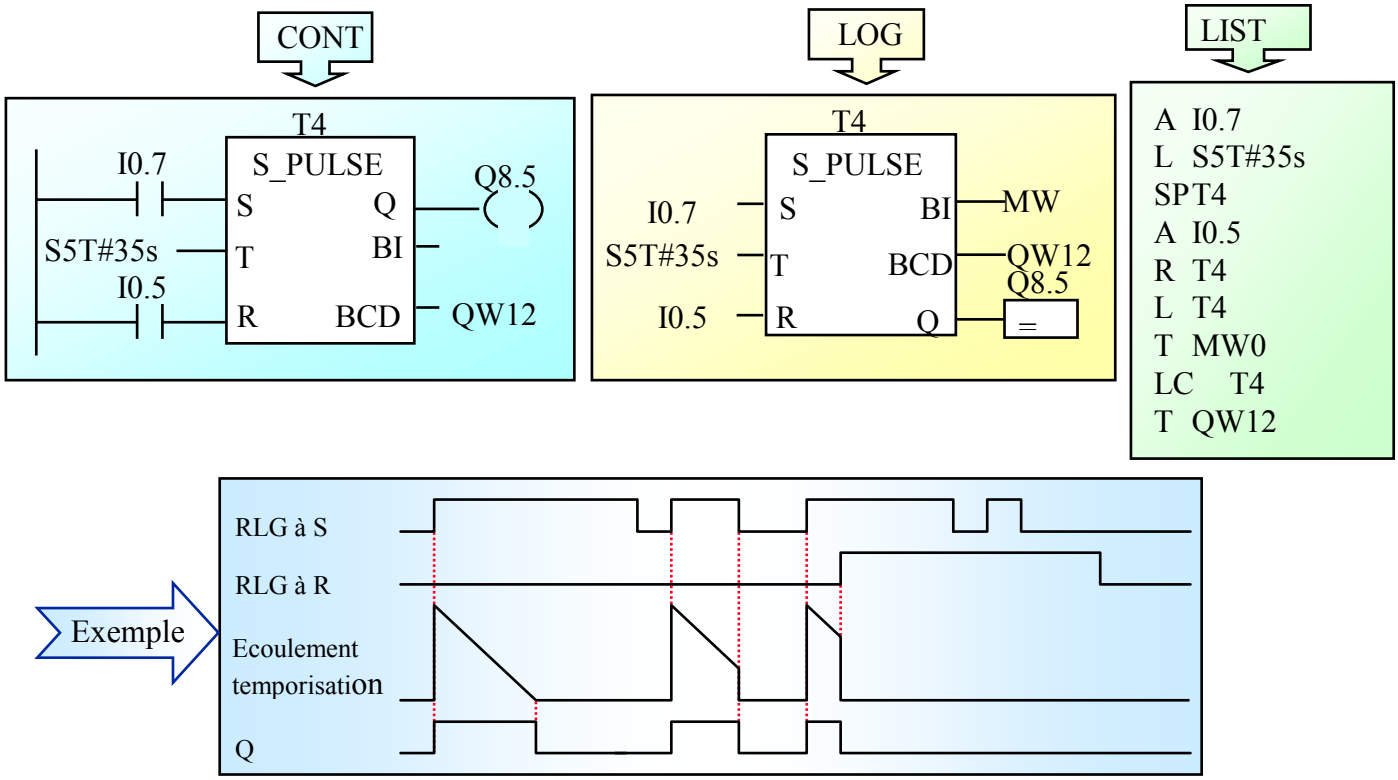
Il existe les différents types de temporisation :

S_IMPULS S_PULSE temporisation sous forme d´impulsion prolongée Limiteur d'impulsion	Le signal de sortie reste à 1 pendant la durée programmée si le signal d'entrée reste à 1.
S_VIMP S_PEXT temporisation sous forme d´impulsion prolongée calibreur d'impulsion	Le signal de sortie reste à 1 pendant la durée programmée, quelle que soit la durée pendant laquelle le signal d'entrée reste à 1.
S_EVERZ S_ODT temporisation sous forme de retard à la montée Retard à l'enclenchement	Le signal de sortie est égal à 1 uniquement lorsque le temps programmé s'est écoulé et que le signal d'entrée est toujours à 1.
S_SEVERZ S_ODTS temporisation sous forme de retard à la montée mémorisé Retard à l'enclenchement mémorisé	Le signal de sortie passe de 0 à 1 uniquement lorsque le temps programmé s'est écoulé, quelle que soit la durée pendant laquelle le signal d'entrée reste à 1.
S_AVERZ S_OFFDT temporisation sous forme de retard à la retombée Retard au déclenchement	Le signal de sortie est égal à 1 lorsque le signal d'entrée est égal à 1 ou lorsque la temporisation s'exécute. La temporisation est démarrée lorsque le signal d'entrée passe de 1 à 0.

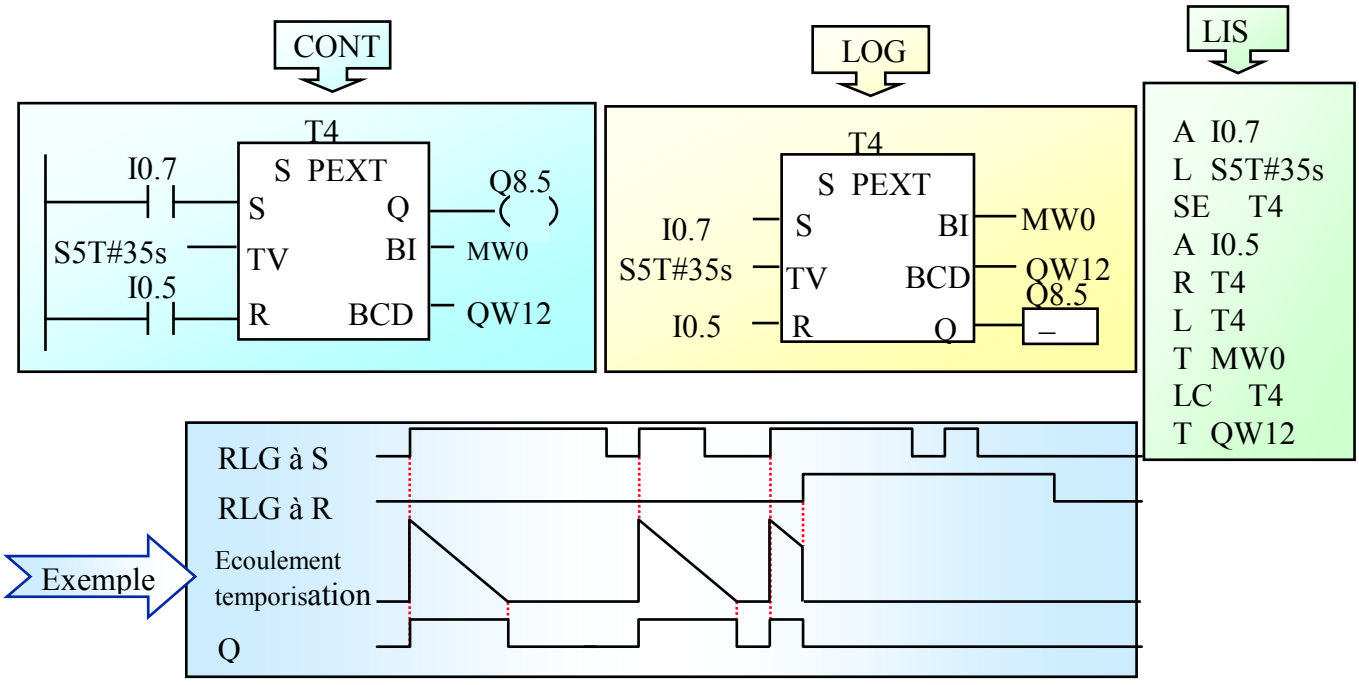
Format du temps du temps



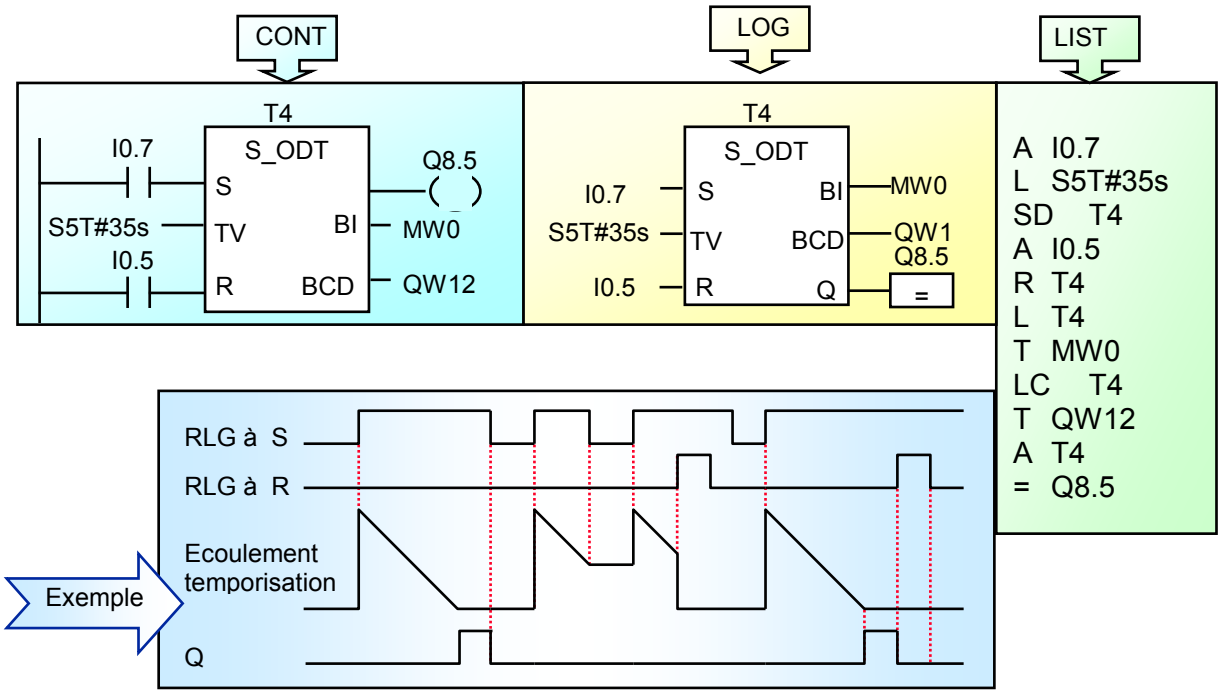
Temporisation sous forme d'impulsion (SP)



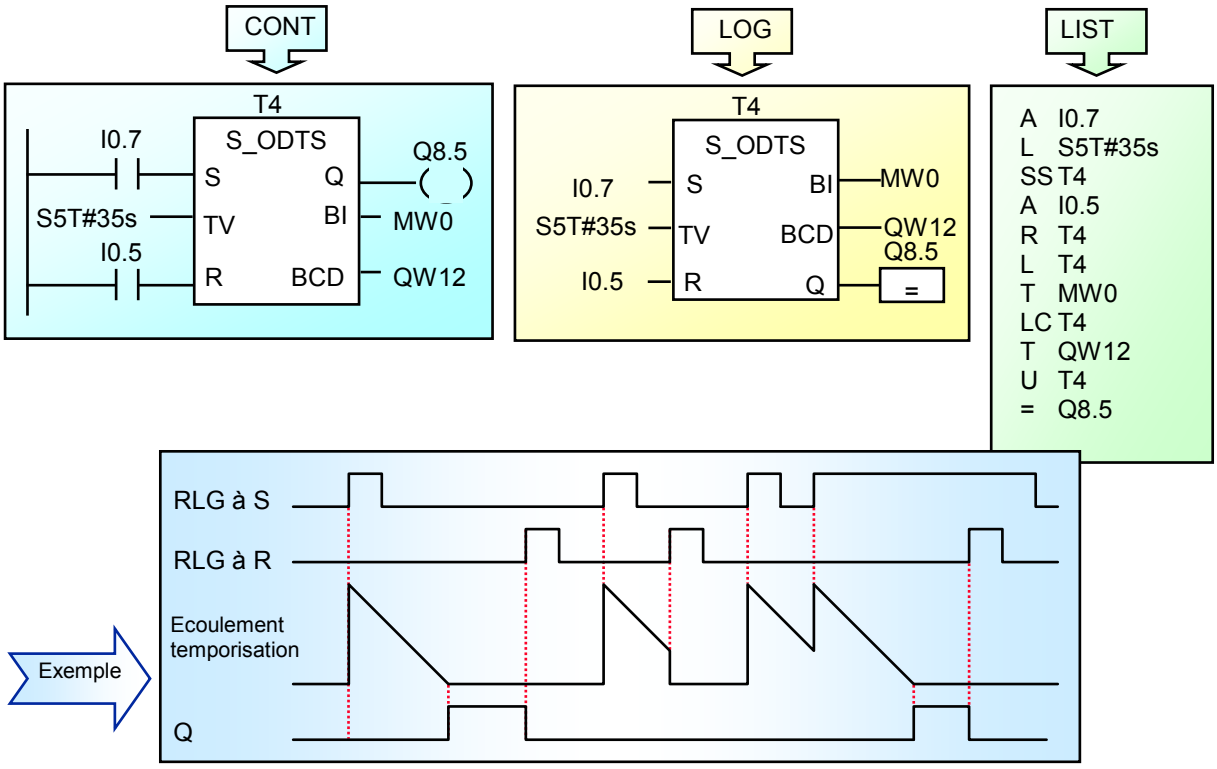
Temporisation sous forme d'impulsion prolongée (SE)



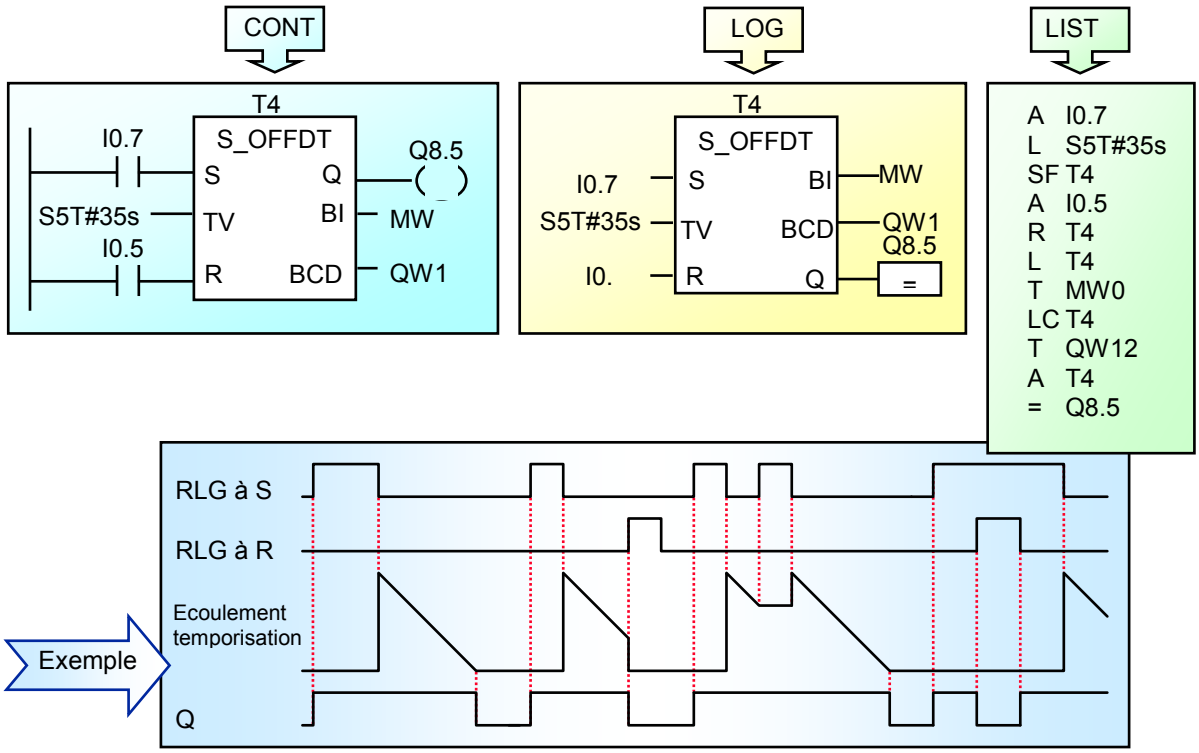
Temporisation sous forme de retard à la montée (SD)



Temporisation sous forme de retard à la montée mémorisé (SS)



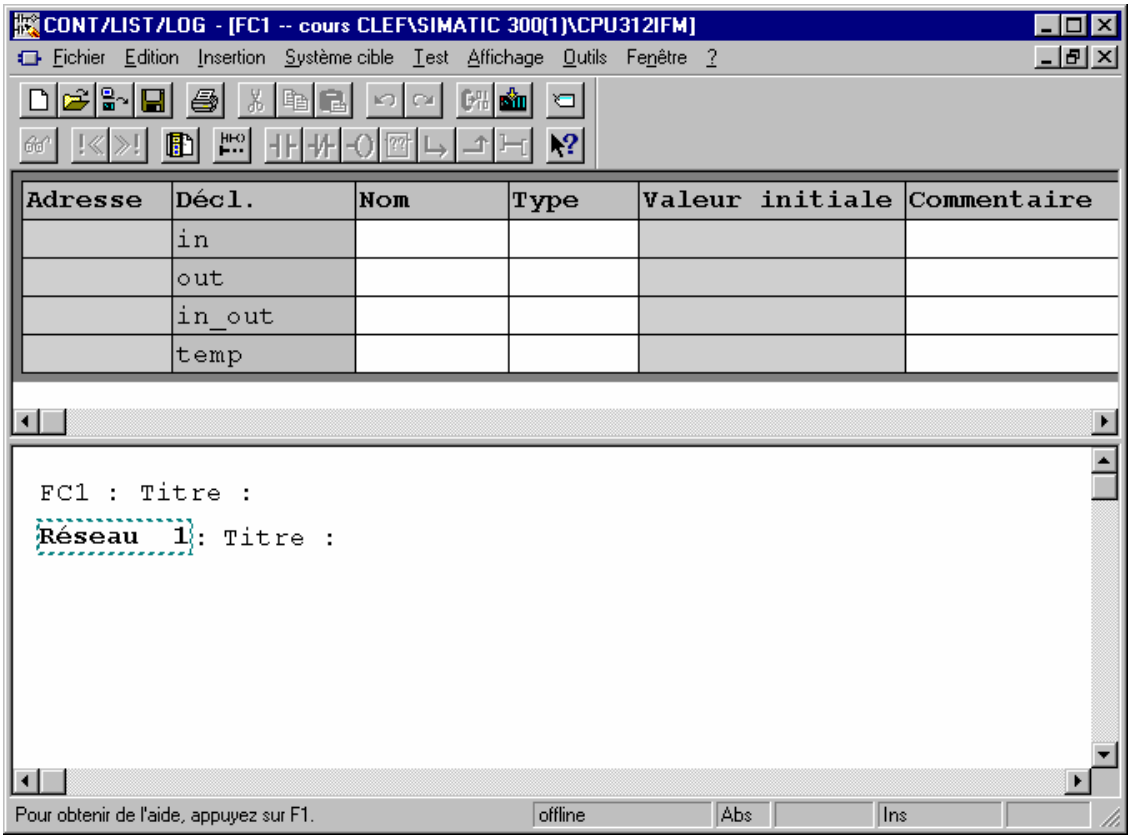
Temporisation sous forme de retard à la retombée (SF)



5.5.4 Les variables et l'adressage

5.5.4.1 Composantes d'un bloc

Lorsqu'un bloc est ouvert, deux fenêtres apparaissent dans l'éditeur de programmes. Une fenêtre contient la table de déclaration des variables, l'autre comporte la partie instruction dans laquelle est écrit le programme proprement dit.

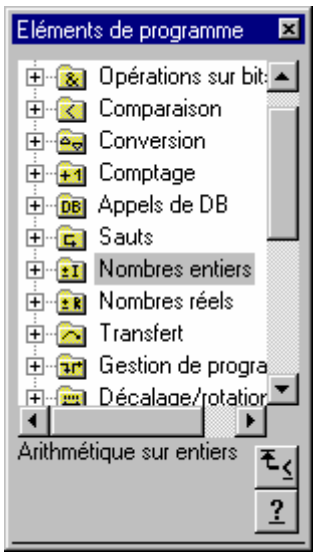


La table de déclaration des variables sert à définir le nom et la taille des variables locales au bloc. Par opposition aux variables globales, les variables locales aux blocs sont uniquement utilisables dans les blocs dans lesquels elles sont déclarées. Les mnémoniques locaux de bloc sont donc définis dans la partie déclarative du bloc. Cependant, il est possible d'utiliser plusieurs fois les mêmes mnémoniques car ils ne s'appliquent qu'au bloc concerné. Les mnémoniques peuvent être déclarés comme paramètres, comme variables locales au bloc ou comme repères de saut.
Nous pouvons déclarer des variables de types opérandes formels qui seront associés à des opérandes actuels lors de l'appel du bloc, ces paramètres permettent les échanges des valeurs entre le bloc appelant et la partie du programme dans lequel il est appelé. Il existe plusieurs type de paramètres locaux utilisables: in (paramètre d'entrée en lecture), out (paramètre de sortie en écriture), in_out (paramètre d'entrée/sortie en lecture ou en écriture).
Les variables statiques(FB) stat sont mémorisées dans le DB d'instance.

La déclaration des variables locales temporaires temp permet de réserver une place dans la pile des données locales. La valeur d'une variable temporaire n'est pas mémorisée après l'exécution du bloc.

Remarque: lors de l'utilisation de mnémoniques locaux, le compilateur renseigne s'il s'agit de variables locales ou globales. En effet, une variable locale sera écrite de la façon suivante : #nom. Pour une variable globale, l'écriture sera : ″nom″. Cette remarque est très importante notamment lorsque l'on emploie le même nom pour 2 variables de types différents.

La partie instruction sert, quant à elle, à écrire le programme. Après la saisie d'une instruction, l'éditeur de programmes procède à un contrôle de syntaxe et indique en italique rouge les entrées erronées.



Pour la programmation en LOG, l'utilisateur dispose d'un catalogue dans lequel il n'a plus qu'à choisir les instructions.
Une aide succincte sur l'élément sélectionné est affichée dans le bas du catalogue. Pour une aide plus complète il suffit de taper F1.

5.5.4.2 Déclaration des variables

Les différentes variables utilisables pour la programmation des SIMATIC S7 sont :

- L'entrée (I) = zone de la mémoire système de la CPU (mémoire image des entrées).
- La sortie (Q) = zone de la mémoire système de la CPU (mémoire image des sorties).
- Le mémento (M) = zone de la mémoire système de la CPU accessible en écriture et en lecture (par bit, octet, mot et double mot)

- La temporisation (T) = composante de la mémoire système de la CPU. Son contenu est actualisé par le système d'exploitation de manière asynchrone par rapport au programme utilisateur.
- Le compteur (C) = composante de la mémoire système de la CPU. Son contenu peut être modifiée à l'aide d'instructions STEP 7 (exemple : incrémentation ou décrémentation).

Le type de données permet de définir de quelle manière doit être utilisée la valeur d'une variable ou d'une constante dans le programme utilisateur. Dans SIMATIC S7, l'utilisateur dispose de deux sortes de types de données, conformément à la norme CEI 1131-3 :

a) Le type de données élémentaire est le type de données prédéfini, comme par exemple le type de données
   - BOOL qui définit une variable binaire ("Bit").
   - INT qui définit une variable à virgule fixe de 16 bits.

b) Le type de données complexe permet à l'utilisateur de définir des données du type tableau ou structure notamment.
   - Un tableau (ARRAY) est composé d'éléments de données de même type, qui eux-mêmes peuvent être élémentaires ou complexes.
   - Une structure (STRUCT) est composé d'éléments de données de type différents. Chacun de ces éléments de données peut être élémentaire ou complexe.

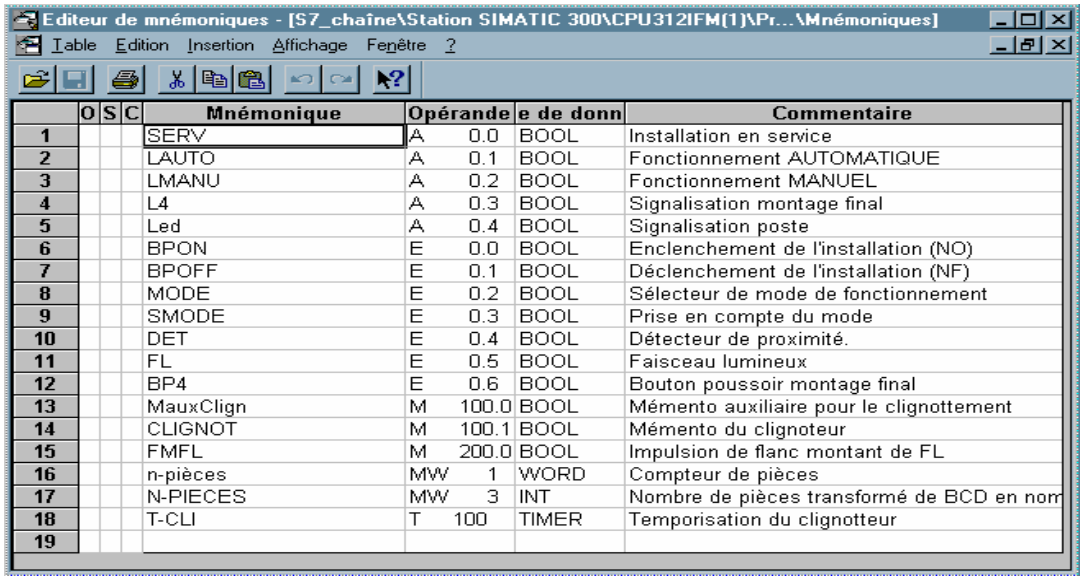
5.5.4.3 Adressage absolu et symbolique

En adressage absolu, l'adresse est indiquée directement, par exemple, nous écrivons l'adresse de l'entrée 1.0 par I 1.0. Dans ce cas, il n'est pas nécessaire de recourir à une liste de mnémoniques, mais le programme est plus difficile à lire.

L'adressage symbolique permet de travailler avec des mnémoniques, par exemple MOTEUR_MARCHE, au lieu d'utiliser des adresses. Les mnémoniques pour les entrées, les sorties, les temporisations, les compteurs, les mémentos et les blocs sont stockés dans la liste des mnémoniques. Dans ce cas, on parle de mnémoniques globaux, car un accès symbolique aux même éléments est possible à partir de tous les blocs.

Par opposition à la symbolique globale, il existe des mnémoniques locaux, valables à l'intérieur d'un seul bloc. Les mnémoniques locaux sont définis dans la partie déclarative du bloc et ne sont utilisables que dans le bloc dans lequel ils sont déclarés.

5.5.4.4 Editeur de mnémoniques



	O	S	C	Mnémonique	Opérande		e de donn	Commentaire
1				SERV	A	0.0	BOOL	Installation en service
2				LAUTO	A	0.1	BOOL	Fonctionnement AUTOMATIQUE
3				LMANU	A	0.2	BOOL	Fonctionnement MANUEL
4				L4	A	0.3	BOOL	Signalisation montage final
5				Led	A	0.4	BOOL	Signalisation poste
6				BPON	E	0.0	BOOL	Enclenchement de l'installation (NO)
7				BPOFF	E	0.1	BOOL	Déclenchement de l'installation (NF)
8				MODE	E	0.2	BOOL	Sélecteur de mode de fonctionnement
9				SMODE	E	0.3	BOOL	Prise en compte du mode
10				DET	E	0.4	BOOL	Détecteur de proximité.
11				FL	E	0.5	BOOL	Faisceau lumineux
12				BP4	E	0.6	BOOL	Bouton poussoir montage final
13				MauxClign	M	100.0	BOOL	Mémento auxiliaire pour le clignottement
14				CLIGNOT	M	100.1	BOOL	Mémento du clignoteur
15				FMFL	M	200.0	BOOL	Impulsion de flanc montant de FL
16				n-pièces	MW	1	WORD	Compteur de pièces
17				N-PIECES	MW	3	INT	Nombre de pièces transformé de BCD en nom
18				T-CLI	T	100	TIMER	Temporisation du clignotteur
19								

La figure ci-dessus montre l'éditeur de mnémoniques.

La barre d'outils propose plusieurs boîtes de sélection : agrandir ou réduire, critère de tri par Mnémoniques ou par Adresses, Affichage de tous les mnémoniques ou des mnémoniques univoques ou non univoques par filtres.
La symbolique doit être univoque au sein du programme utilisateur ; en d'autres termes, un opérande absolu ou un mnémonique ne doit apparaître qu'une seule fois dans la liste de mnémoniques. Si plusieurs mnémoniques ou opérandes absolus identiques apparaissent simultanément dans la liste des mnémoniques, ils sont représentés avec une autre police lorsque le mode d'affichage « Tout » a été sélectionné. Ce cas de figure peut se présenter lorsque l'on a copié une ligne afin d'éviter de saisir plusieurs fois des affectations similaires.
Pour retrouver rapidement ces mnémoniques équivoques (non univoques) dans une longue liste de mnémoniques, il est possible de les visualiser avec l'option d'affichage « Non univoque ».

Il est par ailleurs possible d'enregistrer d'autres formats de fichiers et de les exporter aux formats DIF, SDF, ASCII et ZULI. Ceci permet de reprendre des listes d'assignation et de mnémoniques, mais aussi de réutiliser la liste des mnémoniques avec un éditeur de textes quelconque.

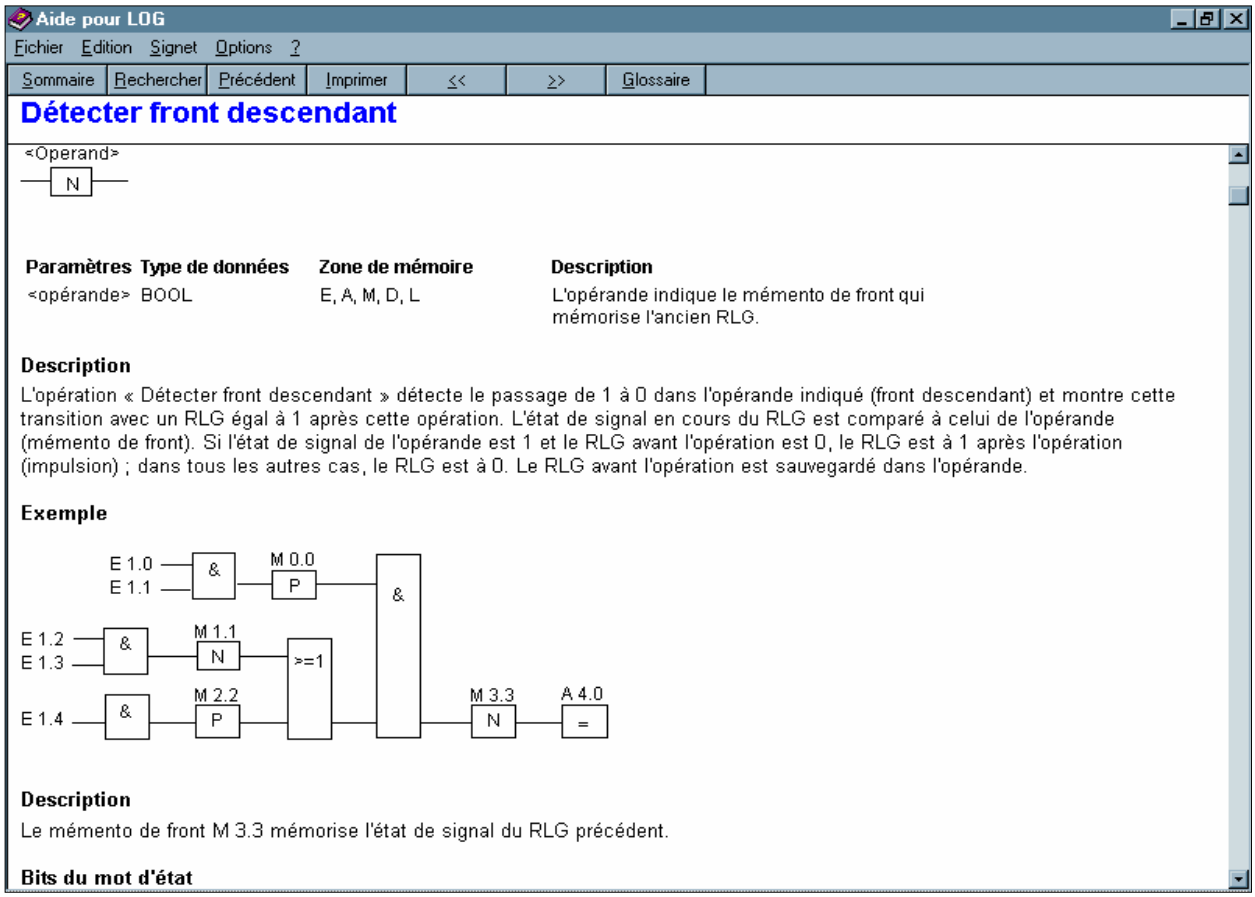
5.5.5 La fonction d'aide

Le logiciel STEP7 dispose d'un important système d'aide qui comprend les éléments suivants :

- menus d'aide : le sommaire et l'index vous permettent d'accéder aux différentes rubriques d'aide.
- boutons d'aide : les différentes boîtes de dialogue comportent des boutons d'aide.
- la touche F1 permet d'appeler le système d'aide à tout moment.

Pour faciliter la programmation, on dispose d'un catalogue des fonctions. La sélection d'une de ces fonctions affiche en bas du catalogue, en deux ou trois mots succincts, le nom ainsi que son rôle.

Pour avoir des informations complémentaires, une pression sur la touche F1 fait apparaître une fenêtre d'aide telle que celle ci-dessous.



5.5.6 Fonctions de diagnostique.

5.5.6.1 Visualisation de l'état du programme

L'éditeur CONT/LIST permet de visualiser l'état du programme et le trajet des signaux dans chacun des langages de programmation. En CONT ou LOG, l'état du programme visualise le trajet du courant entre les différents éléments et les valeurs d'entrée et de sortie d'un bloc. En LIST, l'état du programme visualise l'opérande, le RLG et les registres importants pour chaque instruction du programme.

Lancement de Etat du programme :

1. Ouvrir le bloc de programme en ligne avec l'outil Programmation de blocs (CONT/LIST).
2. Sélectionner Test➔Visualiser (dans le menu de l'éditeur CONT/LIST).
3. Résultat : l'état est actualisé à partir du réseau sélectionné.

Il existe deux modes de fonctionnement pour la **fonction Etat de bloc**. Il est ainsi possible de sélectionner les modes Processus et Laboratoire pour un bloc ouvert en ligne. Dans le Mode Processus, l'état des instructions figurant dans les boucles programmées n'est déterminé que lors du premier passage. Ce mode correspond à la plus faible charge du temps de cycle. Par contre, dans le mode Laboratoire, l'état des instructions figurant dans les boucles programmées est déterminé à chaque passage. Ce mode augmente considérablement la charge du temps de cycle.

5.5.6.2 Visualisation et forçage de variables

L'outil Visualisation et forçage de variables permet d'afficher les opérandes d'un programme. Cet outil offre en outre la possibilité de définir des tables dans lesquelles les variables peuvent être visualisées dans différents formats. Les variables sont actualisées indépendamment du programme. Cette caractéristique autorise ainsi la recherche des défauts matériels de l'API, par exemple sur les E/S.

L'option Système cible ➔ Visualiser et forcer des variables permet également de les modifier en ligne dans la CPU. Il est possible de procéder à l'actualisation des variables modifiées en début de cycle, en fin du cycle et à la transition RUN➔STOP.

5.5.6.3 Diagnostic et tests

On appelle « diagnostic » les fonctions intégrées de détection et d'enregistrement des erreurs de la CPU. La zone dans laquelle sont enregistrées les informations d'erreur est appelée « tampon de diagnostic ». La taille du tampon est fonction de la CPU. La CPU procède à son diagnostic à chaque scrutation cyclique. Si une erreur ou un événement survient, les conséquences sont les suivantes :

1. Un message horodaté est inscrit dans le tampon de diagnostic. Le dernier message entré est enregistré au début du tampon. Lorsque le tampon est plein, les inscriptions les plus anciennes sont effacées.
2. Le tampon enregistre la description de l'événement de diagnostic.
3. L'événement active le cas échéant un OB (bloc d'organisation) d'erreur.

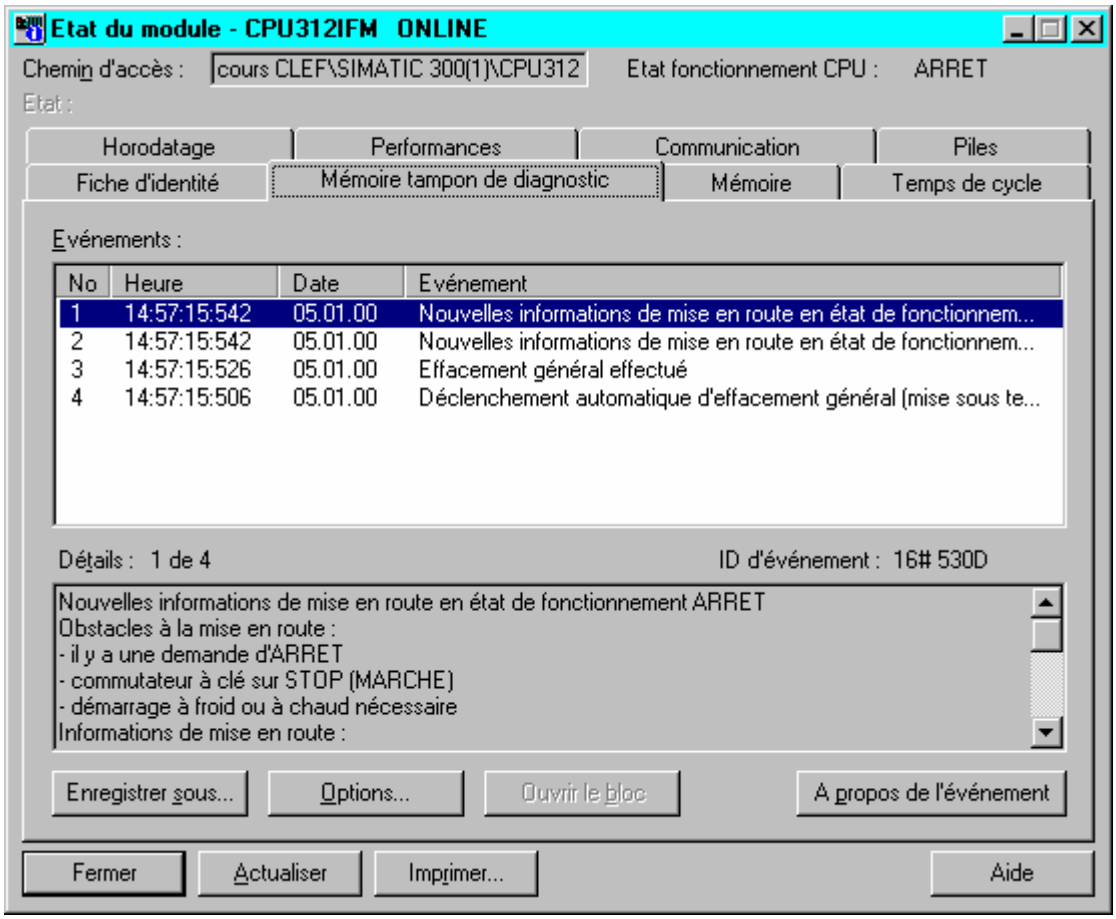
Le diagnostic de la CPU permet de déceler les erreurs suivantes :

* erreur système dans la CPU,
* défaut d'une carte,
* Erreur de programme dans la CPU.

On distingue encore les erreurs suivantes :

- les erreurs synchrones qui sont directement liées à une instruction précise du programme
- Les erreurs asynchrones qui ne sont pas liées à une instruction erronée du programme, par exemple alarme de diagnostic d'une carte.

Pour analyser l'erreur, il faut recourir à une console de programmation. Pour accéder au tampon de diagnostic, sélectionner Système Cible➔Etat du module➔Mémoire tampon de diagnostic dans l'éditeur de programme ou dans le SIMATIC Manager.



Dans le S7-300, des blocs d'organisation permettent de programmer la réaction de la CPU face à des erreurs. Ces OB peuvent être programmées de manière à indiquer à la CPU ce qu'elle doit faire en cas de détection d'une erreur. Si l'OB correspondant n'a pas été programmé, la CPU passe en mode STOP.

Laboratoire

Projet : Chaîne d'emballage

- **Description technologique.**

La chaîne de montage se compose de trois postes d'assemblage et d'un poste d'emballage final.
Un tapis roulant transporte les pièces d'un des trois postes vers le poste d'emballage final. Un faisceau lumineux détecte l'arrivée de la pièce au poste final.
Sur chacun des trois postes d'assemblage, un détecteur de proximité permet de détecter qu'une pièce se trouve sur le tapis (DET1 DET2 DET3). Les 4 postes sont munis d'une lampe de signalisation (L1 L2 L3 L4) et d'un bouton poussoir (BP1 BP2 BP3 BP4).

- **Fonctionnement**

L'installation peut être commandée suivant deux modes de fonctionnement : *mode manuel et mode automatique*.
En mode automatique, les lampes L1, L2 et L3 indiquent qu'une pièce peut être déposée sur le tapis. Lorsqu'une pièce est détectée à un poste, les lampes des autres postes s'éteignent, un klaxon s'enclenche pendant 3 secondes signalant que le tapis va fonctionner. Le tapis ne peut être mis en marche que lorsqu'un seul des trois postes présente une pièce. Le tapis transporte la pièce jusqu'au poste final. La lampe du poste clignote lorsque le tapis est en mouvement. La détection de la pièce par le faisceau lumineux (FL) arrête le tapis et permet le comptage des pièces. Cette procédure se répète jusqu'à ce que 5 pièces soient arrivées au montage final. La lampe L4 de l'emballage est allumée et le tapis est bloqué. Les lampes des trois postes sont éteintes. Un nouveau cycle peut recommencer par activation du bouton poussoir BP4.
En mode manuel, le tapis peut être déplacé dans les deux directions à l'aide des boutons poussoirs BPTPAV et BPTPAR du pupitre de commande.
Les lampes des postes doivent clignoter dès que le tapis tourne, quel que soit le mode et le sens.

- **Commande EN/HORS Fonctionnement.**

La commande est enclenchée par le bouton poussoir BPON et déclenchée par le bouton poussoir BPOFF (contact d'ouverture). A l'état enclenché, la lampe SERV est allumée. Les lampes et le moteur du tapis ne doivent pouvoir être enclenchés qu'à l'état SERV de la commande.
Le mode MANUEL peut s'enclencher si le sélecteur de mode MODE délivre un signal 0 et que le bouton poussoir de «validation de mode» SMODE est actionné. La lampe LMANU signale ce mode de fonctionnement
Le mode AUTO est enclenché si MODE est à 1 et si une impulsion est donnée sur SMODE. La signalisation de ce mode est fournie par LAUTO.
L'annulation de ces deux modes est réalisée par le déclenchement de l'installation. Une modification de l'état du sélecteur MODE annule le mode sélectionné.

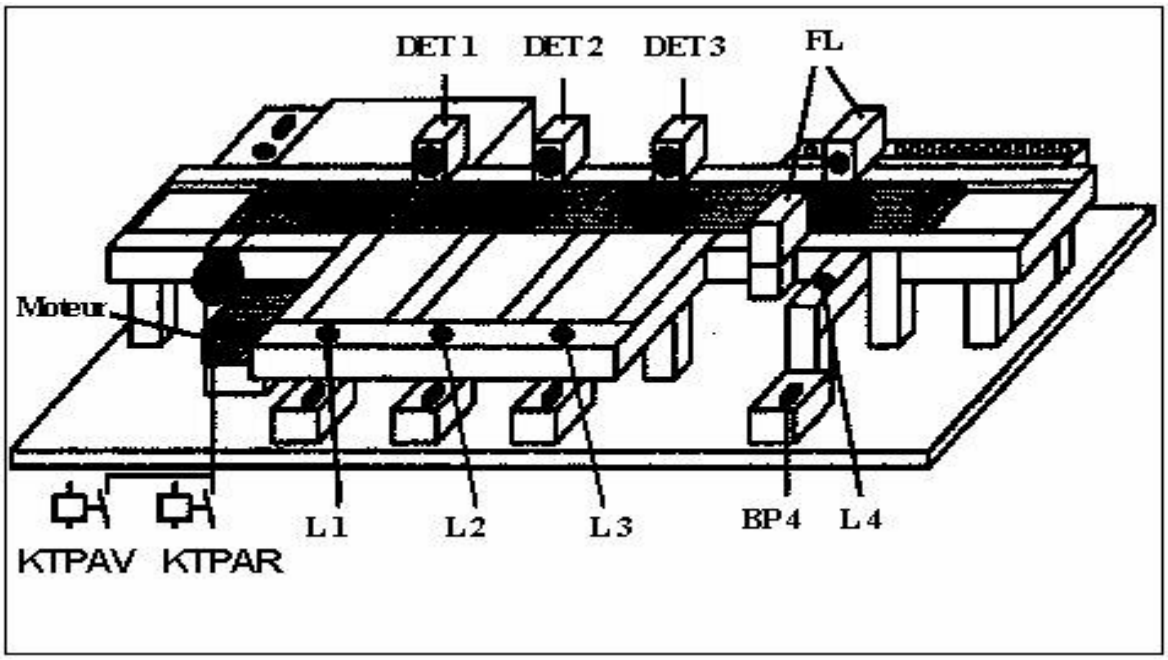
- **Liste d'assignation**

Entrées :

Symbole	Rôle
BPON	Enclenchement de l'installation (NO)
BPOFF	Déclenchement de l'installation (NF)
MODE	Sélecteur de mode de fonctionnement
SMODE	Prise en compte du mode
BPTPAV	Tapis en fonctionnement manuel - avant
PTPAR	Tapis en fonctionnement manuel - arrière
DET1	Détecteur de proximité poste 1
BP4	Bouton poussoir poste final
FL	Faisceau lumineux

Sorties

Symbole	Rôle
SERV	Installation en SERVice
KLAXON	Klaxon
LMANU	Fonctionnement MANUEL
LAUTO	Fonctionnement AUTOMATIQUE
L1	Signalisation poste 1
L4	Signalisation emballage
KTPAV	Commande moteur en avant
KTPAR	Commande moteur en arrière
AFFICH	Nombre de pièces produites
CLIGNOT	Mémento de clignotement



FC1 : Modes de fonctionnement

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
I 0.0	***BPON***	***BP de mise en service de l'installation.***	***NO***	***0 :Inactivé => pas d'effet***
				1 :Activé => mise en service
I 0.1	***BPOFF***	***BP pour arrêt de l'installation***	***NF***	***0 : Activé => arrêt***
				1 : Inactivé => pas d'effet
I 0.3	***Valmode***	***Validation du mode***	***NO***	***0 :Inactivé => pas d'effet***
				1 :Activé => validation
I 0.2	***Mode***	***Sélecteur de mode de fonctionnement***	***Switch***	***0 : mode manu***
				1 : mode auto

Sorties :

<table>
<tr><td></td><td>Symbole</td><td>Rôle</td><td colspan="2">Signification logique</td></tr>
<tr><td rowspan="3">Q4.0</td><td rowspan="3">SERV</td><td rowspan="2">Témoin de l'installation en service</td><td colspan="2">0 : Installation à l'arrêt</td></tr>
<tr><td colspan="2">1 : Installation en service</td></tr>
<tr><td colspan="3">Enclenchement : BPON = 1

Déclenchement : BPOFF = 0</td></tr>
<tr><td rowspan="3">Q4.2</td><td rowspan="3">Lmanu</td><td rowspan="2">Témoin du fonctionnement manuel</td><td>0 :Pas en manu</td></tr>
<tr><td>1 : mode manu</td></tr>
<tr><td colspan="2">Enclenchement : mode = 0 et valmode =1

Déclenchement : pas en service ou mode =1</td></tr>
<tr><td rowspan="3">Q4.3</td><td rowspan="3">Lauto</td><td rowspan="2">Témoin du fonctionnement automatique</td><td>0 : pas en auto</td></tr>
<tr><td>1 : mode auto</td></tr>
<tr><td colspan="2">Enclenchement : mode = 1 et valmode =1

Déclenchement : pas en service ou mode =0</td></tr>
</table>

FC2 : Commande du tapis

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
I 0.7	***DET 1***	***Détecter la pièce au poste 1***	***NO***	***0 : Pas de pièce***
				1 : Pièce présente
I 1.4	***FL***	***Détecter la pièce au poste***	***NF***	***0 : Pièce devant faisceau***
				1 : pas de pièce
I 0.4	***BPTPAV***	***BP pour avance manuelle du tapis en marche avant***	***NO***	***0 : pas d'effet***
				1 : commande avant manu
I 0.5	***BPTPAR***	***BP pour avance manuelle du tapis en marche arrière***	***NO***	***0 : pas d'effet***
				1 : commande arrière manu

Sorties :

Adresse	Symbole	Rôle		Signification logique
Q 4.6	***KTPAV***	***Commande du relais marche avant tapis***		0 :
				1 :
		Mode auto	Enclenchement : Déclenchement :	
		Mode manu		
Q 4.7	***KTPAR***			0 :
				1 :

FC3 : Comptage et lampe poste 4

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
I 1.4	***FL***			0 :
				1 :
I 1.6	***BP4***			0 :
				1 :

Sorties :

Adresse	Symbole	Rôle	Signification logique
Q5.4	***L4***		0 :
			1 :
		Enclenchement : Déclenchement :	
QW124	***Affichage***		
C 4	***N-pieces***		0 :
			1 :
		(Dé)Comptage : Remise à zéro ou initialisation :	

FC4 : Lampe L1

Sorties

Symbole	Adresse	Rôle	Signification logique
L1	Q 5.1		0 :
			1 :

SIMATIC lG3CG\Chaine d'emballage3\...\OB1 - <offline> 15/09/2009 15:49:17

OB1 - <offline>

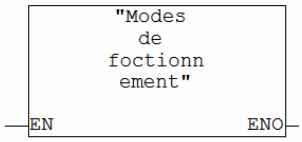
""

Nom : **Famille :**
Auteur : **Version :** 0.1
Version de bloc : 2
Horodatage Code : 15/09/2009 15:48:54
Interface : 15/02/1996 16:51:12
Longueur (bloc/code /données locales) : 00184 00066 00022

Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

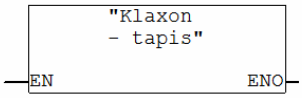
Bloc : OB1 "Main Program Sweep (Cycle)"

Réseau : 1



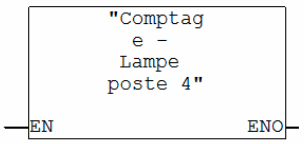
Informations mnémonique
FC1 Modes de foctionnement

Réseau : 2



Informations mnémonique
FC2 Klaxon - tapis

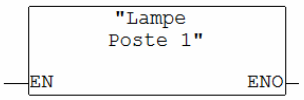
Réseau : 3



Informations mnémonique

FC3	Comptage - Lampe poste 4	

Réseau : 4



Informations mnémonique

FC4	Lampe Poste 1	Commandes du poste 1

SIMATIC lG3CG\Chaine d'emballage3\...\FC1 - <offline> 15/09/2009 15:52:56

FC1 - <offline>

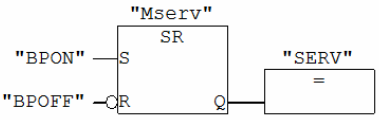
"Modes de foctionnement"

Nom : **Famille :**
Auteur : **Version :** 0.1
Version de bloc : 2
Horodatage Code : 08/03/2005 11:35:14
Interface : 08/03/2005 10:33:34
Longueur (bloc/code /données locales) : 00150 00054 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1 Modes de fonctionnement

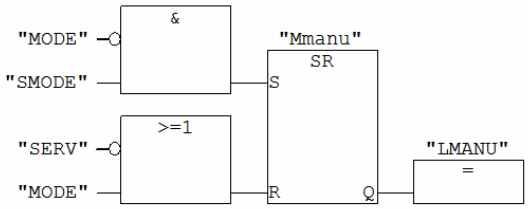
Réseau : 1 Mise en service



Informations mnémonique

M20.0	Mserv
I0.0	BPON
I0.1	BPOFF
Q4.0	SERV

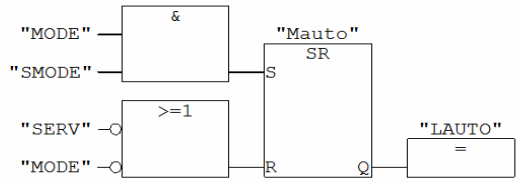
Réseau : 2 Enclenchement Mode manuel



Informations mnémonique

I0.2	MODE
I0.3	SMODE
Q4.0	SERV
M20.2	Mmanu
Q4.2	LMANU

Réseau : 3 Enclenchement mode automatique



Informations mnémonique

I0.2	MODE
I0.3	SMODE
Q4.0	SERV
M20.3	Mauto
Q4.3	LAUTO

SIMATIC lG3CG\Chaine d'emballage3\...\FC2 - <offline> 15/09/2009 15:54:31

FC2 - <offline>

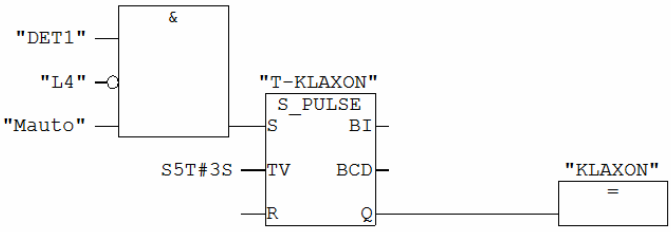
"Klaxon - tapis"

Nom : **Famille :**
Auteur : **Version :** 0.0
Version de bloc : 2
Horodatage Code : 30/10/2006 12:39:45
Interface : 23/09/2002 11:43:08
Longueur (bloc/code /données locales) : 00170 00074 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2 commande tapis

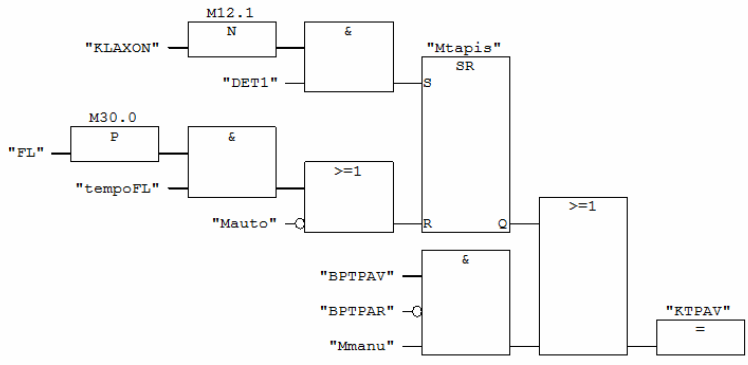
Réseau : 1 Klaxon



Informations mnémonique

I0.7	DET1
Q5.4	L4
M20.3	Mauto
T2	T-KLAXON
Q4.1	KLAXON

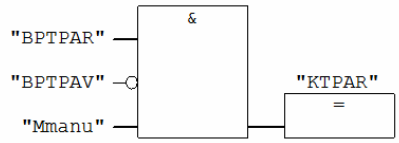
Réseau : 2 Moteur marche avant



SIMATIC lG3CG\Chaine d'emballage3\...\FC2 - <offline> 15/09/2009 15:54:31

Informations	**mnémonique**
Q4.1	KLAXON
I0.7	DET1
I1.4	FL
M30.1	tempoFL
M20.3	Mauto
M20.6	Mtapis
I0.4	BPTPAV
I0.5	BPTPAR
M20.2	Mmanu
Q4.6	KTPAV

Réseau : 3 Moteur marche arrière



Informations	**mnémonique**
I0.5	BPTPAR
I0.4	BPTPAV
M20.2	Mmanu
Q4.7	KTPAR

SIMATIC lG3CG\Chaine d'emballage3\...\FC3 - <offline> 15/09/2009 15:55:20

FC3 - <offline>

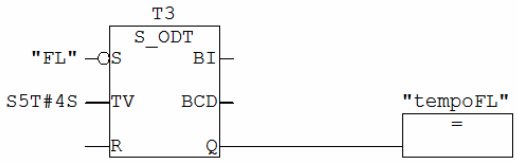
"Comptage - Lampe poste 4"

Nom : **Famille :**
Auteur : **Version :** 0.0
Version de bloc : 2
Horodatage Code : 10/02/2009 15:26:16
Interface : 23/09/2002 13:18:28
Longueur (bloc/code /données locales) : 00188 00088 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC3 Poste 4

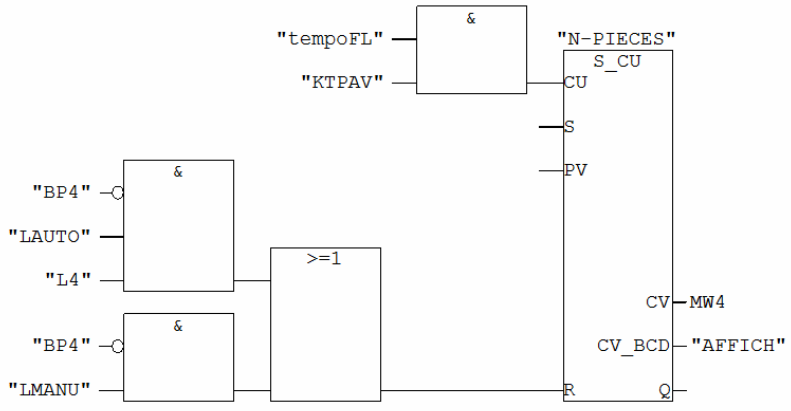
Réseau : 1



Informations mnémonique

I1.4	FL
M30.1	tempoFL

Réseau : 2 comptage pièces

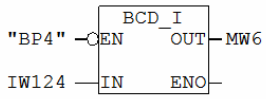


Informations mnémonique

M30.1	tempoFL

Q4.6	KTPAV
I1.1	BP4
Q4.3	LAUTO
Q5.4	L4
Q4.2	LMANU
C4	N-PIECES
QW124	AFFICH

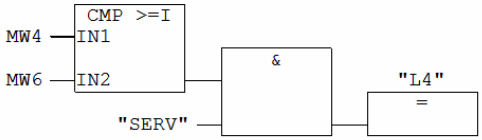
Réseau : 3



Informations mnémonique

I1.1	BP4

Réseau : 4 Lampe L4



Informations mnémonique

Q4.0	SERV
Q5.4	L4

SIMATIC lG3CG\Chaine d'emballage3\...\FC4 - <offline> 15/09/2009 15:56:24

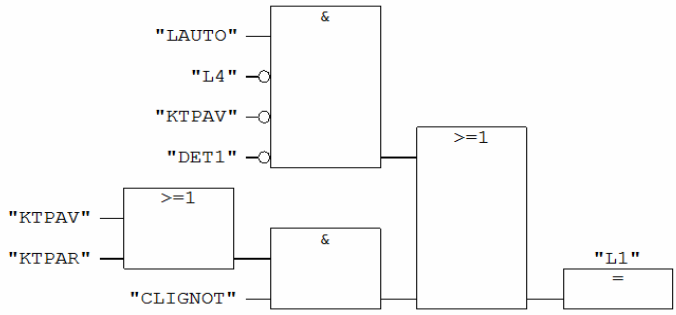
FC4 - <offline>

"Lampe Poste 1" Commandes du poste 1
Nom : **Famille :**
Auteur : **Version :** 0.0
Version de bloc : 2
Horodatage Code : 10/02/2009 15:28:00
Interface : 23/09/2002 13:44:55
Longueur (bloc/code /données locales) : 00116 00024 00000

Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC4 Poste 1

Réseau : 1 lampe L1



Informations mnémonique

Q4.3	LAUTO
Q5.4	L4
Q4.6	KTPAV
I0.7	DET1
Q4.7	KTPAR
M255.3	CLIGNOT
Q5.1	L1

Labo 1 : *Le Step 7 , les opérations combinatoires binaires*

1.1. Présentation et objectifs du labo

1.2. Structure d'un SAP

1.3. Présentation des API

1.4. Présentation du logiciel STEP 7

1.5. Configuration matérielle

=> Configuration matérielle

1.6. Structure d'un programme, types de blocs, traitement cyclique, MIE-MIS

1.7. Présentation du projet chaîne d'emballage; structure du programme

1.8. Programmation : affectation, fonction logique ET, fonction logique OU

Application chaîne d'emballage
FC1 : modes de fonctionnement

- Mise en service simplifiée par switch ON / OFF  I 1.2 Q 4.0
- Mode auto/manu par switch  I 0.2  Q 4.2  Q 4.3

1.9. Test : visualisation dynamique

1.10. Affichage dans les 3 modes de représentation

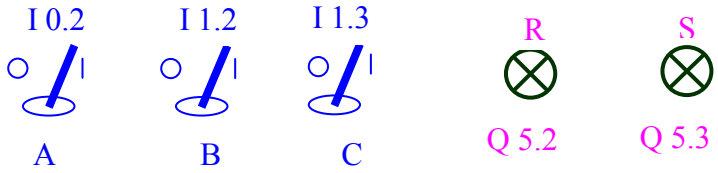
1.11. Introduction des mnémoniques

Exercice 1 : Interrupteurs et commandes de lampes

Trois interrupteurs A, B, C commandent l'allumage de 2 lampes R et S suivant les conditions :

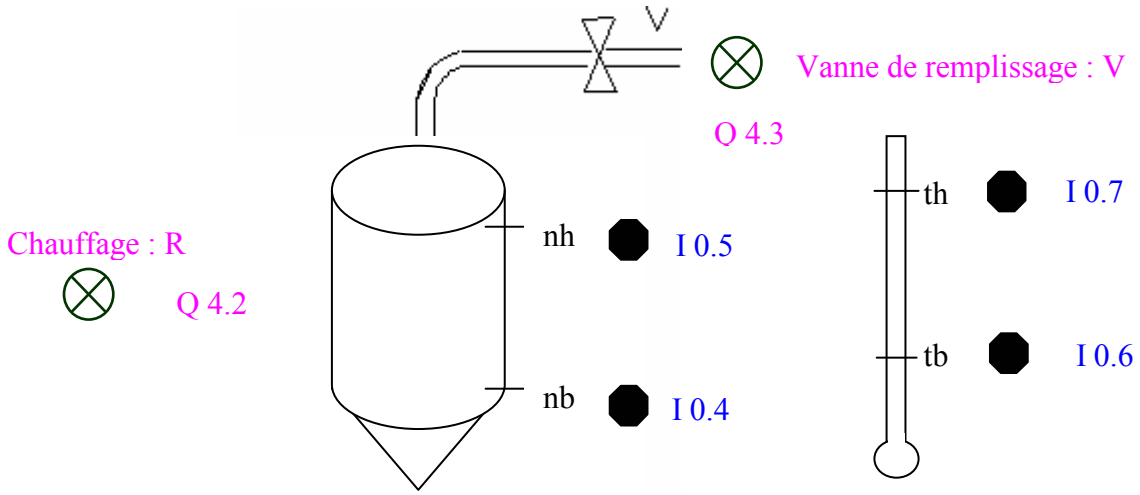
- Dès qu'au moins un interrupteur est activé, R doit s'allumer
- S ne s'allume que si au moins deux interrupteurs sont activés

Déterminez les expressions des fonctions R et S et faites exécuter le programme par l'automate.



Exercice 2 : Surveillance niveau et température de cuve.

Le niveau d'une cuve est contrôlé par 2 capteurs de niveau (nb, nh) et
2 capteurs de température (th, tb).
Une vanne permet le remplissage tant que le niveau haut n'est pas atteint.
Une résistance chauffante assure le chauffage jusqu'à la température maximale.
Une sécurité de fonctionnement interdit le chauffage si le niveau bas est atteint, de même le remplissage est arrêté si la température minimale est atteinte.
Les capteurs de niveau sont à l'état logique 1 lorsque l'eau est présente devant le capteur.
Les capteurs de température sont à l'état logique 1 si la température est supérieure

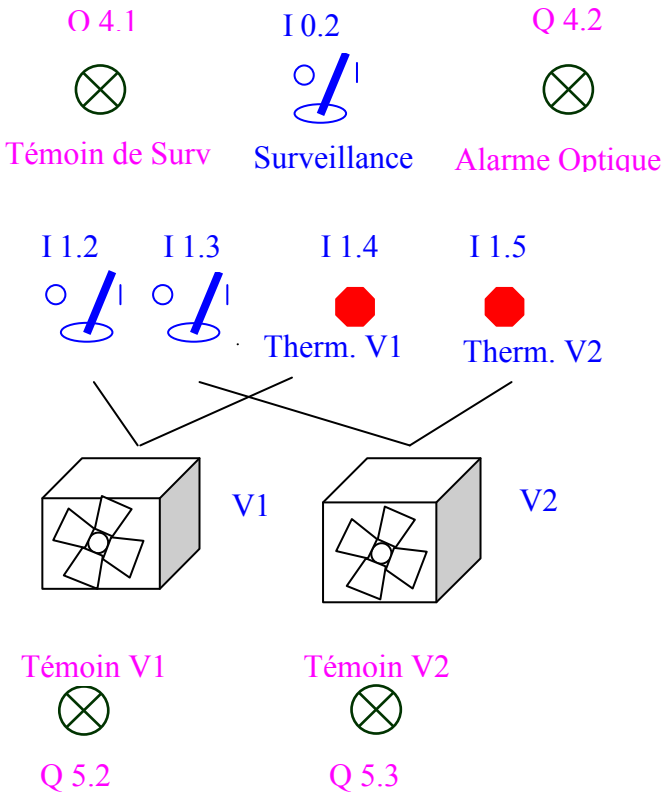


Pour sécuriser le système, on modifie le capteur nh en le câblant en NF. I 1.5

1.12. Contact NO-NF

Exercice 3 : Surveillance 2 ventillateurs (simplifié)

Une unité est refroidie par deux ventilateurs.
Les ventilateurs sont mis en service à l'aide d'interrupteurs.
Le contrôle de la ventilation est effectué grâce à deux thermiques.
Le fonctionnement correct (Venti enclenché et Therm. OK) de chaque unité de ventilation est visualisé par un témoin.
En cas d'arrêt d'un des ventilateurs, une alarme optique est nécessaire ; l'alarme s'éteint lorsque le ventilateur est remis en service (lorsque le défaut a disparu) ou que la surveillance n'est pas branchée.



Labo 2 : *La fonction mémoire et la détection de front*

2.1 Fonction auto-maintien ou fonction mémoire

Application Chaîne d'emballage
FC1: Modes de fonctionnement

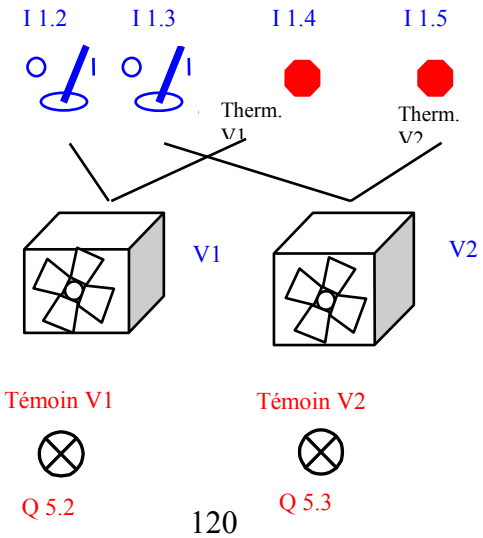
Mise en service : I 0.0 I 0.1 Q 4.0

Mode automatique : I 0.3 I 0.2 Q 4.3

Mode manu : I 0.3 I 0.2 Q 4.2

Exercice 1 : Surveillance des ventilateurs

Une unité (mise en service par le biais d'un BPON et d'un BPOFF) est refroidie par deux ventilateurs. Les ventilateurs sont mis en service à l'aide d'interrupteurs. Le contrôle de la ventilation est effectué grâce à deux thermiques. Le fonctionnement correct (Venti enclenché et Therm. OK) de chaque unité de ventilation est visualisé par un témoin.
Les fonctions d'enclenchement et d'alarme sont à réaliser. Les alarmes ne doivent être actives que lorsque la surveillance est enclenchée.
En cas de défaut d'un des ventilateurs, une alarme optique est nécessaire. L'alarme s'éteint lorsque le ventilateur est remis en service (lorsque le défaut a disparu). L'arrêt des deux ventilateurs donne lieu à une alarme acoustique aussi longtemps que l'unité est en service. Cette alarme sonore sera présente jusqu'à son acquittement par un bouton poussoir prévu à cet effet. L'acquit n'est opérationnel que lorsqu'au moins un des ventilateurs est en service

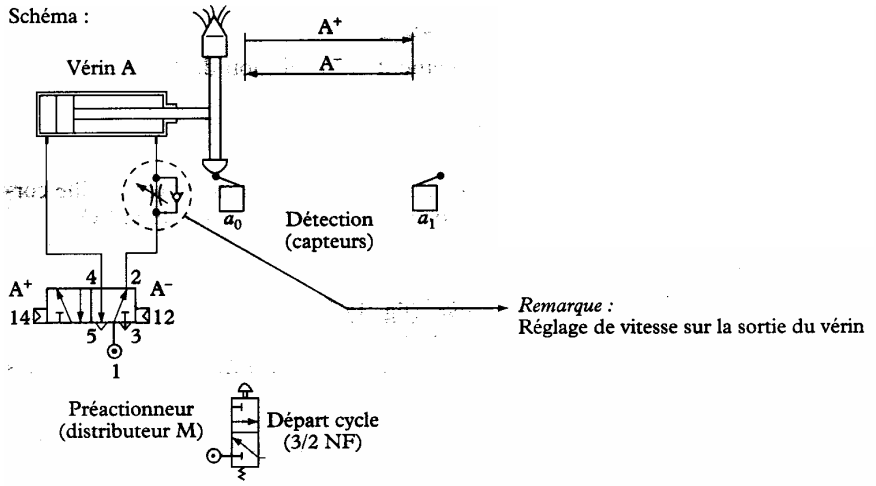


2.2 Fronts d'un signal

Démo : Déplacement d'un pistolet de peinture

Une installation est enclenchée par le bouton poussoir BPON et déclenchée par le bouton poussoir BPOFF (contact d'ouverture). A l'état enclenché, la lampe SERV est allumée et le pistolet de peinture est activé.

Ce pistolet de peinture est fixé à un vérin double effet. Le vérin peut se déplacer entre deux contacts de fin de course a0 et a1 grâce à un distributeur qui permet la commande de la sortie du vérin (A+) et sa rentrée (A-). Lorsque l'opérateur ferme le contact du BP départ cycle (dcy) et à condition que le vérin soit rentré, il se déplace de a0 vers a1. Arrivé en a1, il retourne immédiatement en a0 où il reste arrêté. Lorsque dcy est à nouveau enfoncé, le cycle recommence.



Que se passe-t-il si le contact dcy est maintenu enfoncé lorsque le chariot revient à sa position de départ ?

Comment assurer que le cycle ne sera exécuté qu'une seule fois lorsque le contact dcy est enfoncé ?

Application Chaîne d'emballage

FC2 : Commande du tapis

En mode automatique, lorsqu'une pièce est détectée au poste 1 (DET 1), le tapis démarre et transporte la pièce jusqu'au poste final. Le tapis s'arrête lorsque la pièce a complètement dépassé le faisceau lumineux (FL).
En mode manuel, le tapis peut être déplacé dans les deux directions à l'aide des boutons poussoirs BPTPAV et BPTPAR du pupitre de commande.

Exercice 2 : Allumage des 3 lampes

On demande de réaliser un système de commande qui permet d'allumer successivement 3 lampes par action sur un même BP NO. L'extinction des 3 lampes se fait avec un bouton poussoir NF.

Exercice 3 : Projecteur (télérupteur)

Sur le projecteur de data, un même BP doit assurer la mise en service et l'arrêt d'une installation

Tableau : Surveillance de deux ventilateurs

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :

Sorties :

Adresse	Symbole	Rôle	Signification logique
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :

Labo 3 :

Format des nombres, compteurs, comparateurs et temporisations

3.1 Format des nombres

3.2 Fonction de comptage

3.3 Les opérations de comparaisons

Application Chaîne d'emballage
FC3 : Comptage et lampe poste 4

La détection de la pièce par le faisceau lumineux (FL) permet le comptage des pièces. Le nombre de pièces présentes à l'emballage doit être visible sur les afficheurs.

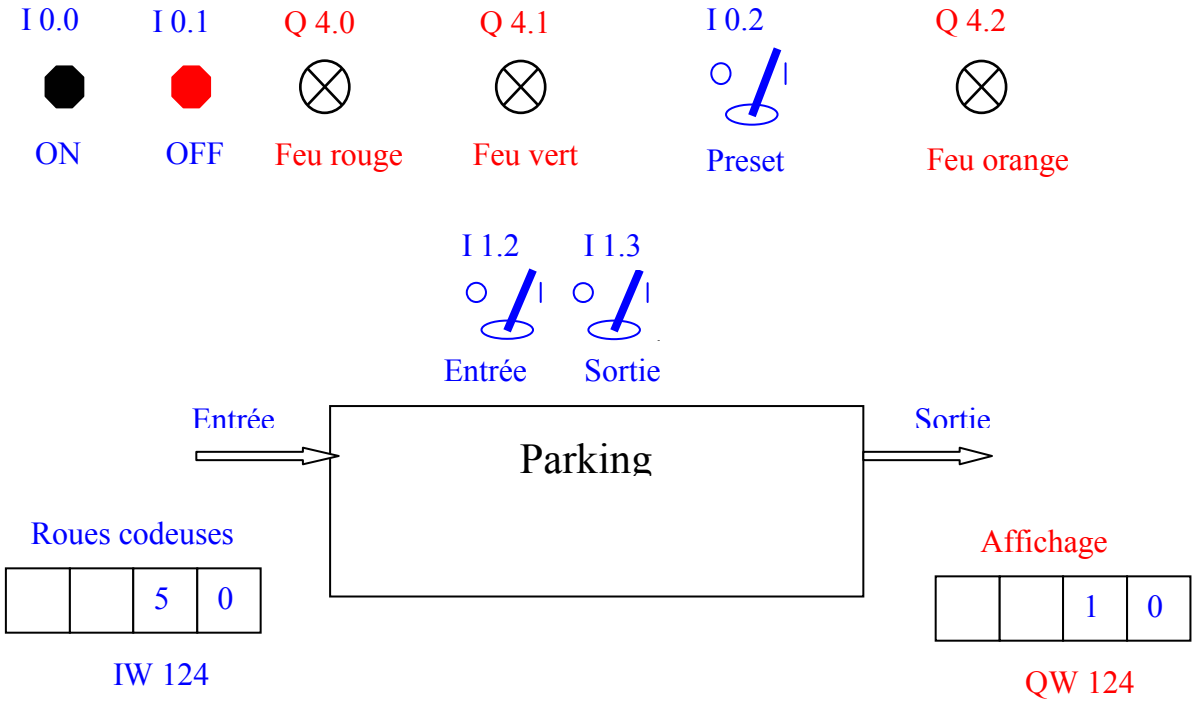
Cette procédure se répète jusqu'à ce que 5 pièces soient arrivées au montage final. La lampe L4 de l'emballage est allumée et le tapis est bloqué. Un nouveau cycle peut recommencer par activation du bouton poussoir BP4.

Amélioration *:* L'opérateur doit pouvoir choisir le nombre de pièces à emballer (>10) en le sélectionnant sur les roues codeuses (IW124)

Exercice 1 :Gestion d'un parking

Un compteur surveille le nombre de places libres d'un parking ; ce nombre de places sera affiché.
Ce nombre de places disponibles au départ peut être reconfiguré par l'intermédiaire de roues codeuses et d'un bouton preset.
Lorsque le parking est fermé (pas en service) le feu rouge est allumé.
Si le parking est ouvert, les couleurs des feux donnent les indications suivantes :

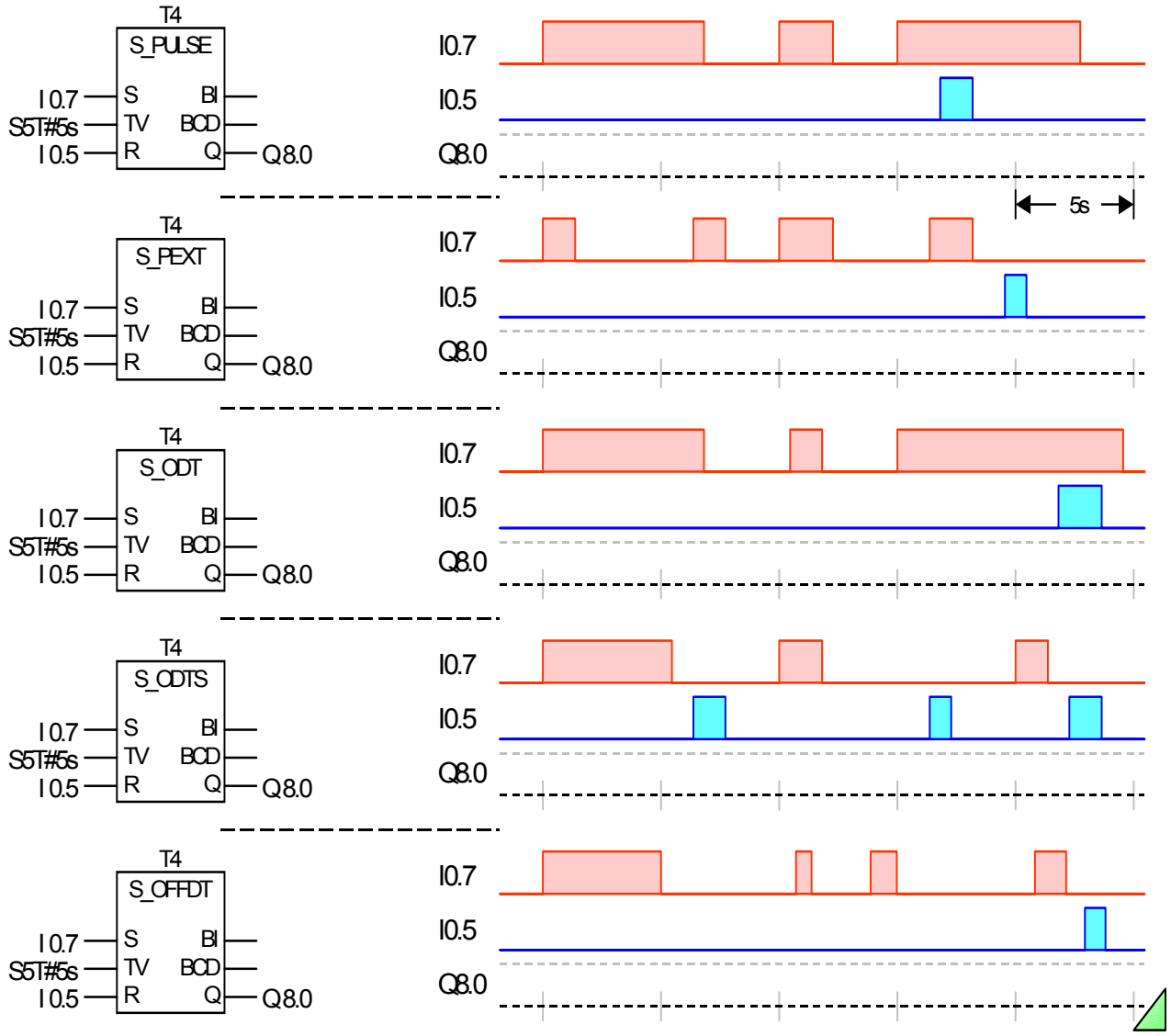
- Tant qu'il reste de la place, le feu vert est allumé.
- S'il reste moins de 5 places libres, le feu orange s'allume (le vert aussi).
- S'il n'y a plus de places libres le feu vert est éteint et le feu rouge allumé.


I 0.0
I 0.1
Q 4.0
Q 4.1
I 0.2
Q 4.2
ON
OFF
Feu rouge
Feu vert
Preset
Feu orange
I 1.2
I 1.3
Entrée
Sortie
Entrée
Parking
Sortie
Roues codeuses
5
0
IW 124
Affichage
1
0
QW 124

3.4 Fonction temporisation

Exercices

Complétez les chronogrammes suivants en traçant l'allure des signaux de sortie Q8.0 des différentes tempos et associez aux exemples



Exemples

Presse : l'opérateur enclenche l'opération de pliage est activant les deux poignées de la presse. Cette opération durera 10s à moins qu'il ne lâche ne fût-ce qu'une poignée.

Lorsque le BP d'un étage est enclenché, une minuterie permet que la lampe d'une cage d'escalier d'un immeuble soit allumée pendant 30s.

Une bande transporteuse est munie d'un faisceau lumineux qui détecte la présence de palettes qui doivent être comptées. Afin d'éviter toute incrémentation intempestive du compteur due à des parasites (déchet, mouche, stagiaire qui passe là où il ne doit pas …), on va exploiter le fait qu'une palette à compter interrompt le faisceau de manière continue pendant une durée d'au moins 4s.

Application Chaîne d'emballage
FC 4 : Lampe poste 1 FC2 : klaxon

En mode automatique, **la lampe L1 indique qu'une pièce peut être déposée sur le tapis.** Lorsqu'une pièce est détectée au poste 1(DET 1), **un klaxon s'enclenche pendant 3 secondes signalant que le tapis va fonctionner** pour transporter la pièce jusqu'au poste final. **La lampe du poste clignote lorsque le tapis est en mouvement.** Le tapis s'arrête lorsque la pièce a complètement dépassé le faisceau lumineux (FL).

Application Chaîne d'emballage
FC 3 : Comptage des pièces

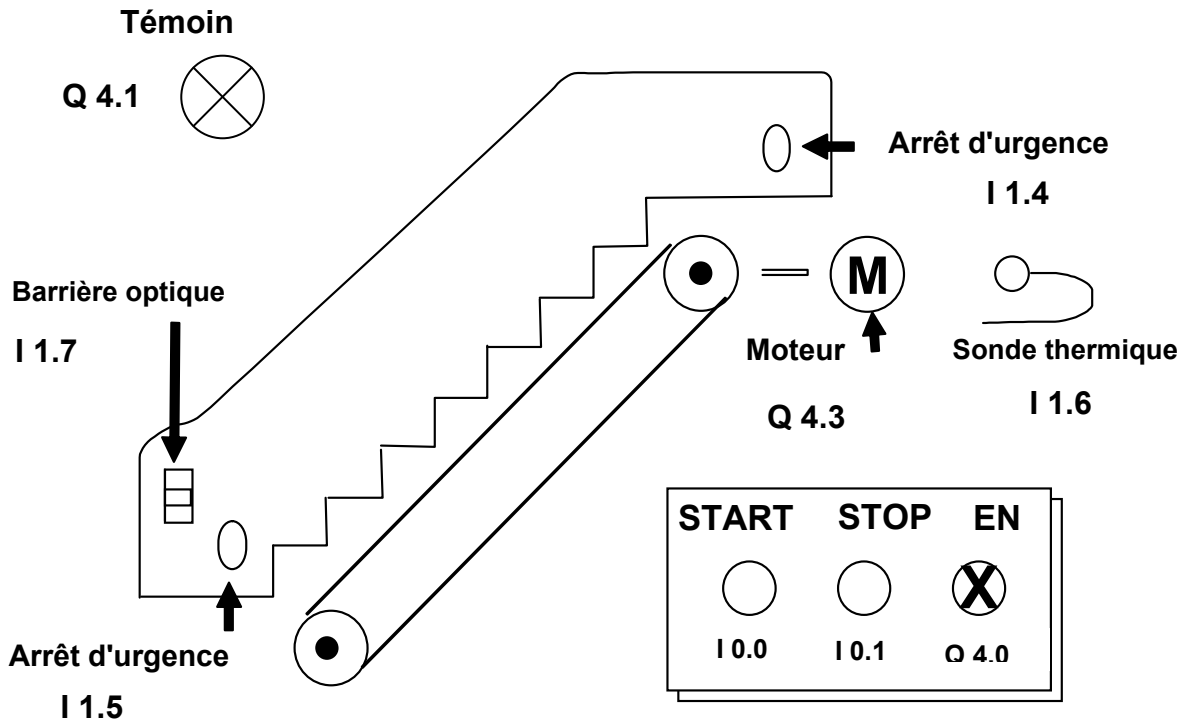
Le programme de comptage des pièces doit éviter que des coupures parasites du faisceau (mouche, micro-coupure, geste intempestifs, ...) ne soient prises en compte comme des pièces. On peut estimer que le temps de passage d'une pièce dans le faisceau est de 4 s.

Exercice 2 : Escalator

Un escalator est mis en service par bouton poussoir "marche". Le démarrage de cet escalator ne se produira que lorsqu'une cellule photo-électrique est interrompue. Après la libération de cette cellule, l'escalator doit encore fonctionner pendant 40 secondes.
L'arrêt est provoqué par :

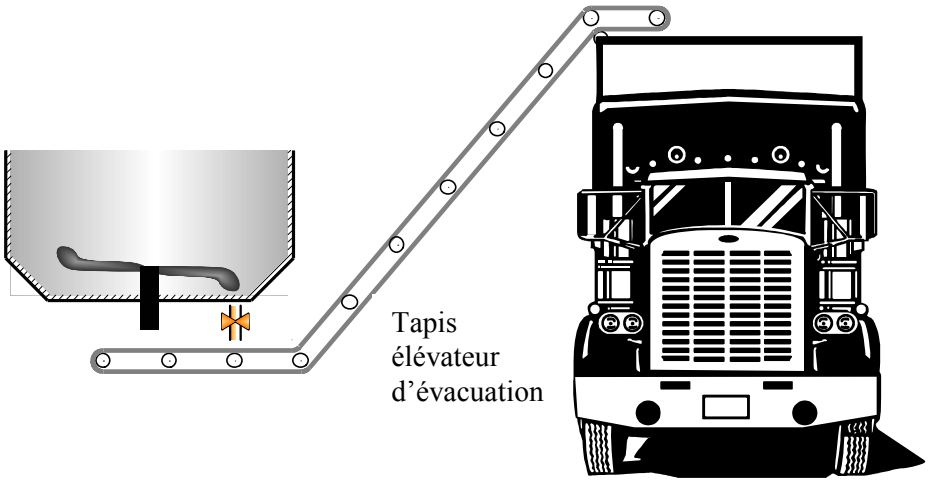
- bouton poussoir " arrêt"
- sonde thermique au niveau du moteur
- arrêt d'urgence (x 2)

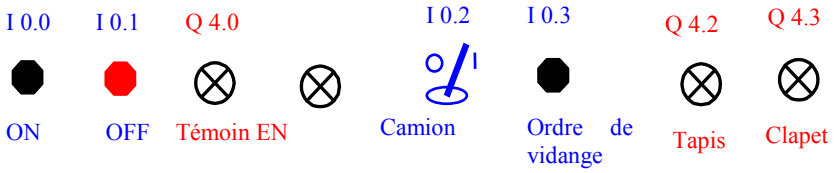
La mise en service est signalée par un voyant lumineux "en service". Un témoin lumineux clignotant indique que le moteur est en fonctionnement.



Exercice 3 : Vidange silo

Un opérateur peut vidanger un silo de sable dont le contenu se déverse sur une bande transporteuse pour être amené à un camion. Si le camion est bien présent et sur l'ordre de vidanger, le tapis se met en marche. 2sec plus tard (afin de laisser le temps au tapis d'atteindre sa vitesse de fonctionnement normal), le clapet de vidange du silo est ouvert. Dès que l'opérateur relâche l'ordre de vidange, le clapet se referme. Le tapis reste encore en fonctionnement pendant 10 sec afin d'évacuer tout le sable vidangé.





Exercice 4 : Gestion d'un pont

Un pont peut supporter 7 tonnes au maximum. Il faut donc surveiller le poids des véhicules se présentant aux deux extrémités A et B du pont où deux balances mesurent les poids respectifs a et b. On suppose que chaque véhicule a un poids inférieur à 7 tonnes :
Si un seul véhicule se présente, la barrière correspondante A ou B s'ouvre
Si a+b≤7 tonnes, les barrières A et B s'ouvrent
Si a+b>7tonnes, la barrière correspondant au véhicule le plus lourd s'ouvre
Si a=b, la barrière A s'ouvre en priorité

Tableau : gestion d'un parking

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :

Sorties :

Adresse	Symbole	Rôle	Signification logique
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :

Tableau : Escalator

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :

Sorties :

Adresse	Symbole	Rôle	Signification logique
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :

Tableau : vidange silo

Entrées :

Adresse	Symbole	Rôle	Type	Signification logique
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :
				0 :
				1 :

Sorties :

Adresse	Symbole	Rôle	Signification logique
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :
			0 :
			1 :