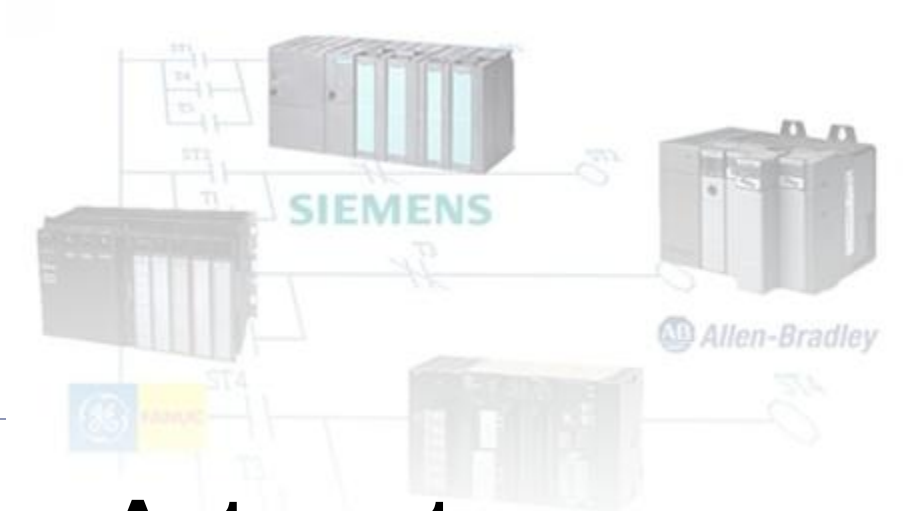
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE SUPERIEURE DES SCIENCES
APPLIQUEES DE TLEMCEN

وزارة التعليم العالي و البحث العلمي

المدرسة العليا العلوم التطبيقية تلمسان



Automates programmables industriels

Zoheir KARAOUZENE
ESSAT-TLEMCEN

zoheir_karaouzene@yahoo.fr

Automates programmables industriels

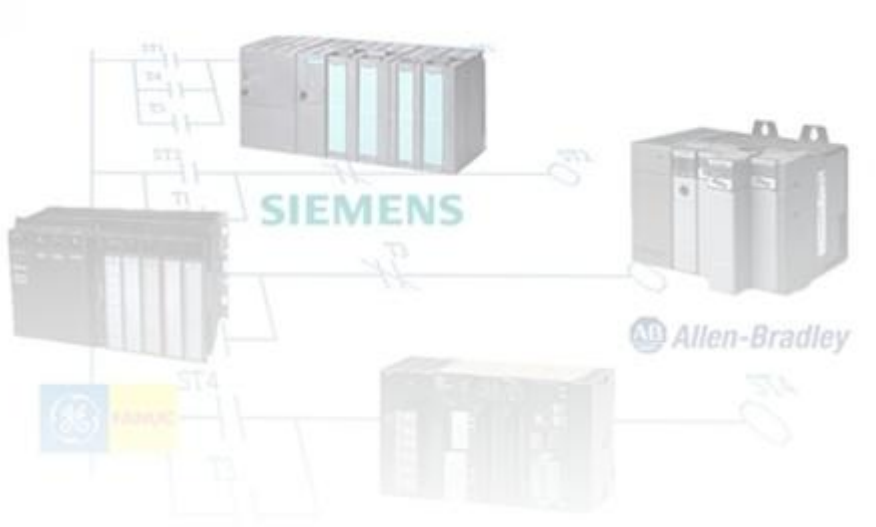
- **Pour 2eme année électrotechnique (UEF71)**
  - **Cours: 1h30**
  - **TD: 1h30**
  - **TP: 1h30**
  - **Crédit: 4**
  - **Coefficient: 4**
  - **CC: 40%**
  - **EF: 60%**

Automates programmables industriels

- **Introduction aux systèmes automatisés**
  - Système de production
  - Automatisation
  - Structure d'un système automatisé

- **I. Architecture d'un API**
  - Le processeur, la mémoire, le bus, l'alimentation, les modules d'entrée, les modules de sortie, les modules de comptage et la temporisation
  - les entrées logiques
  - Les entrée analogiques
  - Comptage et temporisation

- **II. Mise en œuvre d'un automate programmable industri**
  - Choix d'un automate programmable industriel
  - Raccordement de l'alimentation de l'unité de traitement
  - Raccordement des entrées logiques de l'unité de traitemer
  - Raccordement des entrées analogiques de l'unité de traiter
  - Raccordement des entrées spécialisées à l'unité de traitem
  - Raccordement des sorties logiques de l'unité de traitement
  - Raccordement des sorties analogiques de l'unité de traiter
  - Exemple de mise en œuvre de quelques automates



- **III. Programmation**
  - les langages littéraux
  - Les langages graphiques

I-Introduction aux systèmes automatisés

- Système de production
  - Un *système de production* regroupe l'ensemble des éléments matériels et immatériels qui sont nécessaires à la production de biens ou de services qui répondent à des objectifs de quantité, de prix, de qualité et de délai.

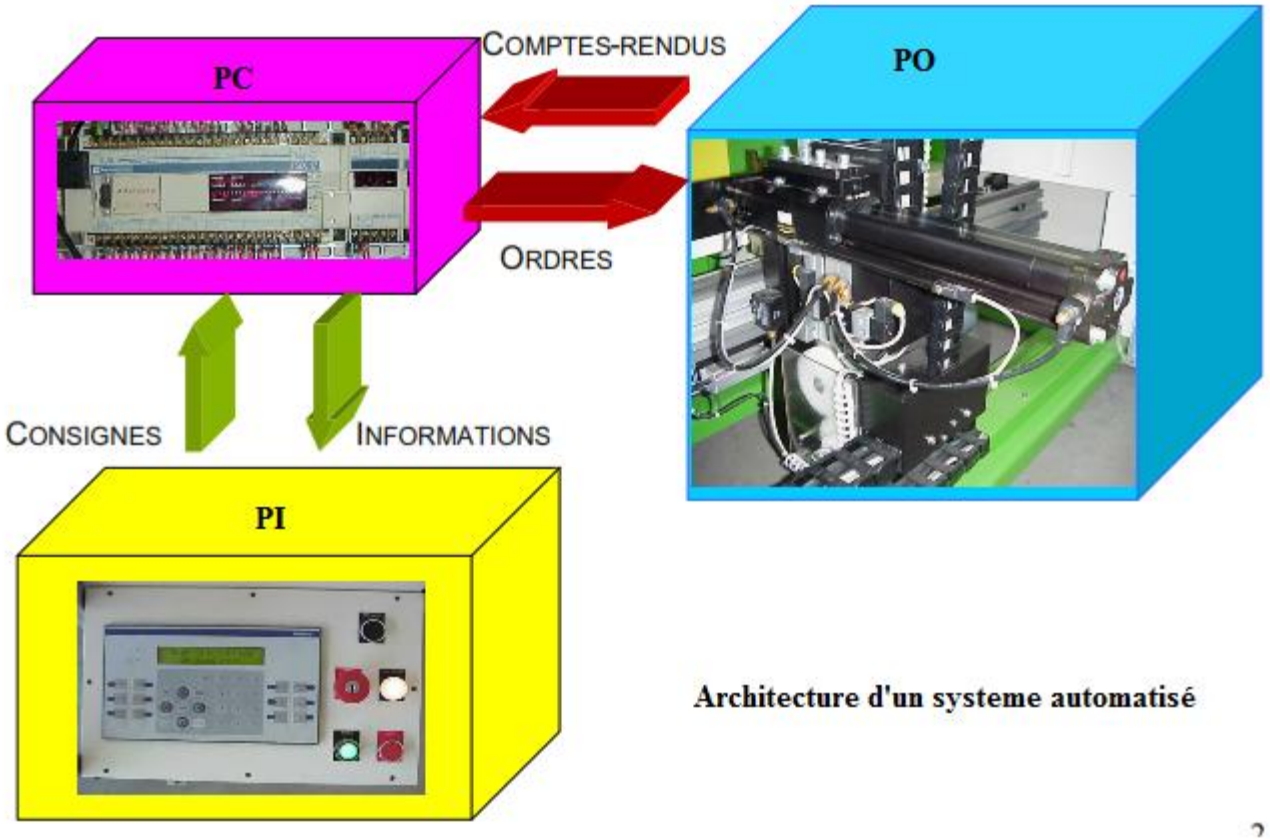
I-Introduction aux systèmes automatisés

- Automatisation
  - La réalisation de tâches par des machines est le cœur de l'automatisation industrielle. Elle est guidée par l'amélioration des opérations réalisées par des opérateurs, tout en augmentant la productivité et en améliorant la qualité.
  - L'objectif ultime est de remplacer l'homme pour les tâches pénibles et/ou répétitives (!!!!!)

I-Introduction aux systèmes automatisés

- Structure d'un système automatisé
  - Tout système automatisé COMPORTE :
    - Une PARTIE OPERATIVE (P.O.) procédant au traitement des matières d'œuvre afin d'élaborer la valeur ajoutée,
      - Machines;
        - Actionneurs
          - Vérins, moteurs …
      - Capteurs…
    - Une PARTIE COMMANDE (P.C.) coordonnant la succession des actions sur la PO
    - Partie interface (PI): entre les deux faces PO et PC, traduisant les ordres et les informations.
      - Lecture des valeurs système.
      - Paramétrage.
      - Statistiques …

I-Introduction aux systèmes automatisés



Architecture d'un systeme automatisé

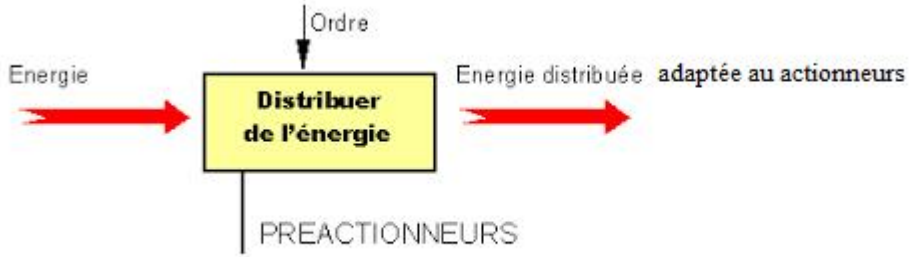
I-Introduction aux systèmes automatisés

La partie PO comporte :

- **Les capteurs**, qui communiquent à la partie commande des informations sur la position d'un mobile, une vitesse, la présence d'une pièce, une pression…
  - Les capteurs T.O.R. (tout ou rien), qui délivrent un signal de sortie logique, c'est à dire 0 ou 1.
    - Exemple : détecteur de fin de course.
  - Les capteurs numériques, ou « incrémentaux », qui associés à un compteur, délivrent des signaux de sortie numérique.
    - Exemple : capteur ou codeur incrémental utilisé pour la mesure des déplacements des chariots de machine à commande numérique
  - Les capteurs analogiques, ou proportionnels » qui permettent de prendre en compte la valeur réelle d'une grandeur physique.
    - Exemple : Sonde de température.

I-Introduction aux systèmes automatisés

- **Les actionneurs :**
  - Dans la littérature un actionneur est un objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système.
  - pouvant être un vérin hydraulique ou pneumatique, moteur électrique...
- **Les préactionneurs :**
  - Un *préactionneur* est un constituant dont le rôle est de distribuer, sur ordre de la P.C et à partir d'une énergie faible (entrée du préactionneur) l'énergie utile aux actionneurs.



I-Introduction aux systèmes automatisés

- **Les deux principaux types de préactionneur:**
  - 1°)Les préactionneurs pneumatiques : Les Distributeurs. Un distributeur est constitué d'une partie fixe et d'une partie mobile(le tiroir) :
    - La partie fixe est dotée d'orifices connectés à la source d'énergie (air comprimé), à l'actionneur et à l'échappement.
    - Le tiroir mobile, coulissant dans la partie fixe est doté de conduites permettant le passage de l'air entre les différents orifices et lapartie fixe.
  - 2°)Les préactionneurs électriques : Les Relais et Contacteurs.
    - Relais est le terme général qui désigne les préactionneurs électriques. Les contacteurs sont des relais conçus pour commuter des courantsélectriques forts.
    - Le contacteur est composé de contacts distribuant l'énergie à l'actionneur et d'un électro-aimant agissant sur ces contacts. En l'absence d'ordre de la P.C., les contacts sont au repos. Quand la partie commande (l'automate) envoie l'ordre de commande (signal de 24 V), le courant électrique crée un champ magnétique dans la bobine, qui pousse la barre de commande. Les contacts changent alors d'état. Dès la disparition de l'ordre, les contacts reprennent leur état de repos.

I-Introduction aux systèmes automatisés

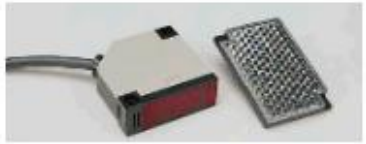
- Exemples de capteurs

Capteur de proximité à ultrasons

Capteur de niveau de liquide

Bouton poussoir

Capteur d'humidité



Cellule photoélectrique

Détecteur de gaz

Détecteur de choc

Détecteur de mouvement

Bouton d'arrêt d'urgence

I-Introduction aux systèmes automatisés

- Exemples d'actionneurs


Moteur pas à pas


Afficheur 7 segments


Voyants


Électro-vanne


Vérin rotatif


Ventilateur


Buzzer


Vérin

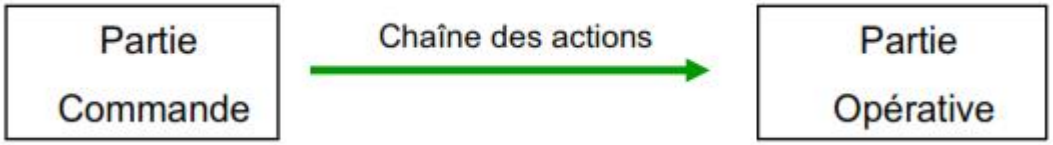

Résistance chauffante

I-Introduction aux systèmes automatisés

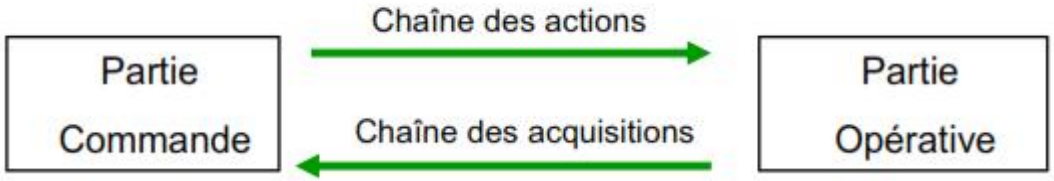
- Mode de commande

Un système automatisé peut utiliser deux modes de commande :

- Mode de commande directe ;



- Mode de commande avec compte-rendu d'exécution
  - *beaucoup plus fiable*



I-Introduction aux systèmes automatisés

- Outils de représentation/programmation (pour la partie commande):

  Il existe différents langages de programmation définis par la CEI 61131-3 :

  - IL (Instruction List), le langage List est très proche du langage assembleur
  - ST (Structured Text), Ce langage structuré ressemble aux langages de haut niveau utilisés pour les ordinateurs
  - LD (Ladder Diagram), le langage Ladder (échelle en anglais) ressemble aux schémas électriques
  - Boîtes fonctionnelles (FBD), le FBD se présente sous forme diagramme : suite de blocs, connectables entre eux, réalisant des opérations, simples ou très sophistiquées.
  - SFC Sequential function chart, dérivé du grafcet (Graphe Fonctionnel de Commande des Étapes et Transitions), le grafcet est dédié à la spécification, alors que SFC est plus appliqué à la programmation.

I-Introduction aux systèmes automatisés

- Selon sa complexité, la réalisation de la partie commande (PC) fait appel à diverses technologies dont les plus couramment utilisées sont :

- les relais électromécaniques
- les relais statiques électroniques
- les relais pneumatiques

} LOGIQUE CABLEE

- l'automate programmable
- les cartes électroniques à base d'un microcontrôleur

} LOGIQUE PRGRAMMEE

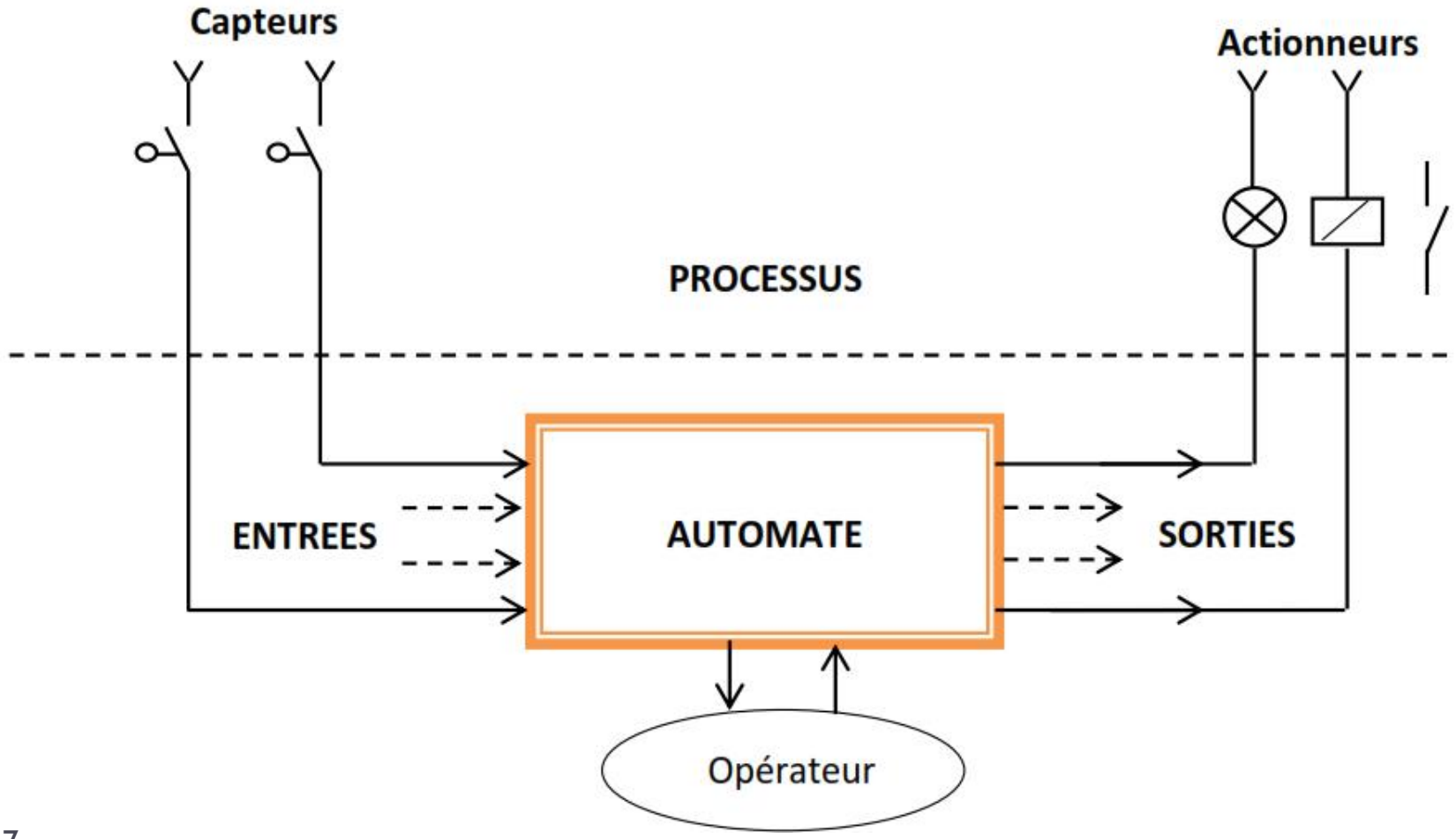
I-Introduction aux systèmes automatisés

- A partir d'une certaine, complexité, les relais électromécaniques et les relais statiques deviennent lourds à mettre en œuvre et le cout de l'automatisation est difficile à estimer.
- Les API autorisent la réalisation aisée d'automatismes comprenant de quelque dizaines jusqu'à plusieurs milliers d'entrées/sorties.

A la fin des années 60, Un fabricant américain de voitures décide de remplacer les systèmes de commande à base de logique cablée (relais électrique) par une logique programmée.

I-Introduction aux systèmes automatisés

L'automatisation???



II-Architecture d'un API

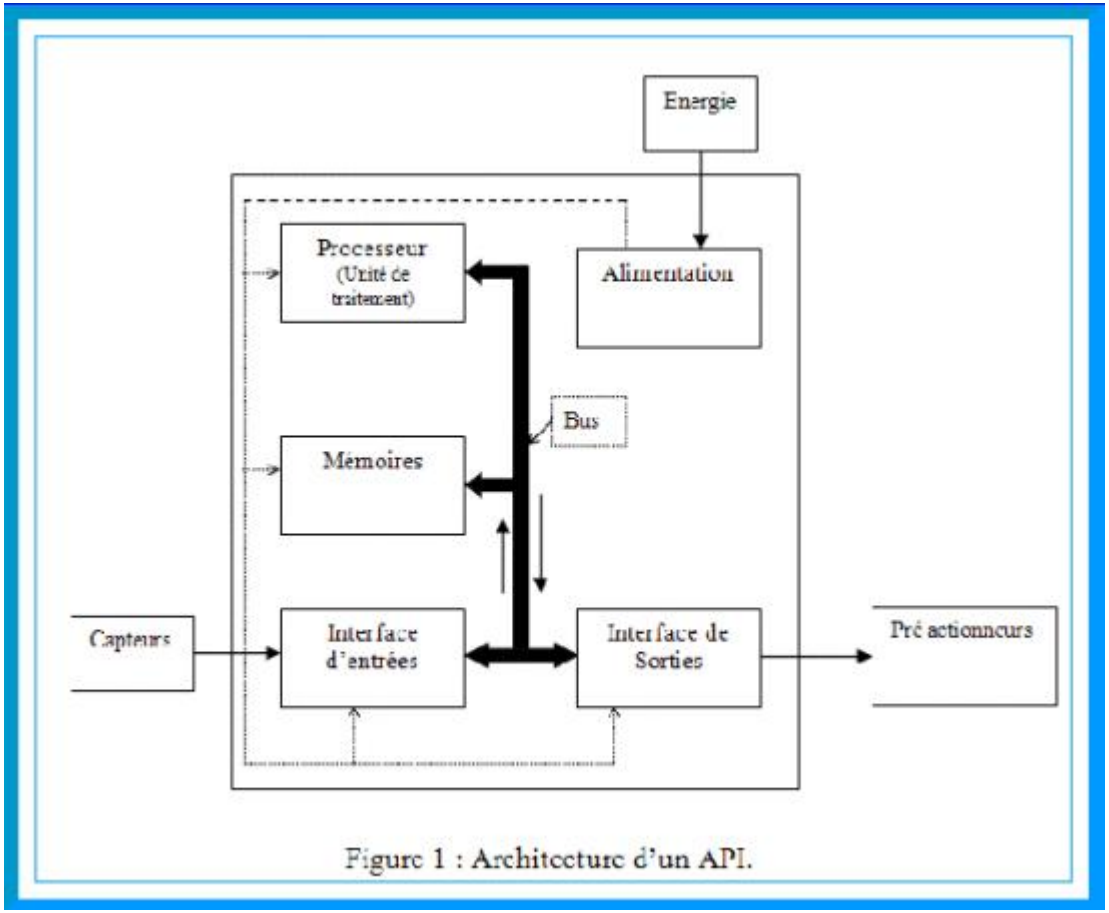
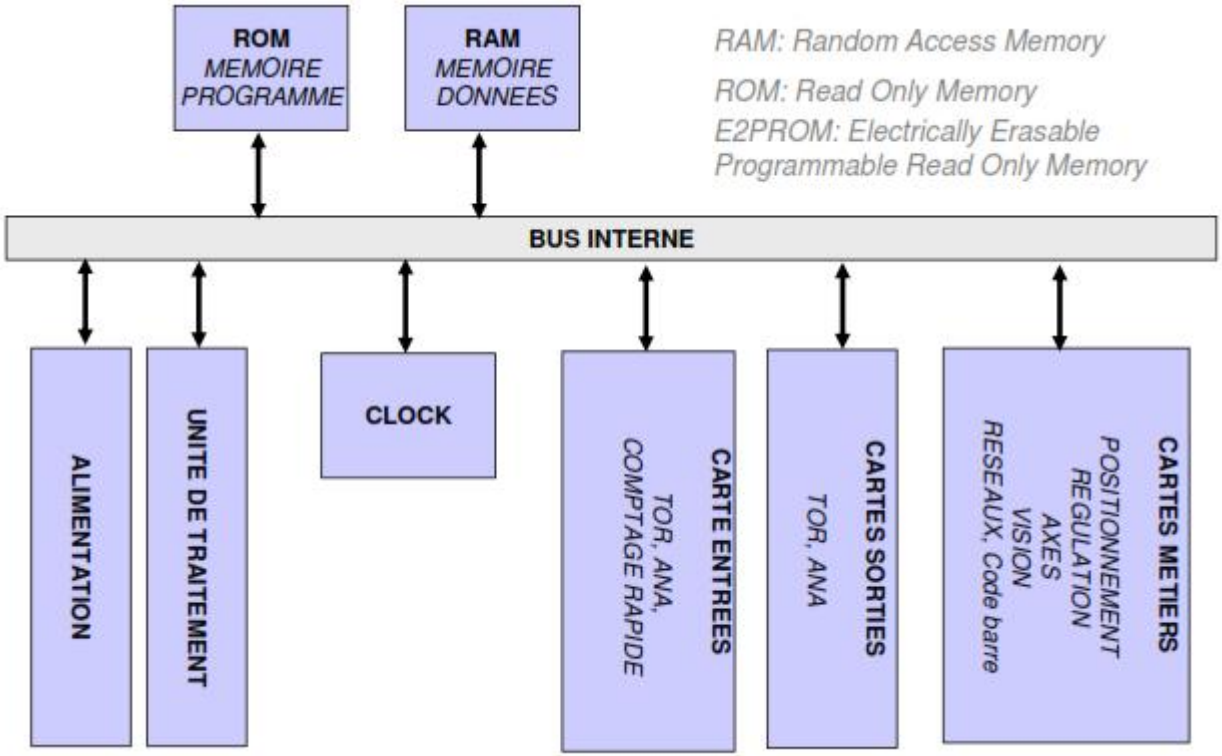


Figure 1 : Architecture d'un API.

Architecture d'un API


ROM
MEMOIRE
PROGRAMME
RAM
MEMOIRE
DONNEES
RAM: Random Access Memory
ROM: Read Only Memory
E2PROM: Electrically Erasable
Programmable Read Only Memory
BUS INTERNE
ALIMENTATION
UNITE DE TRAITEMENT
CLOCK
CARTE ENTREES
TOR, ANA,
COMPTAGE RAPIDE
CARTES SORTIES
TOR, ANA
CARTES METIERS
POSITIONNEMENT
REGULATION
AXES
VISION
RESEAUX, Code barre

II-Architecture d'un API

- Processeur:
  - Le processeur (UC), a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement
    - les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul PID, etc..).
  - Le processeur réalise également d'autres fonctions :
    - Gestion des entrées/sorties.
    - Surveillance et diagnostic de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
    - Dialogue avec le terminal de programmation, aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications des données.
  - Un ou plusieurs processeurs exécutent ces fonctions grâce à un micro logiciel préprogrammé dans une mémoire de commande (SE), ou mémoire système. Cette mémoire morte définit les fonctionnalités de l'automate. Elle n'est pas accessible à l'utilisateur.

II-Architecture d'un API

La mémoire:

destinée au stockage des instructions qui constituent le programme de fonctionnement de l'automatisme, ainsi que des données qui peuvent être :

- Des informations susceptibles d'évoluer en cours de fonctionnement de l'application.
- Des informations qui n'évoluent pas au cours de fonctionnement, mais qui peuvent en cas de besoin être modifiées par l'utilisateur (constants).

- Deux familles de mémoires sont utilisées dans les automates programmables :
  - Les mémoires vives, ou mémoires à accès aléatoire « Random Access Memory (RAM) ». Le contenu de ces mémoires peut être lu et modifié à volonté, mais il est perdu en cas de manque de tension (mémoire volatiles). Elles nécessitent par conséquent une sauvegarde par batterie. Les mémoires vives sont utilisées pour l'écriture et la mise au point du programme, et pour le stockage des données.
  - Mémoires à lecture seule, les informations ne sont pas perdues lors de la coupure de l'alimentation des circuits
    - ROM « Read Only Memory » : Elle est programmée par le constructeur et son programme ne peut être modifié.
    - PROM « Programmable ROM » : Elle est livrée non enregistrée par le fabricant. Lorsque celle-ci est programmée, on ne peut pas l'effacer
    - EPROM « Erasable PROM » : C'est une mémoire PROM effaçable par un rayonnement ultraviolet intense.
    - EEPROM « Electrically EPROM » : C'est une mémoire PROM programmable plusieurs fois et effaçable électriquement.
    - Mémoire Flash : C'est une mémoire EEPROM rapide en programmation. L'utilisateur peut effacer un bloc de cases ou toute la mémoire.

II-Architecture d'un API

- Le bus: est organisé en plusieurs sous ensembles destinés chacun à véhiculer un type bien défini d'informations :
  - Bus de données.
  - Bus d'adresses.
  - Bus de contrôle pour les signaux de service tels que tops de synchronisation, sens des échanges, contrôle de validité des échanges, etc..
  - Bus de distribution des tensions issues du bloc d'alimentation.

II-Architecture d'un API

- L'alimentation:
  - Elle élabore à partir d'un réseau 220V en courant alternatif, ou d'une source 12V/24V en courant continu, les tensions internes distribuées aux modules de l'automate.
  - A fin d'assurer le niveau de sûreté requis, elle comporte des dispositifs de détection de baisse ou de coupure de la tension réseau, et de surveillance des tensions internes. En cas de défaut, ces dispositifs peuvent lancer une procédure prioritaire de sauvegarde.

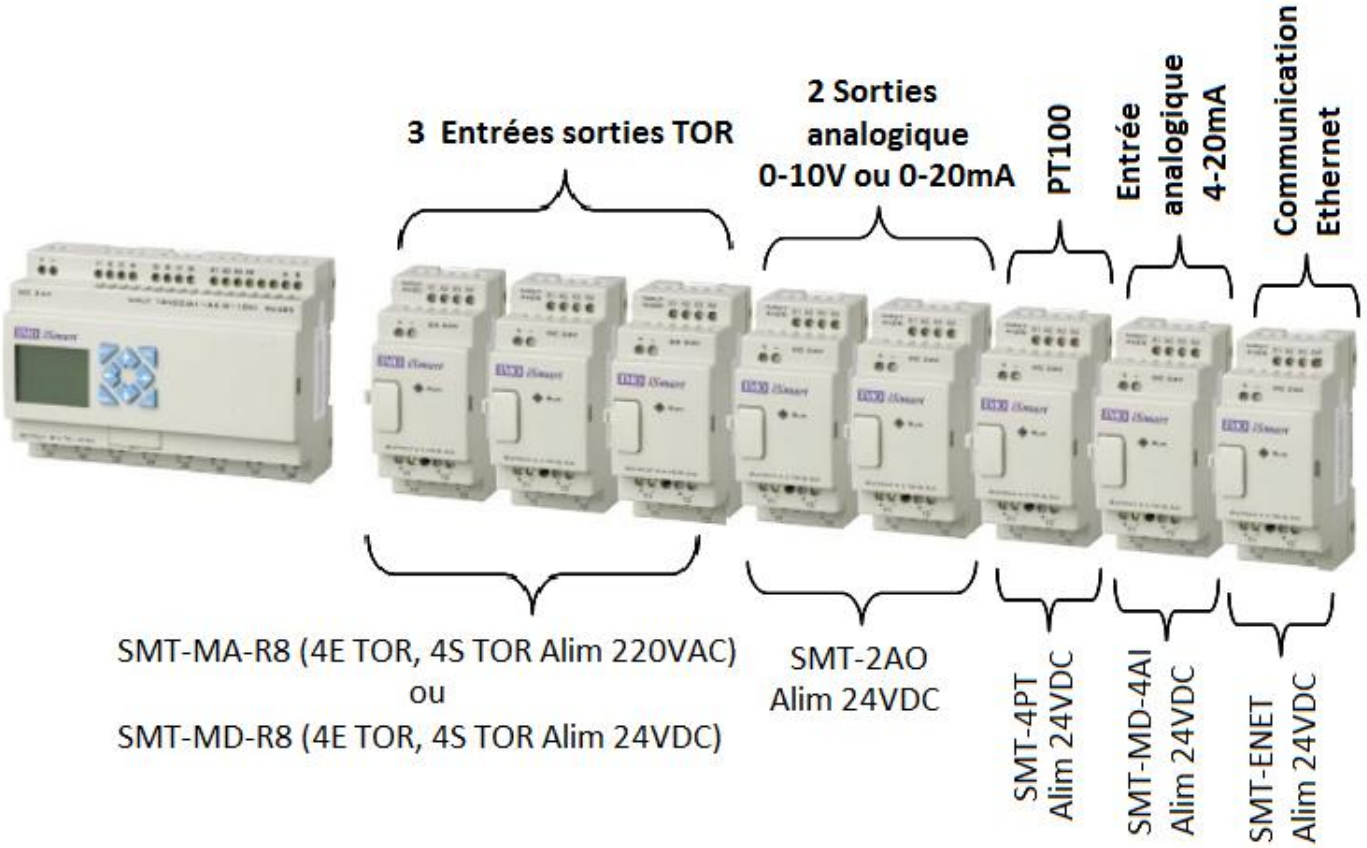
II-Architecture d'un API

- Les modules (ou cartes) d'entrées/sorties :
  - Les modules d'entrée sont destinés à recevoir l'information en provenance des capteurs et adapter le signal en le mettant en forme (signaux compréhensibles par l'automate), en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.
  - Les modules de sortie sont destinées à commander les pré-actionneurs et éléments des signalisations du système et adapter les niveaux de tensions de l'unité de commande à celle de la partie opérative du système (les ordres de commande et de signalisation).

II-Architecture d'un API

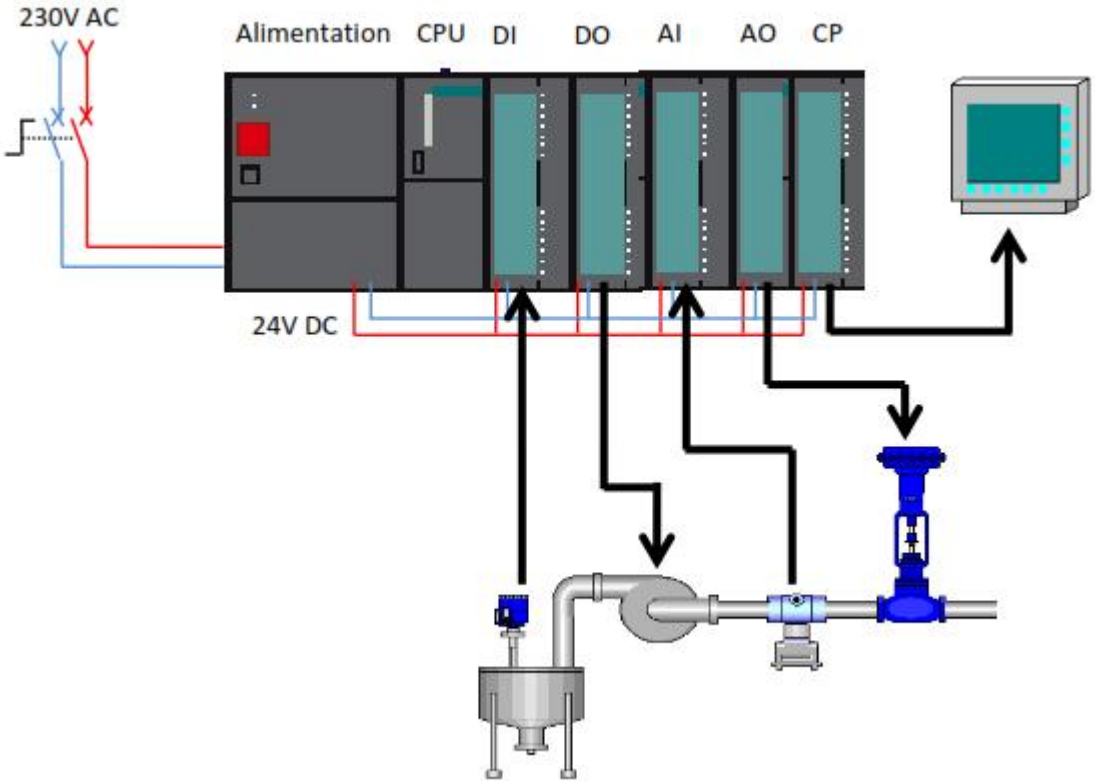
Les modules d'entrées / sorties (exemple)

Exemple



II-Architecture d'un API

Les modules d’entrées / sorties



Exemple d’utilisation des entrées / sorties

II-Architecture d'un API
Les entrées/sorties Logiques (DI/DO)

- TOR (Tous ou rien):
  - l'information à traiter ne peut prendre que deux états (marche / arrêt) , donc deux niveaux logiques 1 ou 0 (entrée numerique).
  - Exemple : capteur de fermeture de porte (entrée)
  - allumage d'une lampe

Microrupteur

Bouton poussoir

II-Architecture d'un API

Les entrées /sorties Logiques (DI/DO)

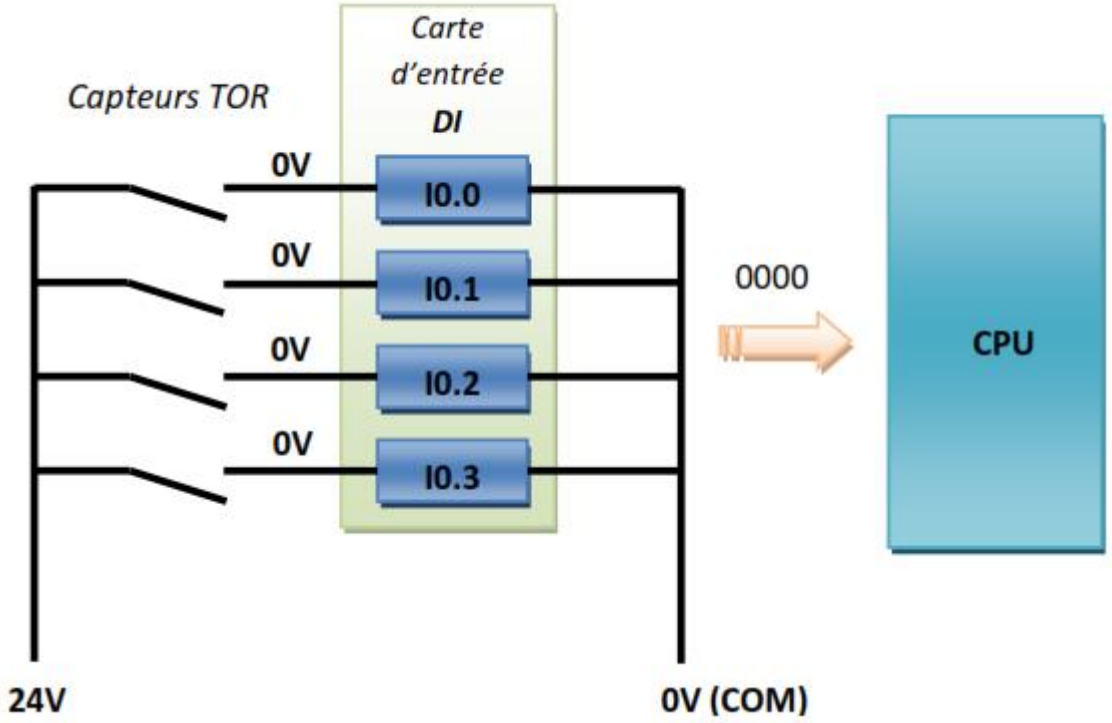
témoin lumineux indicateur

Relais À Usage Général

II-Architecture d'un API

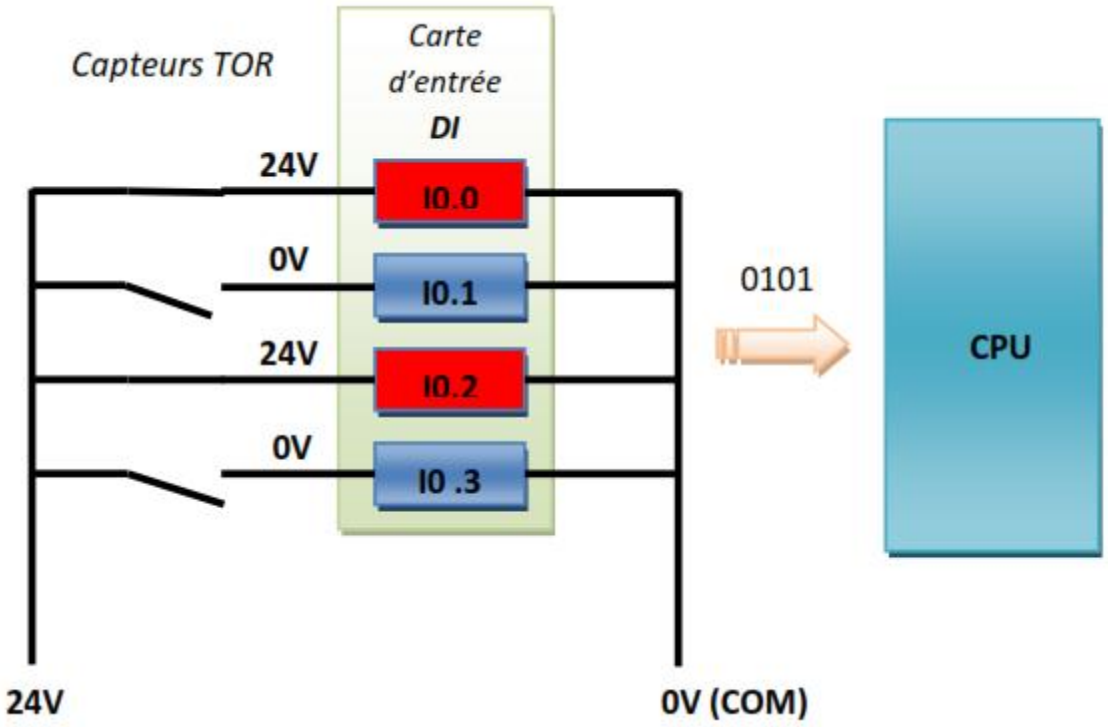
Les entrées/sorties Logiques (DI/DO)

La CPU test en boucle l'état des entrées,
Un bit est réservé pour chaque entrée pour stocker l'état de l'entrée



Principe de connexion des entrées état au repos

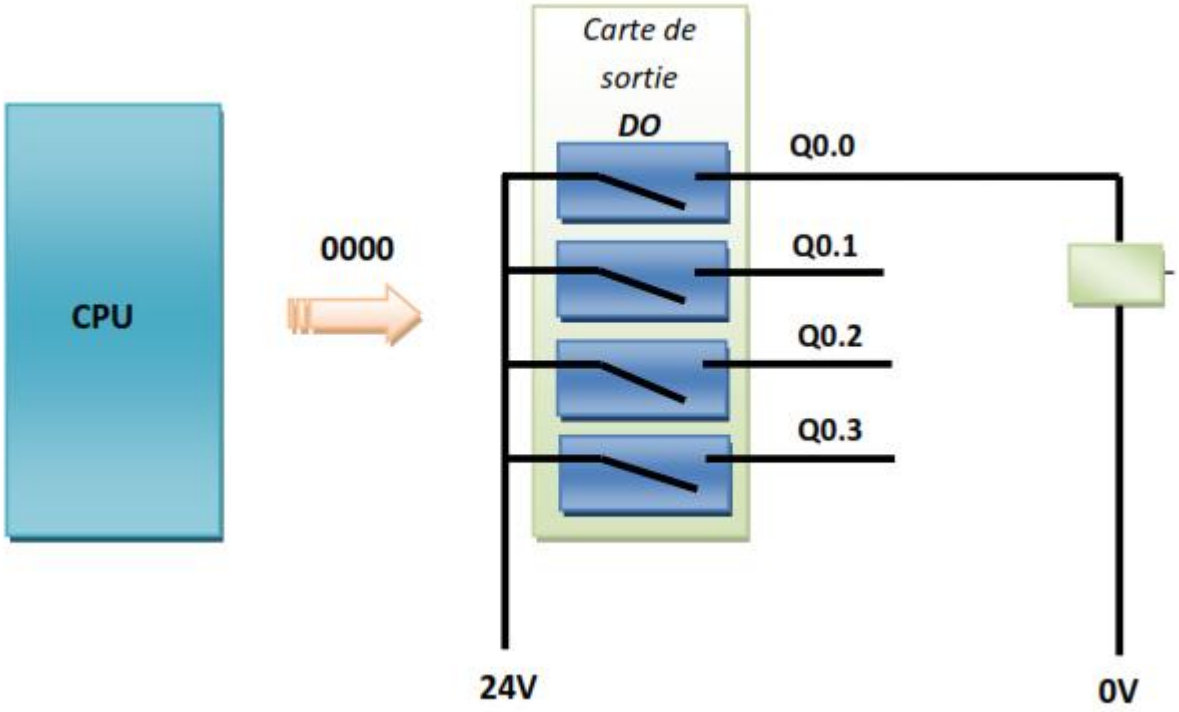
II-Architecture d'un API
Les entrées/sorties Logiques (DI/DO)



Principe de connexion des entrées état actionnées

II-Architecture d'un API

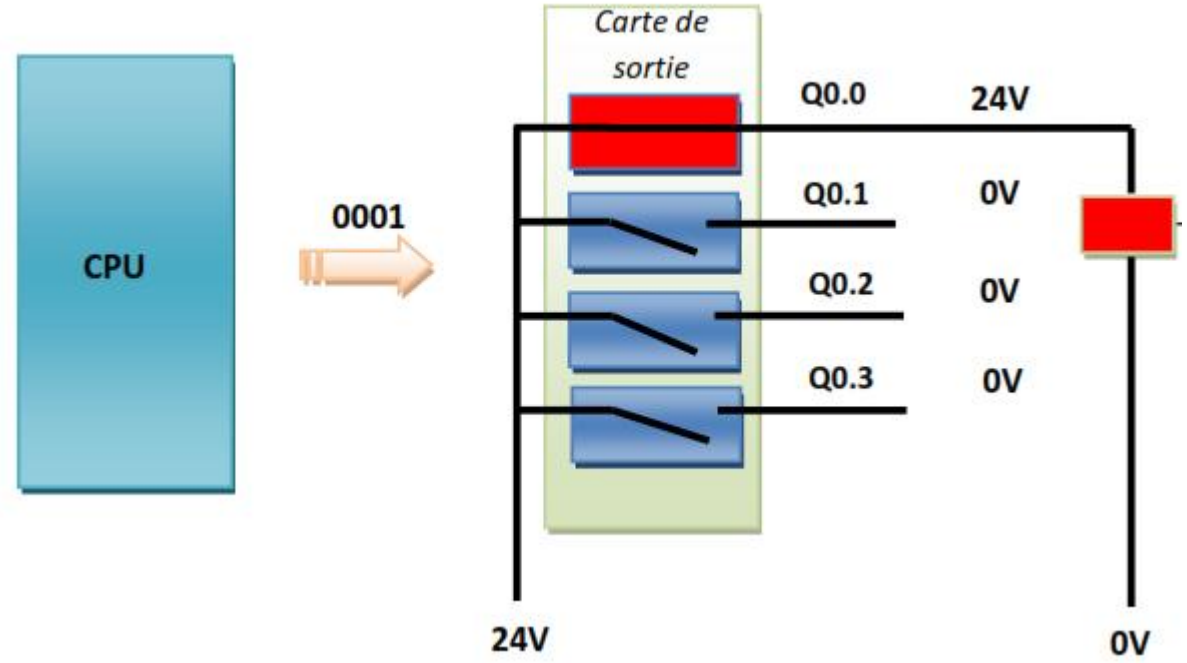
Les entrées/sorties Logiques (DI/DO)



Principe de connexion des sorties état au repos

II-Architecture d'un API

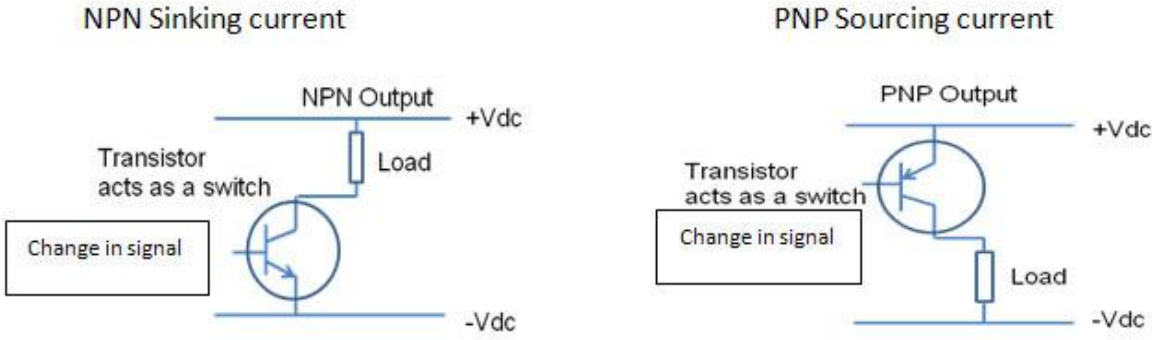
Les entrées/sorties Logiques (DI/DO)



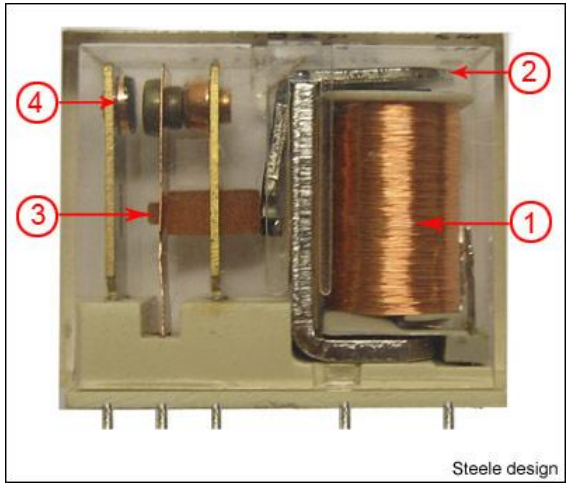
Principe de commande des sorties état actionnée

II-Architecture d'un API
Sortie logique Relais /Statique

- Statique = sortie en tout ou rien (transistor)



- Relais = sortie en tout ou rien (contact sec)



II-Architecture d'un API
Adressage variable

Variables : Notation normalisées CEI 61131 des variables

Bits :
- %IXxxx entrée TOR
- %QXxxx sortie TOR
- %MXxxx bit interne
- %SXxxx bit système

Mots 16bits:
- %IWxxx mot entrée
- %QWxxx mot sortie
- %MWxxx mot interne
- %SWxxx mot système

%KWxxx mot constant

Les outils de développement permettent d'affecter un nom symbolique à chacune des variables physique de l'automate ; cette possibilité améliore la lisibilité d'une application et permet de modifier l'affectation des E/S sans modifier le programme

II-Architecture d'un API
Adressage variable

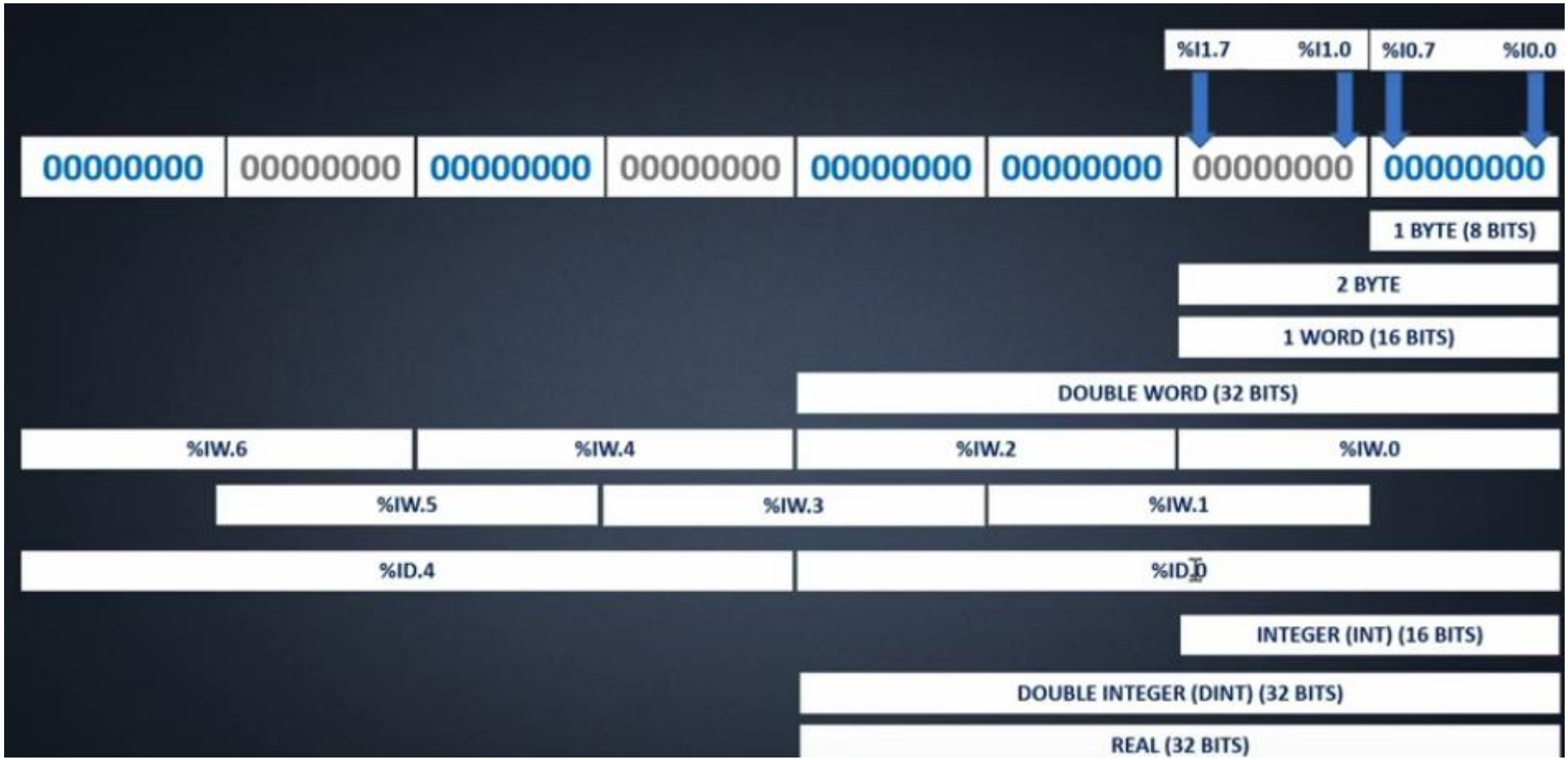
- L ’adresse variable xxx est structurée pour tenir compte de la structure matérielle de l ’automate
- <adresse> = [réseau].<numéro de chassis>.<numero d ’appareil>.<numero de voie>

  Exemple : %IX0.3.5

  entrée TOR n°5 du module d ’entrée 3 sur le chassis 0

II-Architecture d'un API

Adressage variable (exemple siemens)



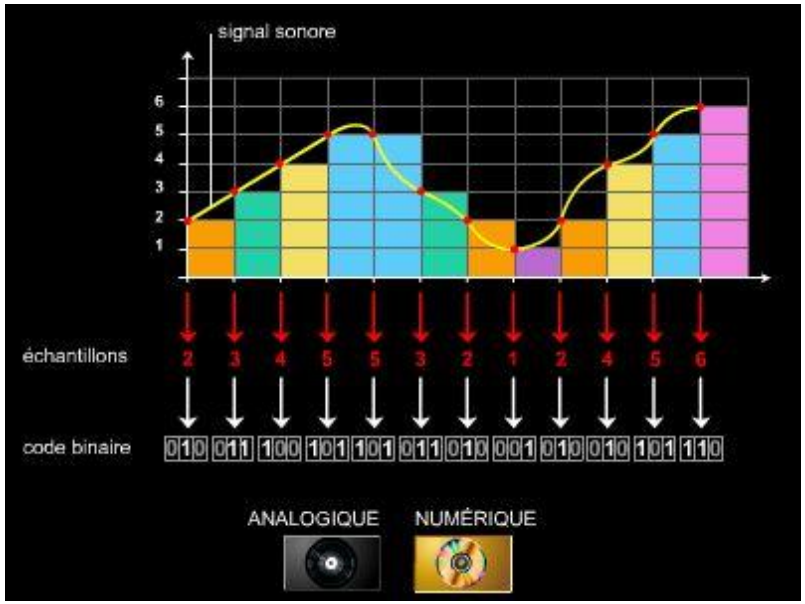
II-Architecture d'un API
La conversion des signaux analogiques :

- Un convertisseur analogique-numérique est un montage électronique dont la fonction est de générer à partir d'une valeur analogique, une valeur numérique, proportionnelle à la valeur analogique entrée.

- Les E/S analogiques sont caractérisées par l'amplitude du signal (VH et VL), par la vitesse de conversion et par la grandeur électrique (courant ou tension).
Il existe, au niveau des entrées, trois types d'entrées analogiques :
Haut niveau : 0-10V, 0-20mA, 4-20mA
Pour thermocouple : 0-20mV, 0-50mV, 0-100mV
Pour sondes Pt100 : 0-100mV, 0-250mV, 0-400mV

De même, les sorties analogiques peuvent être différenciées selon ces intervalles :
0-10V, 0-5V, ±10V, ±5V, 0-20mA, 4-20mA.

II-Architecture d'un API
La conversion des signaux analogiques :



- Les entrées analogiques

transforment une grandeur analogique variant d'une façon continue en un code numérique, généralement sur 11,13 OU 15 bits plus un bit de signe. Ces entrées disposent d'un seul convertisseur A/N (CAN), elles sont scrutées les unes à la suite des autres par un multiplexeur (MUX). Par contre, les sorties analogiques disposent d'un seul convertisseur par voie.

II-Architecture d'un API
La conversion des signaux analogiques :

- Deux fonctions sont utilisées pour manipuler une grandeur analogique
  - Normaliser: transforme une valeur numérique en une valeur comprise entre 0 et 1.
    - Exemple : pour l'automate siemens S7 1200 , dispose deux entrées analogiques, la valeur numérique est codée sur 15 bits + 1 bit de signe
      - Donc min=0

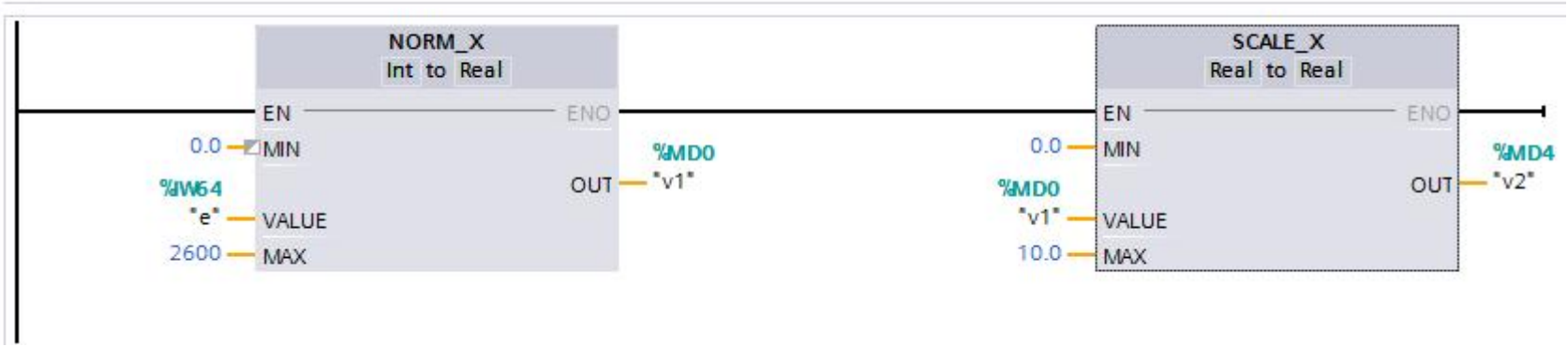
        max = 2^15 – 1 = 32767

      MAIS pour des raisons techniques de cet automate le max=27648

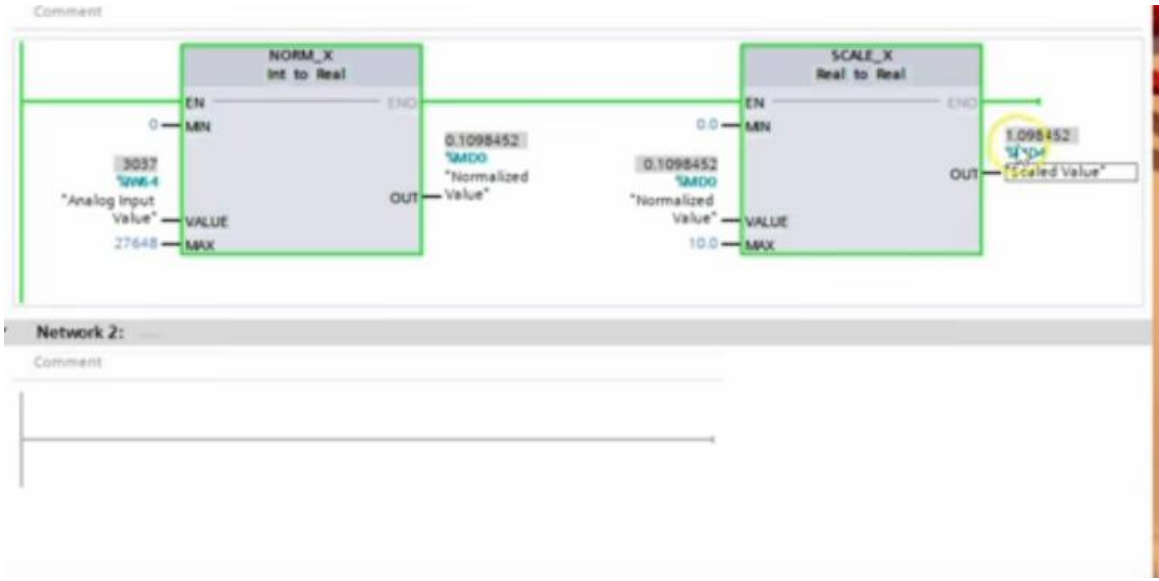
      Dans ce cas une valeur de 3037 , la valeur normalisée = 0.1098452

II-Architecture d'un API
La conversion des signaux analogiques :

- La fonction de mise en échelle (scaling): permet le passage d'une valeur normalisée à une valeur calculée dans un échelle donné
  - Exemple : l'entrée analogique de l'automate S7 1200 mésure des tensions comprises entre 0 et 10 v , si:
    - La valeur numérisée d'une tension en entrée = 3037
    - La valeur normalisée = 0.1098452
    - Donc dans l'interval 0 -10 v la valeur 0.1098452 = 1.098452 v

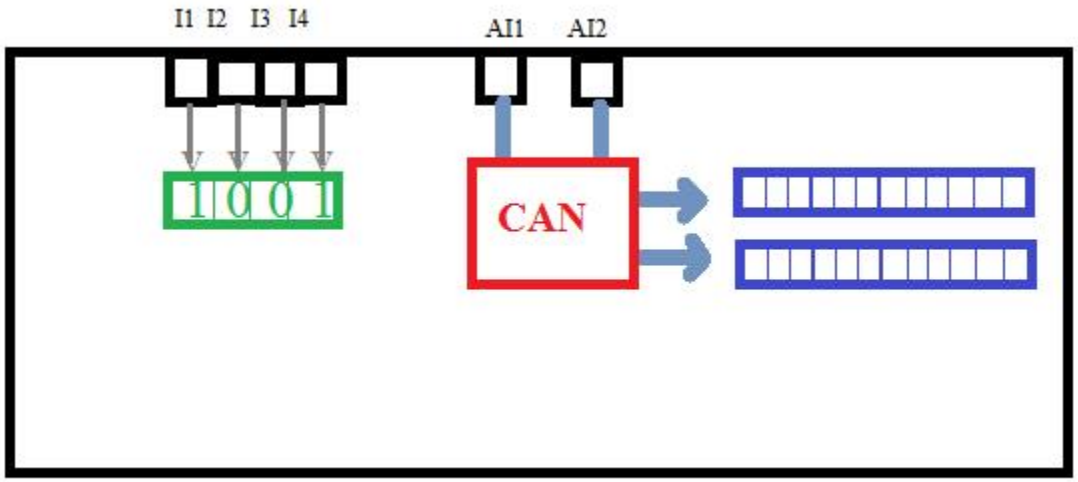


II-Architecture d'un API
La conversion des signaux analogiques :



II-Architecture d'un API

- Exemple d'un module 4 entrées logiques et deux entrées analogiques à 12 bits



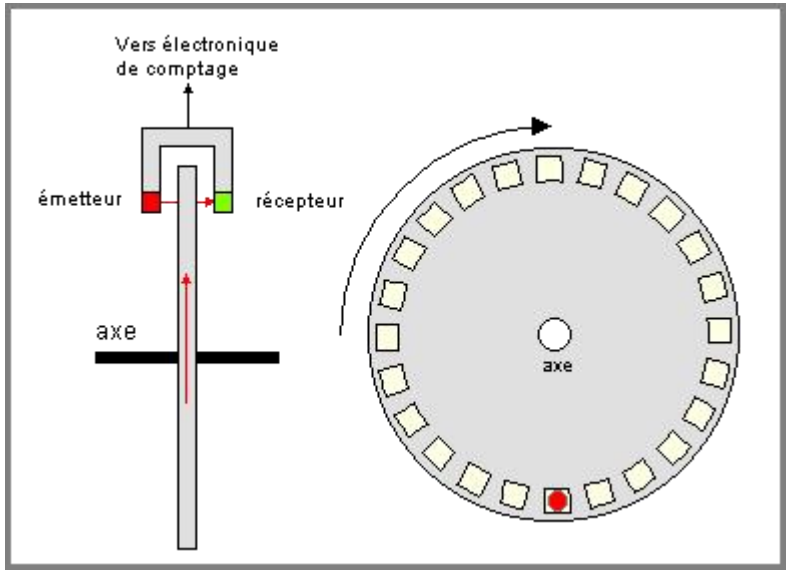
II-Architecture d'un API
Les modules de comptage

- Le module compteur rapide (”High Speed Counter” ou HSC), fournit un traitement autonome des signaux à impulsion rapide. Le comptage matériel s'impose lorsque la vitesse d'acquisition est très élevée.
- Un module de comptage comporte un compteur/décompteur dont le contenu est incrémenté ou décrémenté par des impulsions en provenance de l'extérieur,
- Exemple : pour les applications de contrôle industriel telles que :
  - La mesure de vitesse
  - Le contrôle de mouvement…
- Le traitement autonome, c’est la capacité du module à lire les entrées, à traiter les informations des entrées de comptage *et à contrôler les sorties* sans avoir besoin de communiquer avec l’UC.

II-Architecture d'un API
Les modules de comptage

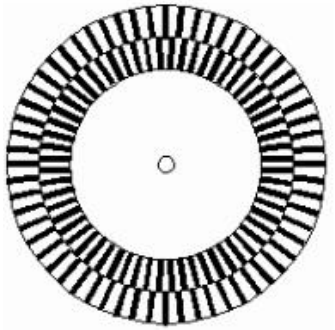
- exemple

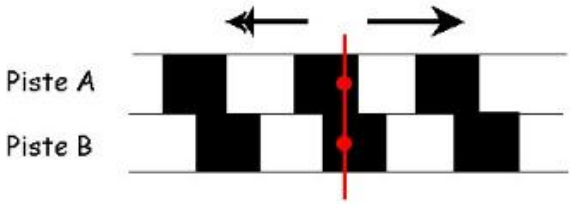
• Exemple: Comptage des impulsions d 'une roue dentée

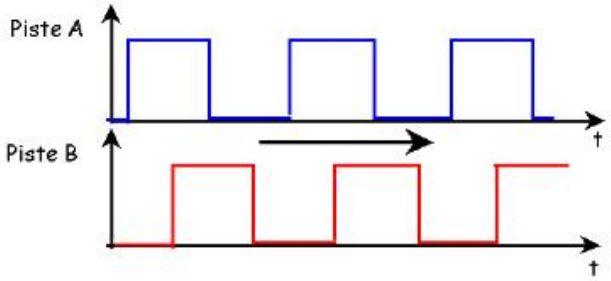


II-Architecture d'un API
Les modules de comptage

- Codeur AB
  - La valeur numérique est l ’intégration signée des impulsions reçues





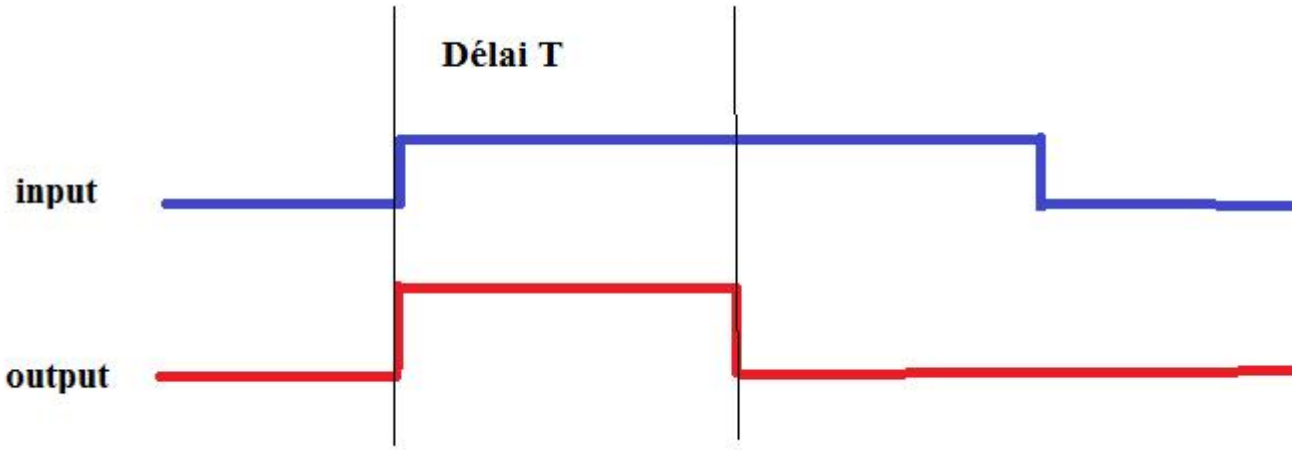


II-Architecture d'un API
Les modules de comptage

- Exemple : codeur rotatif incrémental

II-Architecture d'un API
Modules de temporisation

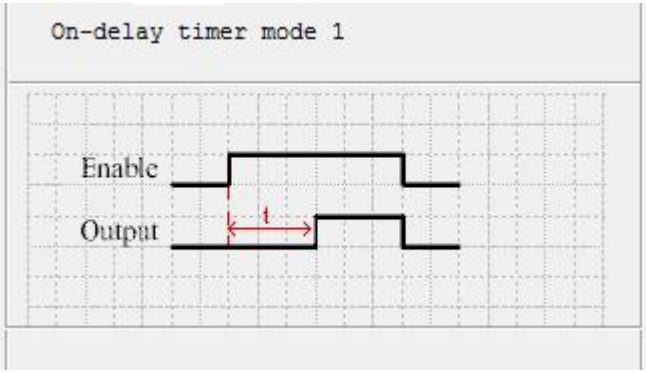
- Temporisation
  - Delay : passage de la sortie de 0 à 1 au début de temporisation, après l'écoulement du délai la sortie revient à 0

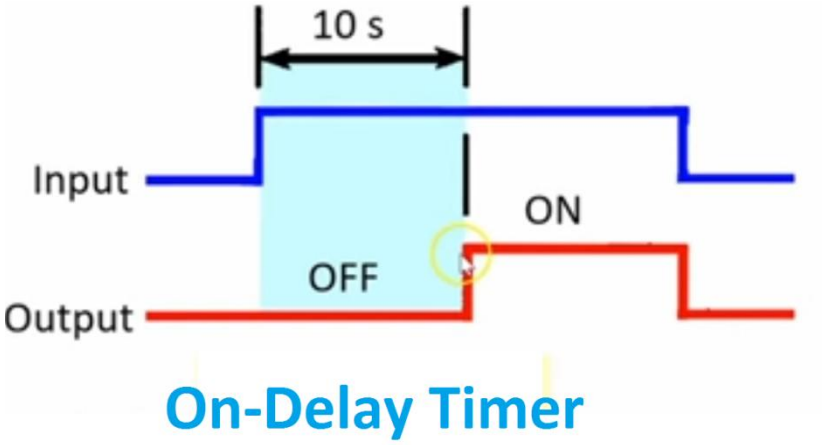


II-Architecture d'un API
Modules de temporisation

Temporisation

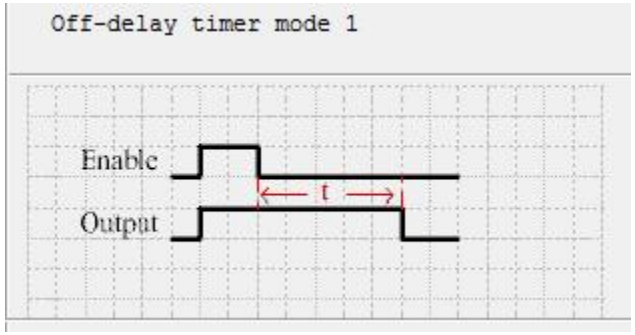
- On delay : passage de la sortie de 0 à 1 à la fin de la temporisation

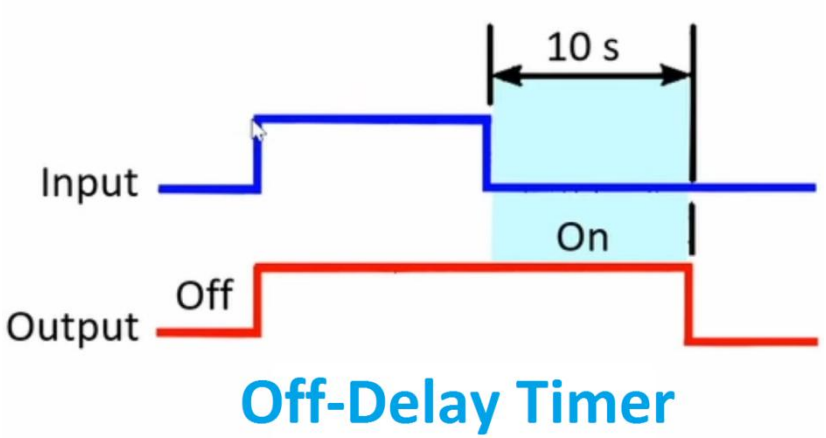




II-Architecture d'un API
Modules de temporisation

- Off delay : temporisation avec passage de la sortie de 1 à 0 à la fin de la temporisation

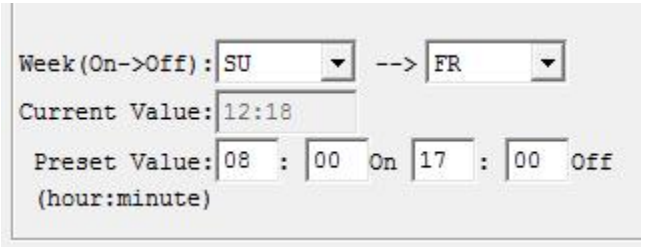




II-Architecture d'un API
Modules de temporisation

- RTC : horloge temps réel:

  exemple: éclairage tous les soirs à une heure précise

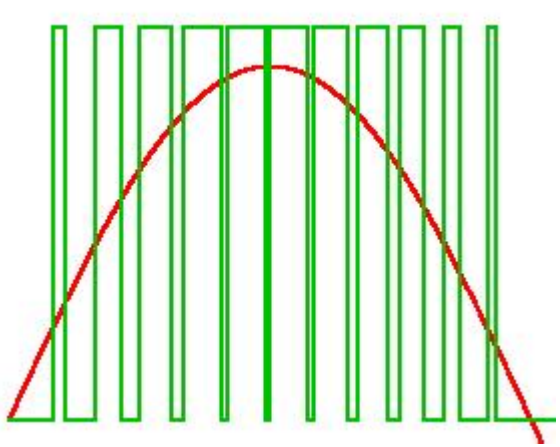


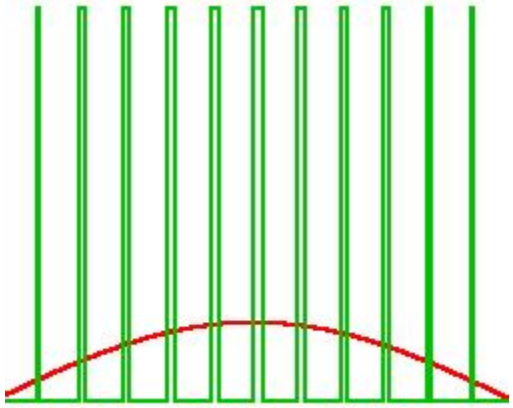
II-Architecture d'un API
Sorties modulées MLI

- Modulation de Largeur d ’impulsion MLI (PWM)
  - La modulation de largeur d'impulsions (MLI ; en anglais : Pulse Width Modulation, soit PWM), est une technique couramment utilisée pour synthétiser des signaux continus à l'aide de circuits à fonctionnement tout ou rien, ou plus généralement à états discrets.

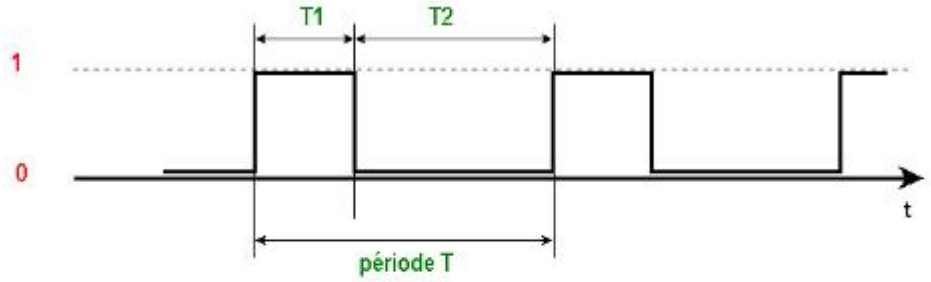
II-Architecture d'un API
Sorties modulées MLI

- Principe de modulation MLI





II-Architecture d'un API
Sorties modulées MLI



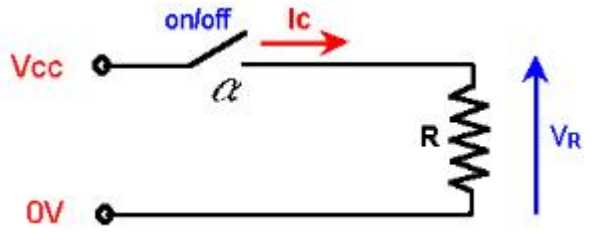
Grandeurs caractéristiques:

- ⇨ période T
- ⇨ fréquence F = 1/T
- ⇨ rapport cyclique $\alpha = \frac{T_1}{T_1 + T_2}$

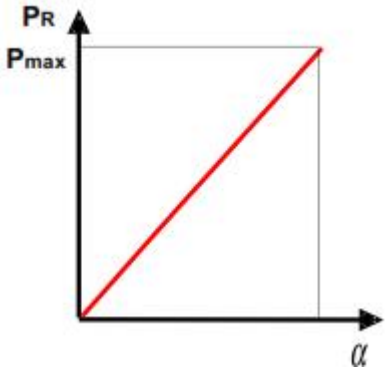
- Exemple
  - T1=20 ms T2= 60 ms donc α= 20/80 = ¼
  - F=1/0.08= 12.5 Hz

II-Architecture d'un API
Sorties modulées MLI

- La relation puissance/rapport cyclique est linéaire :



$$P_R = \alpha \, P_{max}$$



⇨ Applications: contrôle de la luminosité d 'un voyant, modulation d 'un chauffage à faible puissance, d 'une enceinte thermostatée

II-Architecture d'un API

Automate – lecture d'une fiche technique

Fiche technique automate siemens s7 1214c

Ecran	
Avec afficheur	Non
Entrées TOR	
Nombre d'entrées TOR	14; intégré
• dont entrées utilisables pour les fonctions technologiques	6; HSC (compteur rapide)
Sorties TOR	
Nombre de sorties TOR	10
• dont les sorties rapides	4; Sortie de trains d'impulsions 100 KHz
Tension de sortie	
• pour état log. "0", max.	0,1 V; avec charge 10 kohm
• pour état log. "1", mini	20 V

II-Architecture d'un API

Automate – lecture d'une fiche technique

Sorties relais	
• Nombre max. de sorties à relais, intégrées	0
Longueur de câble	
• Longueur de câble blindé, maxi	500 m
• Longueur de câble non blindé, max.	150 m
Entrées analogiques	
Nombre d'entrées analogiques	2
Voies intégrées (EA)	2; 0 à 10 V
Sorties analogiques	
Nombre de sorties analogiques	0
Formation de la valeur analogique	
Temps d'intégration et de conversion/résolution par voie	
• Résolution avec domaine de dépassement (bits avec signe), maxi	10 bit

II-Architecture d'un API

Automate – fiche technique

Fonctions intégrées	
Nombre de compteurs	6
Fréquence de comptage (compteurs), maxi	100 kHz
Fréquencemètre	Oui
Positionnement en boucle ouverte	Oui
Régulateur PID	Oui
Degré et classe de protection	
Degré de protection selon EN 60529 • IP 20	Oui
programmation	
Langage de programmation	
— CONT	Oui
— LOG	Oui
— SCL	Oui

III- Mise en œuvre d'un automate programmable industriel
* Choix d'un API

- Le choix d'un API est fonction de la partie commande à programmer. On doit tenir compte de plusieurs critères.
  - Nombres d'entrées/sorties intégrés.
  - Temps de traitement (scrutation).
  - Capacité de la mémoire.
  - Nombre de compteurs.
  - Nombre de temporisateurs

Transistor PNP et NPN

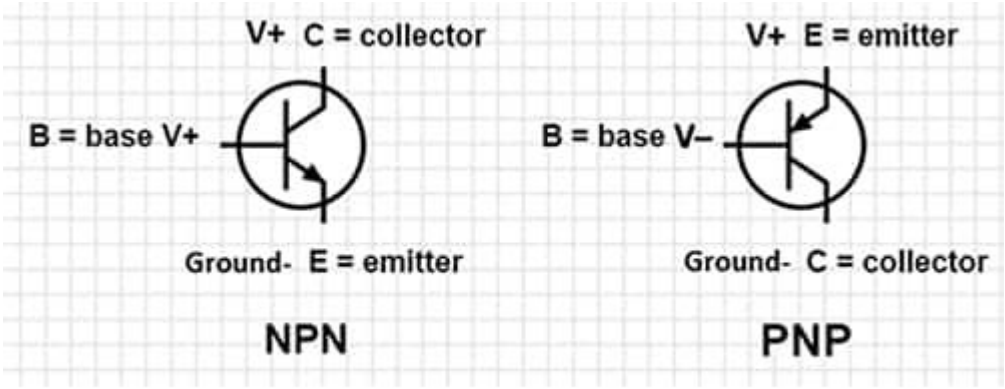
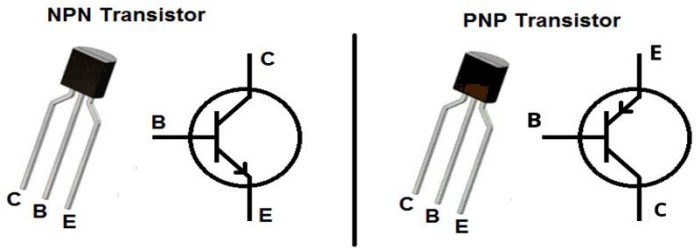


Figure 4a

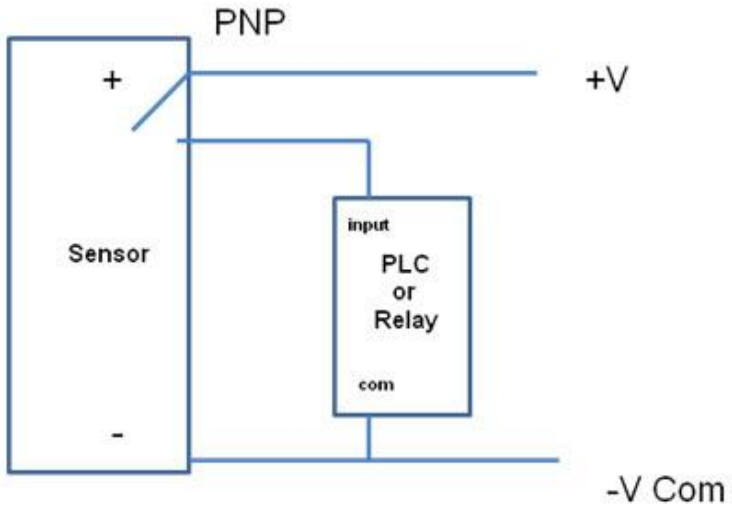
Figure 4b

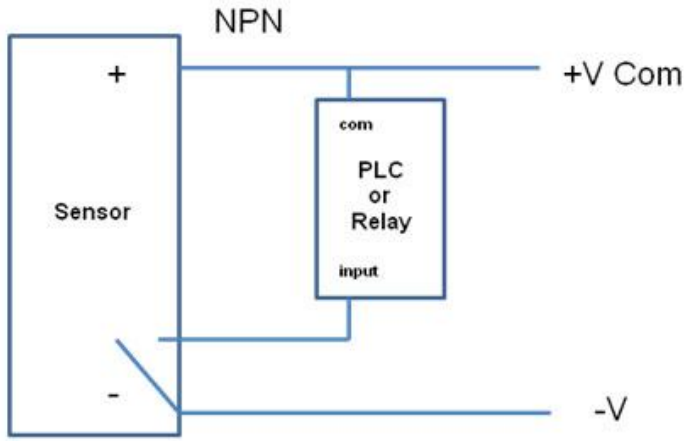


III- Mise en œuvre d'un automate programmable industriel

Les entrées/sorties Analogiques

- Les entrées PNP et NPN

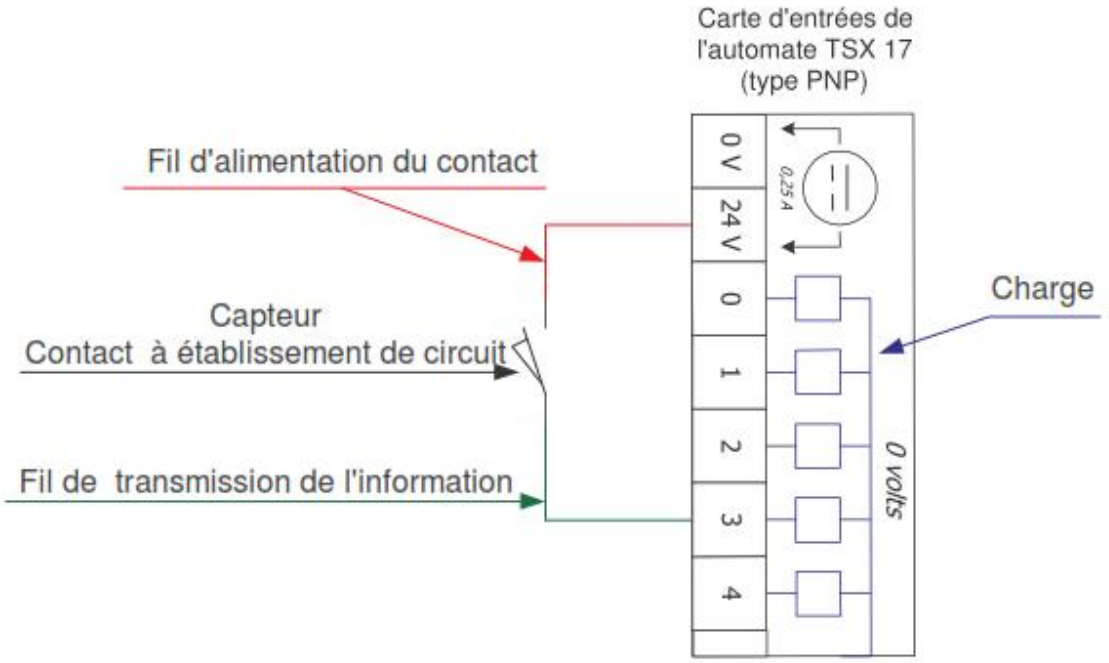




III- Mise en œuvre d'un automate programmable industriel

Les entrées/sorties Analogiques

- Exemple de raccordement d'un capteur deux fils, type PNP



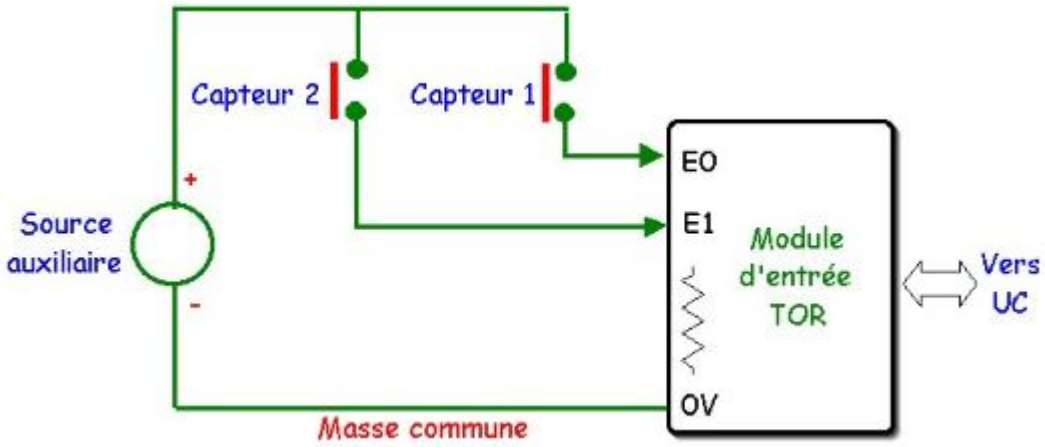
III- Mise en œuvre d'un automate programmable industriel

Raccordement E/S TOR

- Deux type de fonctionnement
  - API en logique positive
    - 12v/24v passe les E/S à l'état 1
  - API en logique négative
    - 0v passe les E/S à l'état 1

III- Mise en œuvre d'un automate programmable industriel
Raccordement E/S TOR

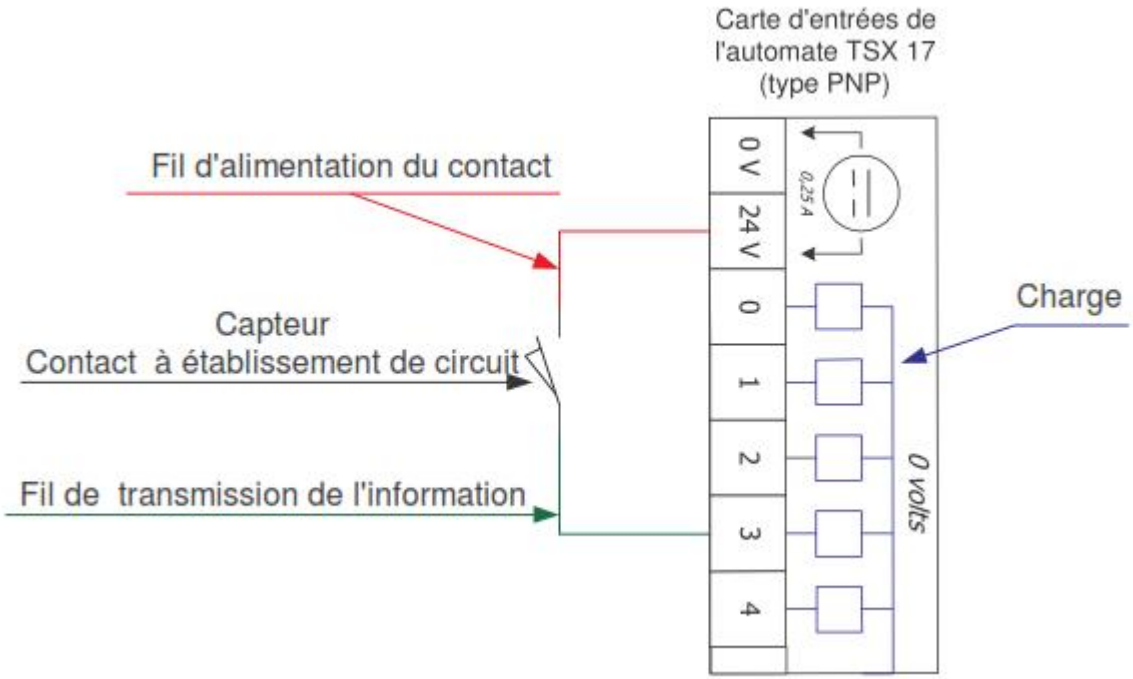
- Connexion des entrées TOR
  - Cas de contacts secs



III- Mise en œuvre d'un automate programmable industriel

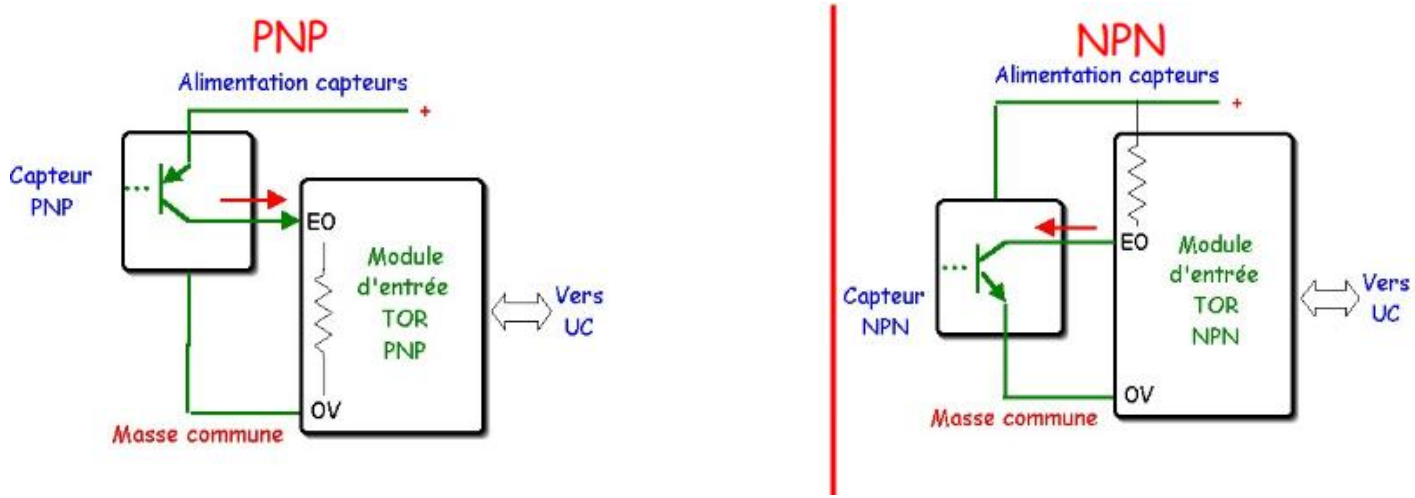
Raccordement E/S TOR

- Exemple de raccordement de contacts secs avec la carte d'entrée TSX 17



III- Mise en œuvre d'un automate programmable industriel
Raccordement E/S TOR

- Cas de contacts statiques

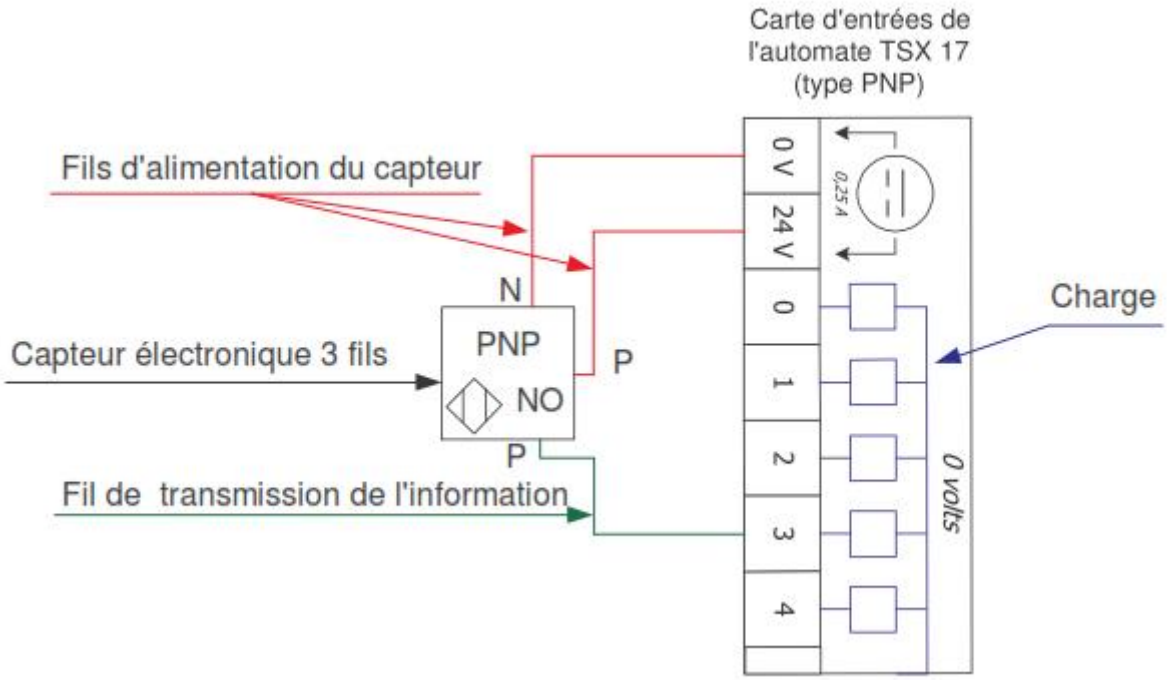


Sur ce type de capteur, on distingue 2 circuits :
· L'alimentation en général fournie par l'A.P.I.(2 fils).
· La transmission du signal réalisée par le 3 ème fil.
Le capteur étant alimenté, dès que l'information à acquérir est présente, un signal électrique est transmis vers l'entrée automate par le3 ème fil (commutation électronique).

III- Mise en œuvre d'un automate programmable industriel

Raccordement E/S TOR

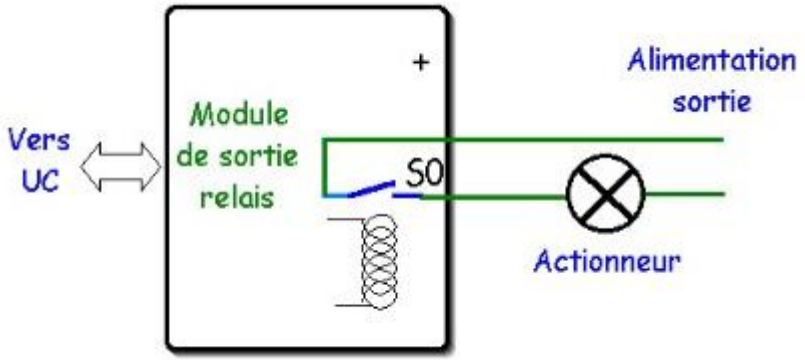
- Exemple de raccordement d'un capteur trois fils, type PNP avec la carte d'entrées TSX 17



III- Mise en œuvre d'un automate programmable industriel

Raccordement E/S TOR

- Connexion des sorties
  - Sorties relais

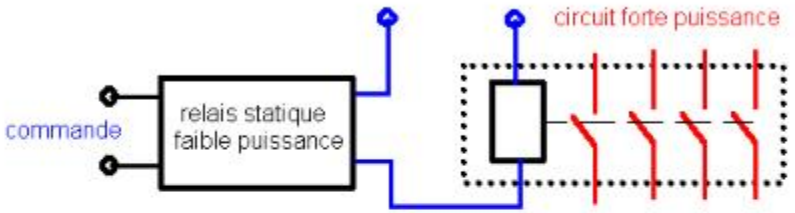


Isolation entre l'automate et les sorties

III- Mise en œuvre d'un automate programmable industriel

Raccordement E/S TOR

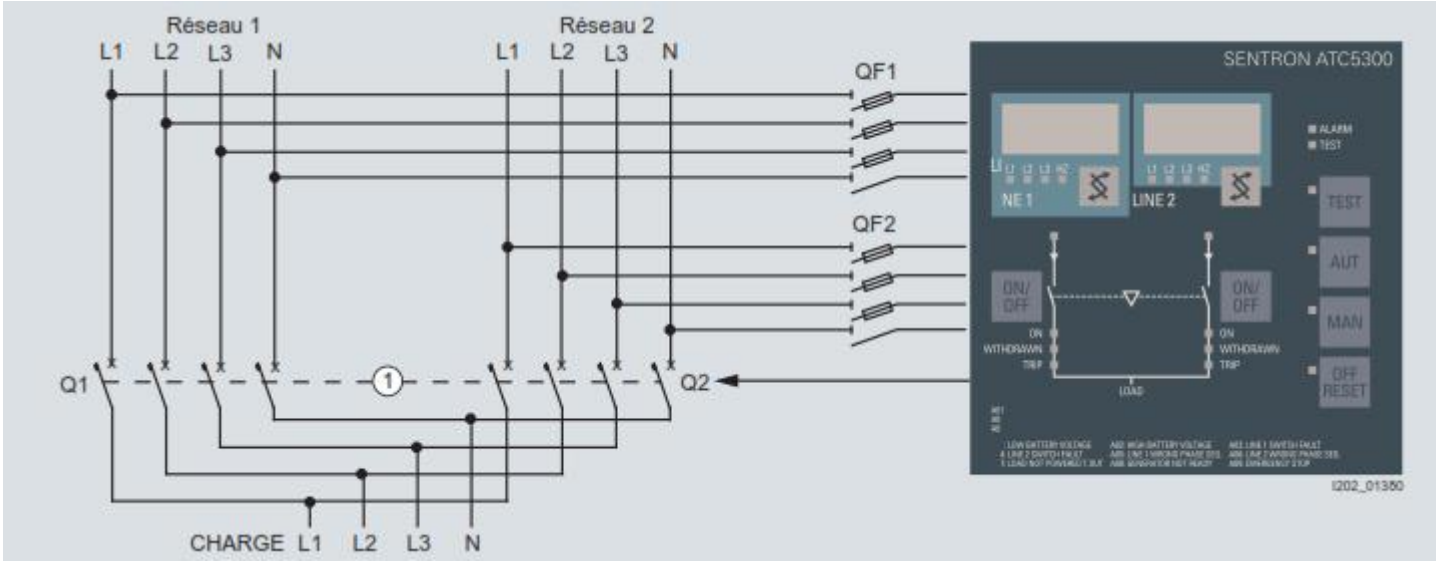
⇨ Cascade de pré-actionneurs



+

III- Mise en œuvre d'un automate programmable industriel
Raccordement E/S TOR

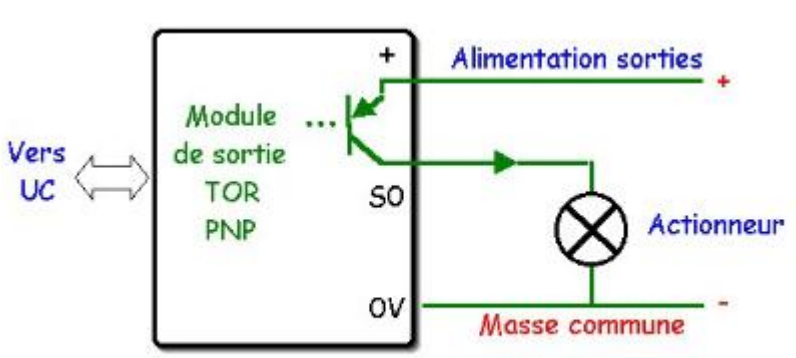
- Exemple



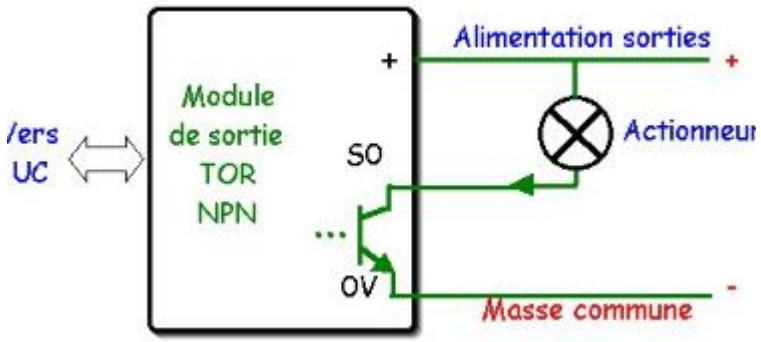
Le commutateur automatique de réseau 3KC ATC5300 est utilisé lorsqu'il est impératif de ne pas avoir de perte de courant, par ex. dans un hôpital, en liaison avec des alimentations sans coupure et pour les processus industriels qui ne peuvent en aucun cas être interrompus.

III- Mise en œuvre d'un automate programmable industriel
Raccordement E/S TOR

- Connexion des sorties
  - Sorties statiques



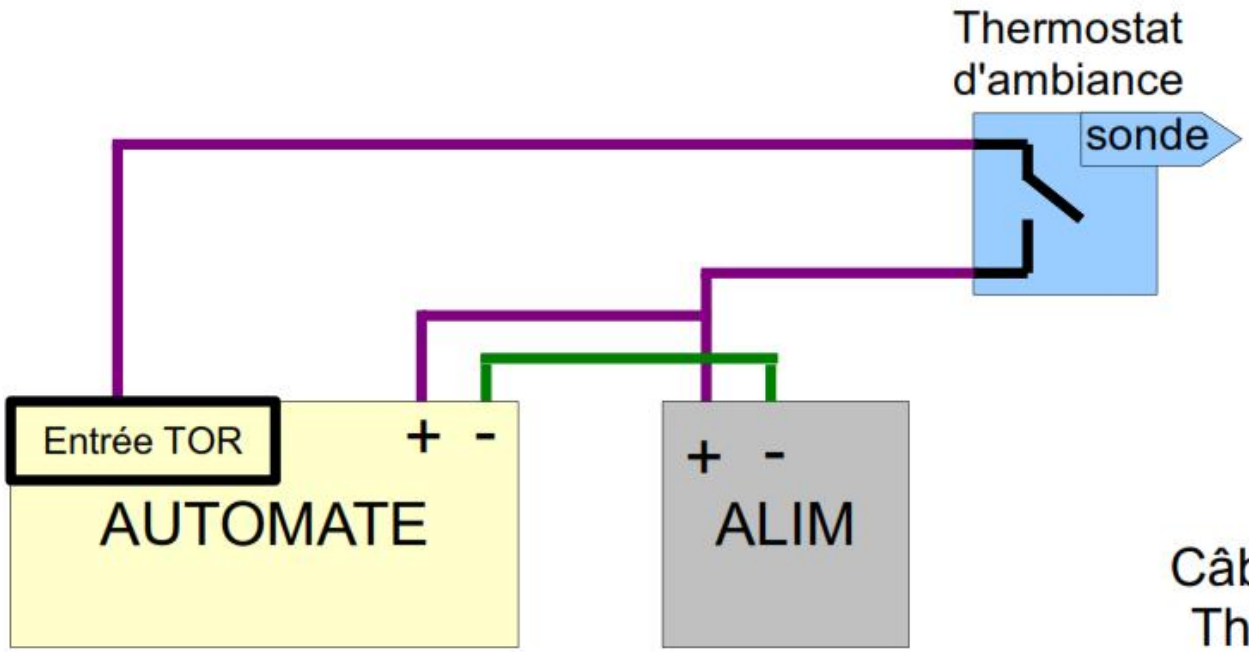
PNP



NPN

III- Mise en œuvre d'un automate programmable industriel

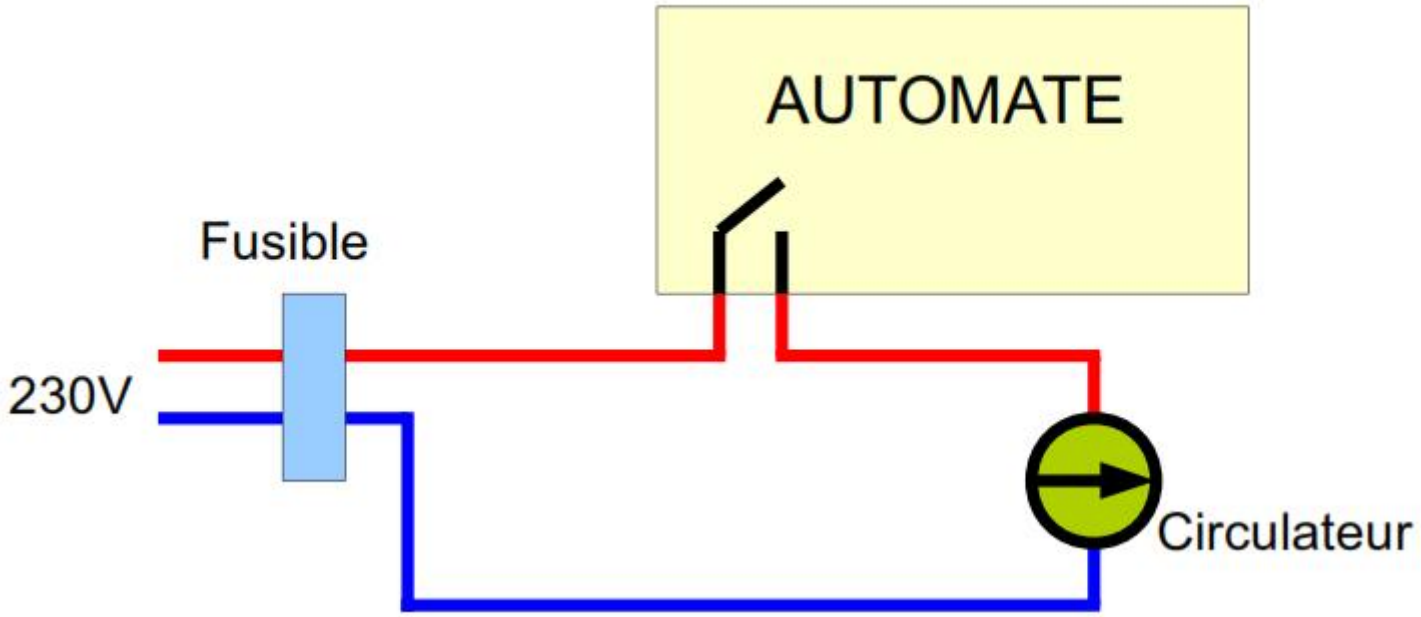
Exemples de câblage (E/S) automate



Câblage d'un Thermostat d'ambiance à contact sec (sans potentiel)

III- Mise en œuvre d'un automate programmable industriel

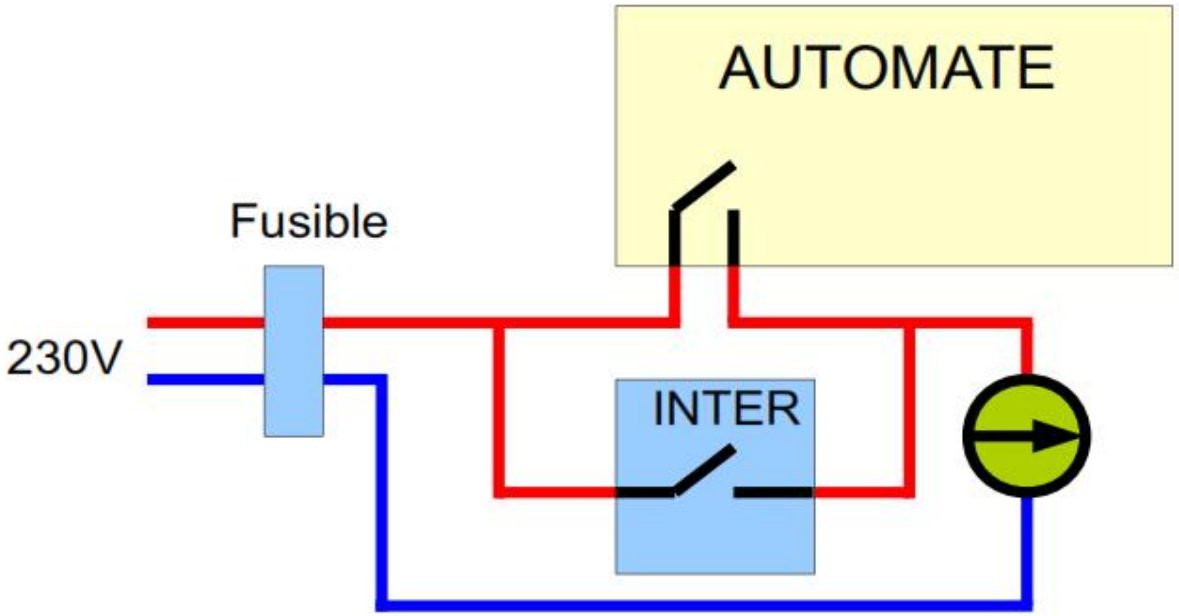
Exemples de câblage (E/S) automate



Câblage pour circulateur,
vanne trois voies monostable,
électrovanne, ampoule etc...

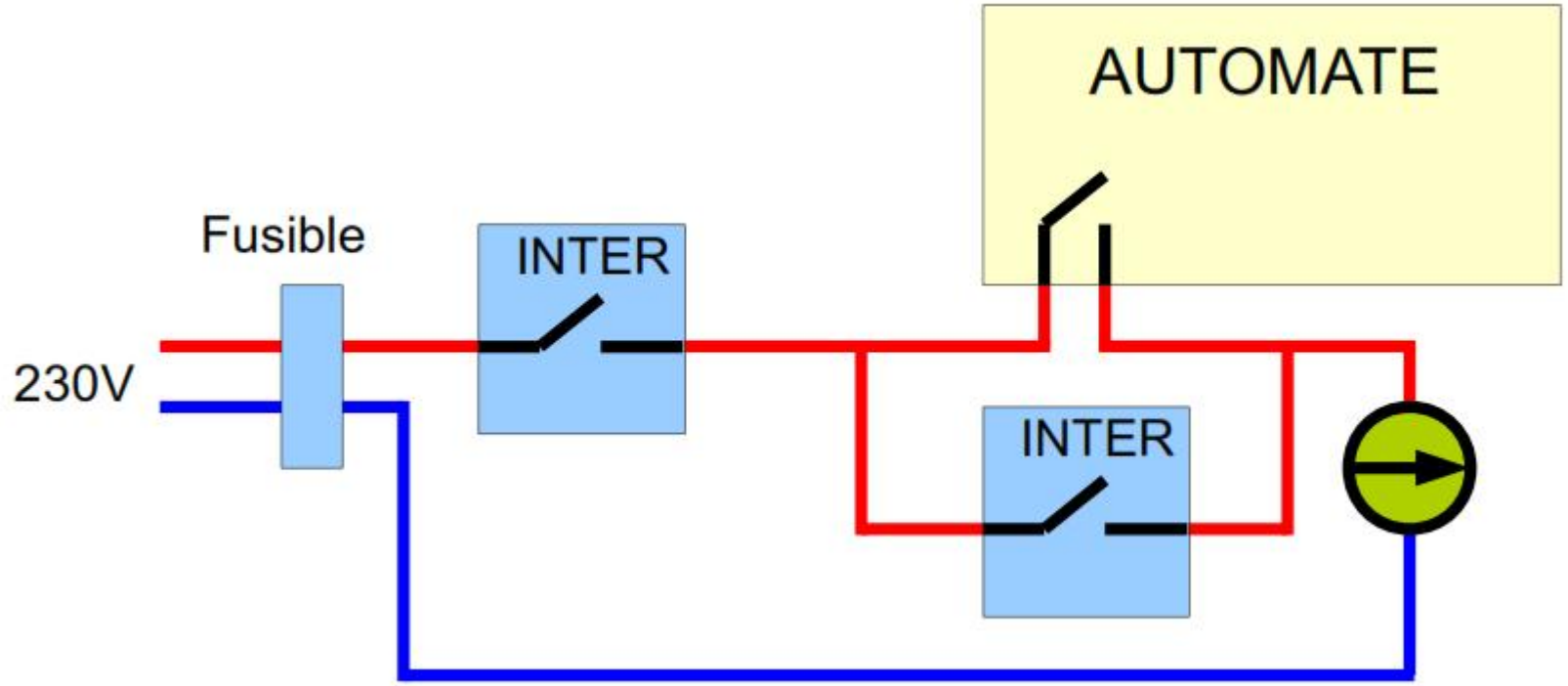
III- Mise en œuvre d'un automate programmable industriel

Exemples de câblage (E/S) automate



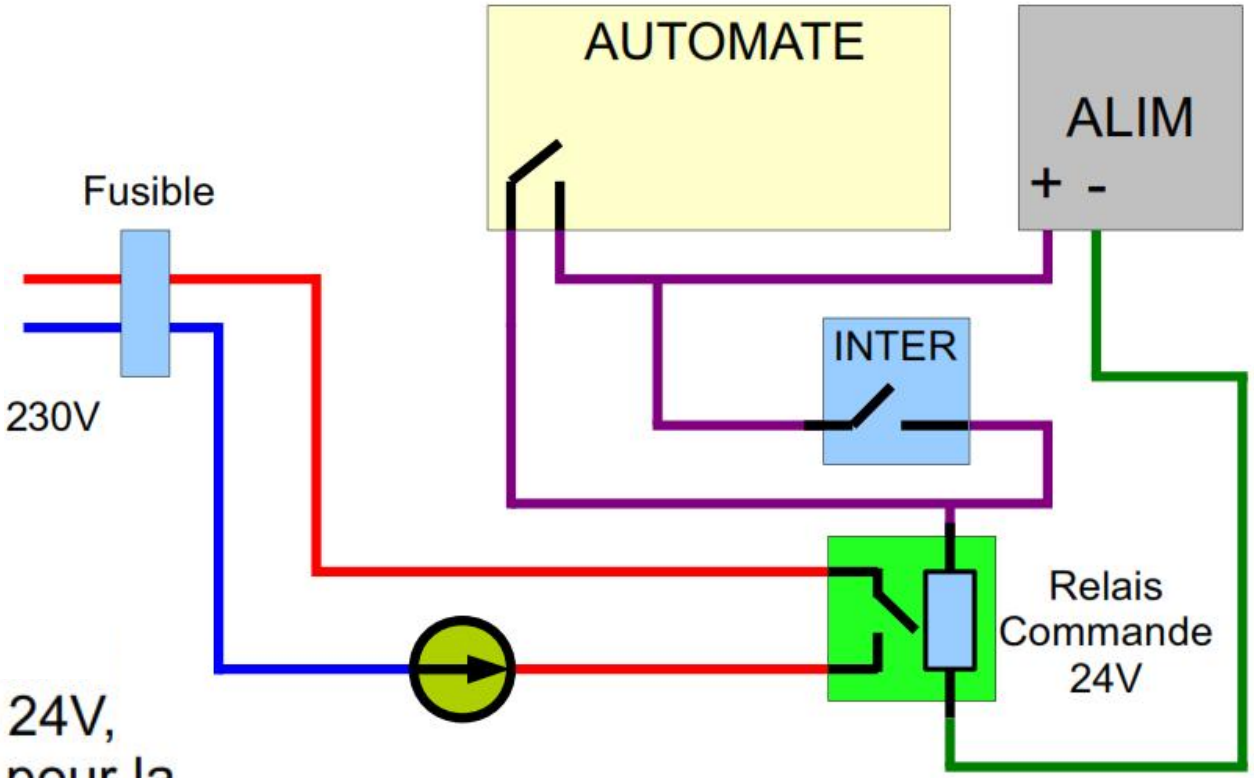
Idem avec un interrupteur pour forcer la commande

Exemples de câblage (E/S) automate



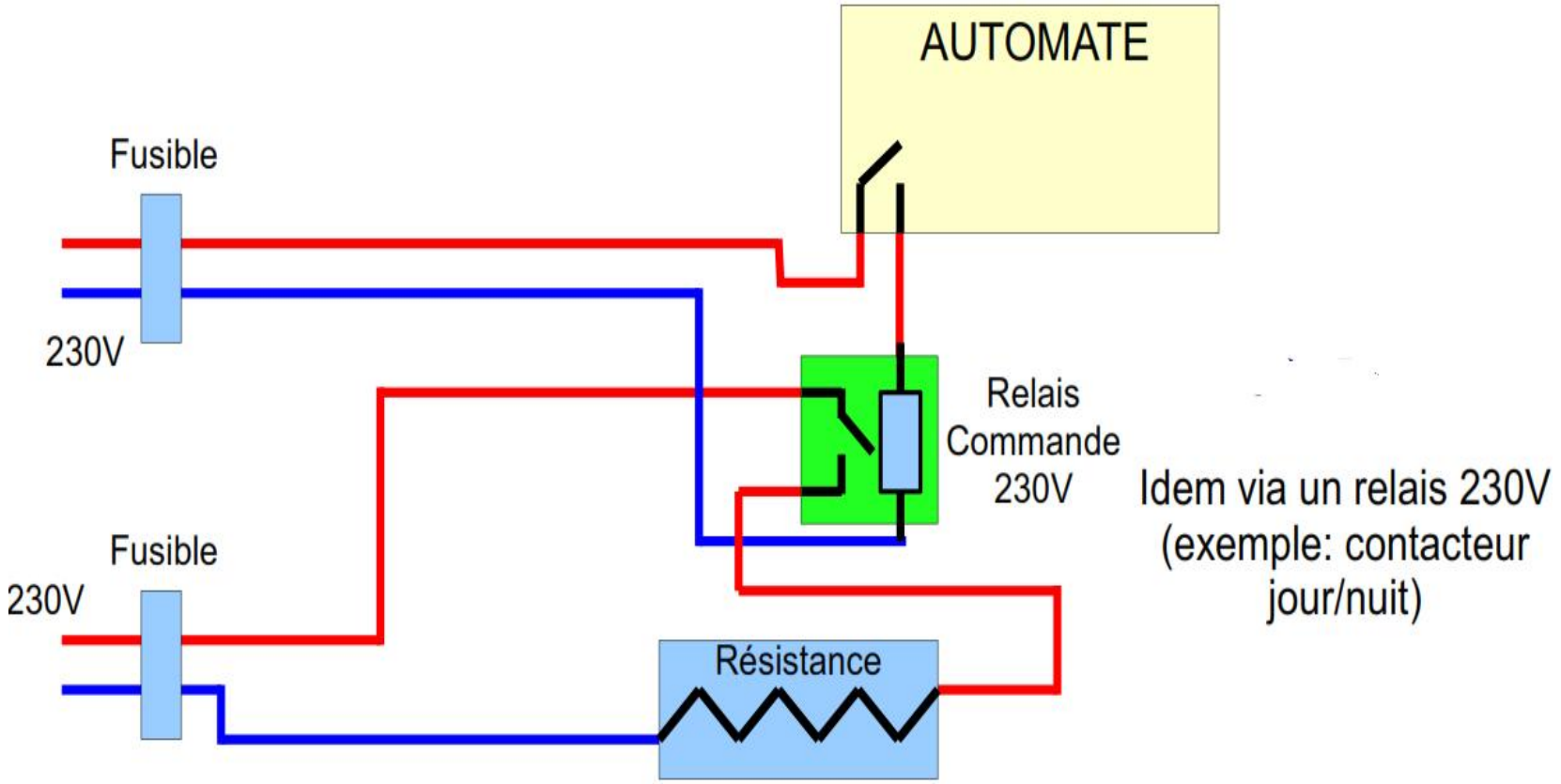
Idem avec un interrupteur supplémentaire pour forcer l'arrêt

Exemples de câblage (E/S) automate

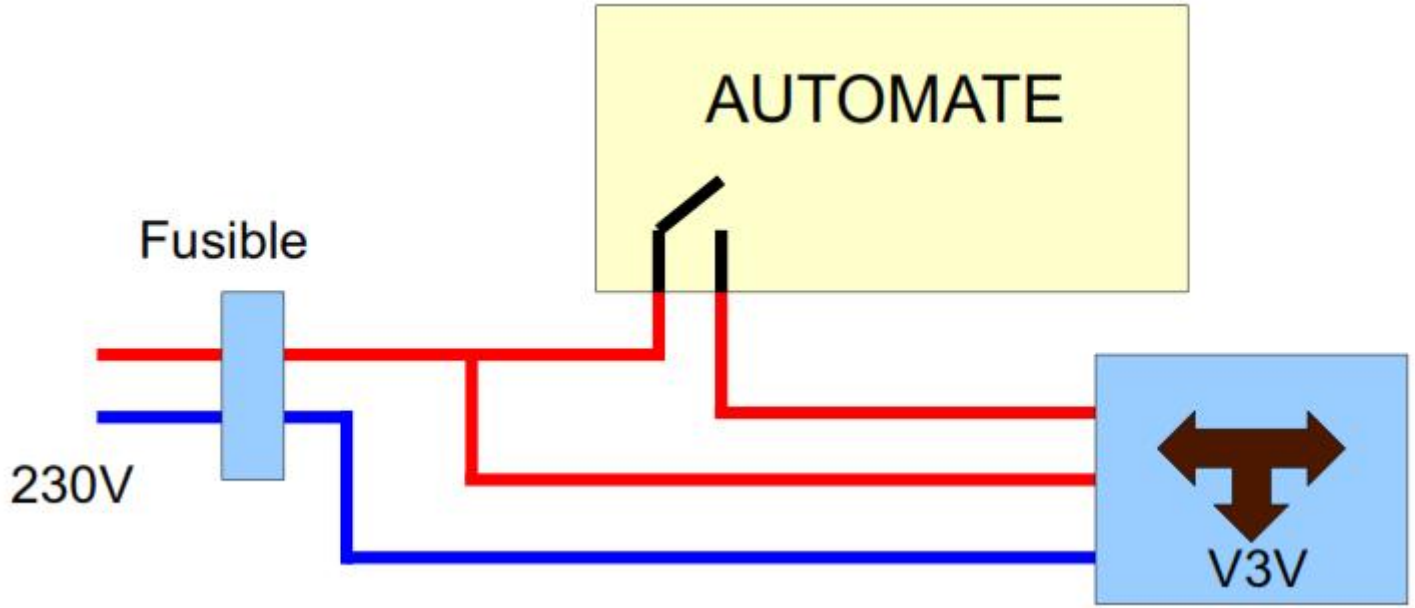


via un relais 24V, convient aussi pour la commande d'une résistance

Exemples de câblage (E/S) automate

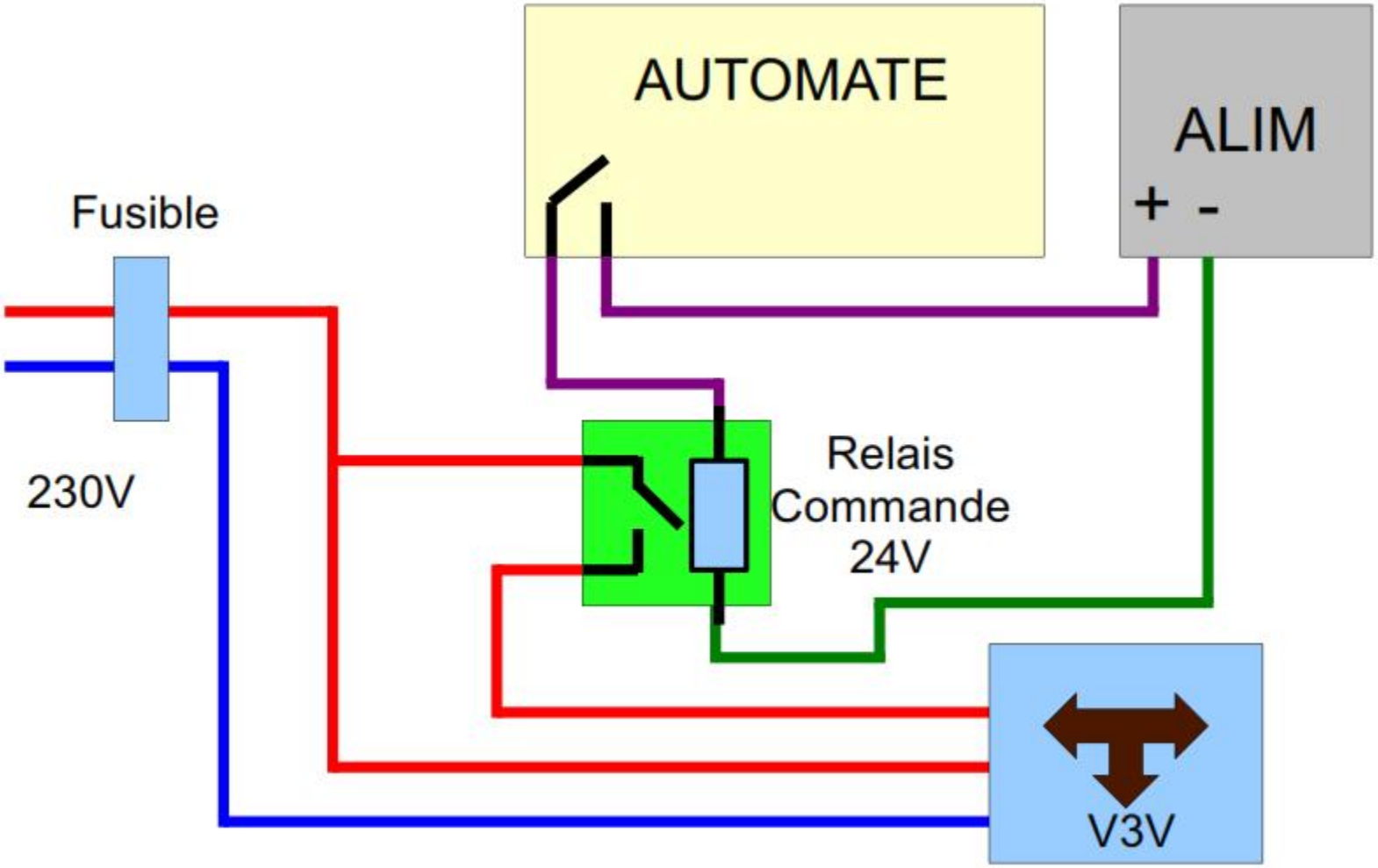


Exemples de câblage (E/S) automate

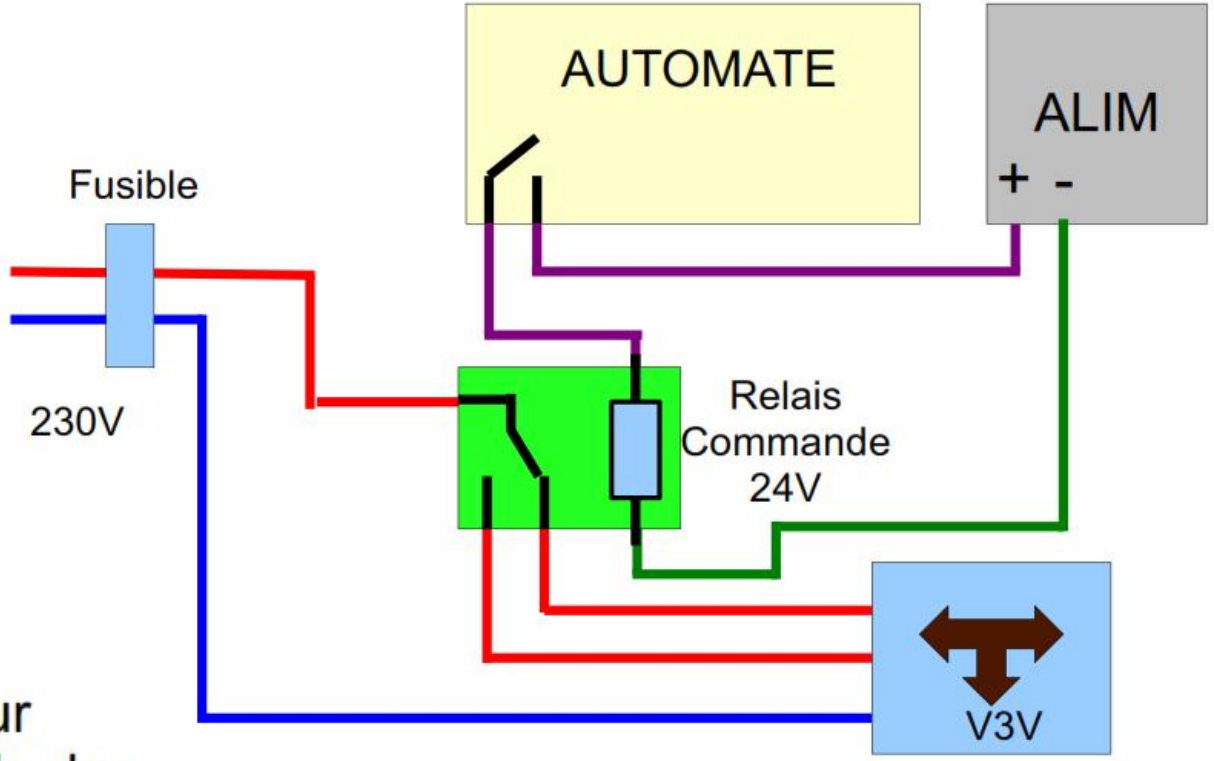


Câblage pour
vanne trois voies (V3V)
tout ou rien (TOR)
bistable
traditionnelle

Exemples de câblage (E/S) automate



Exemples de câblage (E/S) automate



Câblage pour
V3V TOR à 2 fils de
commande
via un relais à 2 contacts

Exemples de câblage (E/S) automate

- Pour les exemples précédents , un automate 24V a été choisi car ce modèle est très largement utilisé par les auto-installateurs.
- Il faut prendre des précautions concernant les sorties, certains API ont des sorties relais (en général avec un pouvoir de coupure de 5 ou 8A). On peut donc y connecter directement les appareils à commander excepté les appareils de puissance ou à forte induction, il est alors conseillé de protéger les relais de l'automate en passant par des relais intermédiaires.
- Par contre, d'autres API ont des sorties statiques (courant de coupure de 0,5A) dans ce cas il faudra utiliser des relais secondaires.

GRAFCET

zoheir_karaouzene@yahoo.fr

GRAFCET

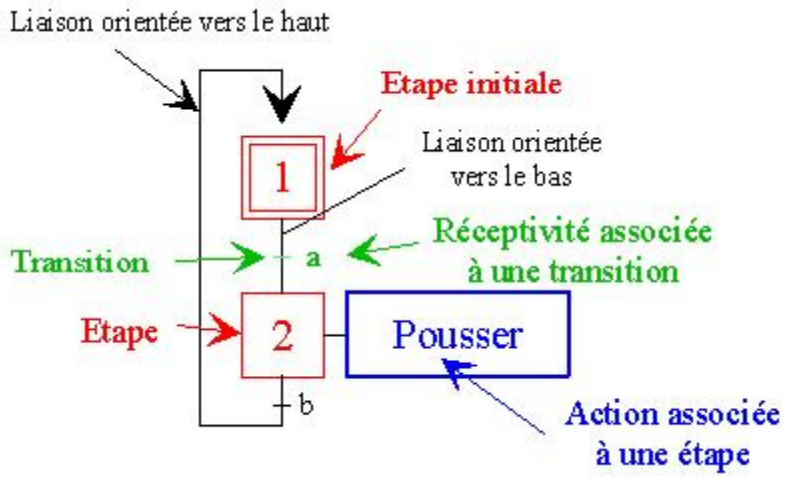
- Le GRAFCET est un outil graphique de description du comportement déterministe de la Partie Commande.
- Il établit une correspondance à caractère séquentiel et combinatoire entre :
  - les ENTREES, c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande,
  - et les SORTIES, transferts d'informations de la Partie Commande vers la Partie Opérative.

GRAFCET

- **Le grafcet est le résultat du travail bénévole d'une commission réunissant, l'AFCET (Association Française pour la Cybernétique Economique et Technique)**
- **Depuis 1988, le grafcet est un outil de description normalisé (Norme C.E.I. 848) qui fonctionne en logique séquentielle**
- **En 1985, SIEMENS (leader européen des automatismes) adopte le grafcet et le promeut en Allemagne**
- **En 1986 ALLEN & BRADLEY (leader mondial des automates programmables) adopte et développe le grafcet, y compris pour le marché américain.**

GRAFCET

- **L'acronyme grafcet signifie : GRAphe Fonctionnel de Commande Etape Transition.**
- **La description du fonctionnement d'un automatisme logique peut alors être représenté graphiquement par un ensemble :**
  - **d'ETAPES auxquelles sont associées des ACTIONS,**
  - **de TRANSITIONS auxquelles sont associées des RECEPTIVITES,**
  - **de LIAISONS (ou ARCS) ORIENTEES,**
- Exemple:

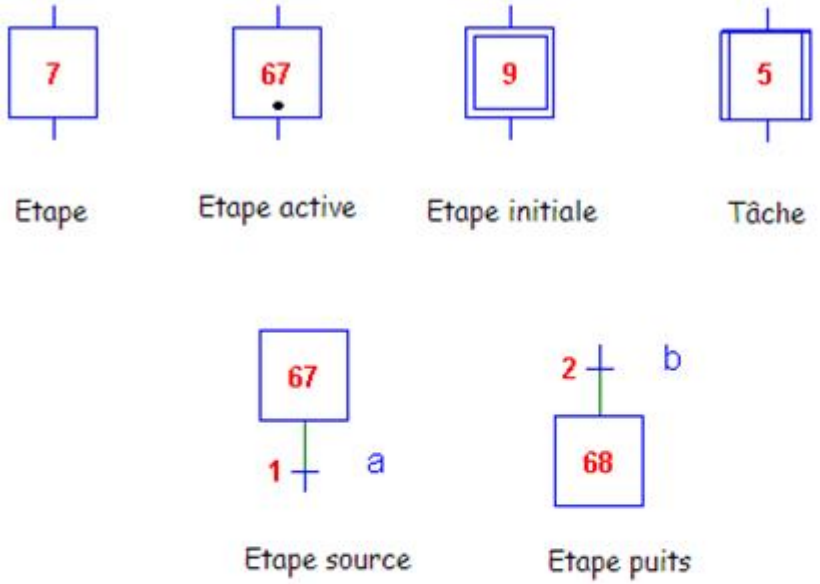


Règle d'établissement du GRAFCET

- **Chaque liaison orientée relie une étape à une transition ou une transition à une étape.**
- **Un grafcet se lit de haut en bas.**
- **Si cette syntaxe n'est pas scrupuleusement respectée, il y aura obligatoirement une erreur dans l'application.**
- **Une flèche peut compléter la liaison en indiquant le sens de lecture s'il y a un risque de confusion.**

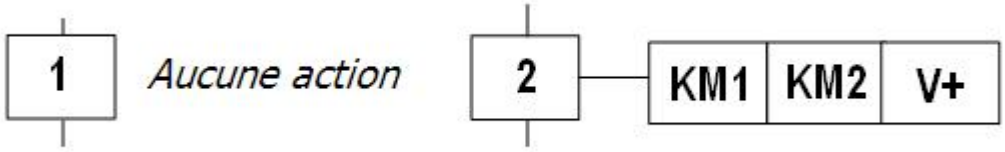
GRAFCET

- **Etapes:**
  - **Sur un grafcet on peut rencontrer différents types d'étapes :**



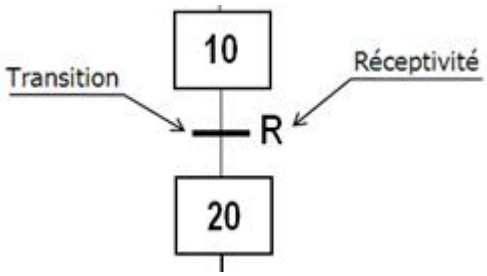
GRAFCET

- **Actions associées à l'étape**



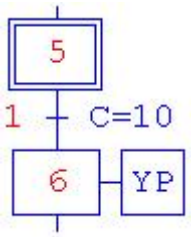
GRAFCET

- **Transition**

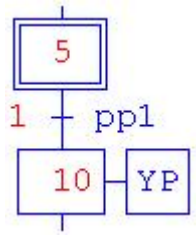


GRAFCET

- Exemples:



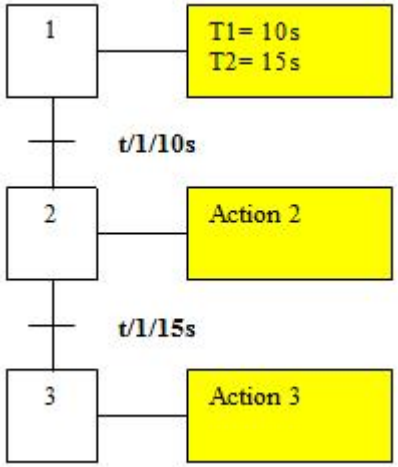
La réceptivité est vraie lorsque la valeur de la consigne du compteur C = 10.



La réceptivité est vraie lorsque le capteur pp1 est actif.

GRAFCET

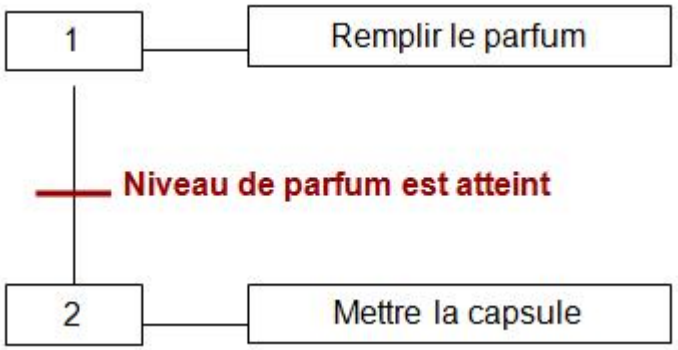
- Exemples



t/1/10s: passer à l'étape 2 si **10s sont écoulées** depuis la dernière ***activation de l'étape 1.***
t/1/15s: passer à l'étape 3 si **15s sont écoulées** depuis la dernière ***activation de l'étape 1***.

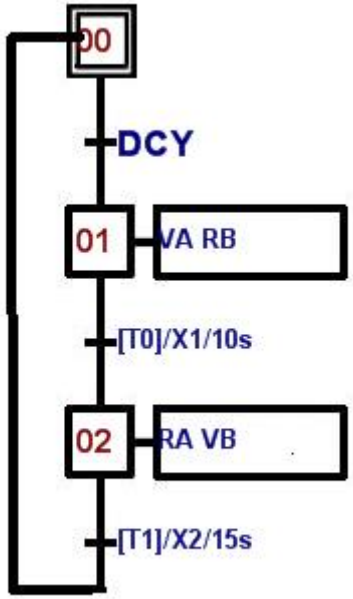
GRAFCET

- Exemple 1



GRAFCET

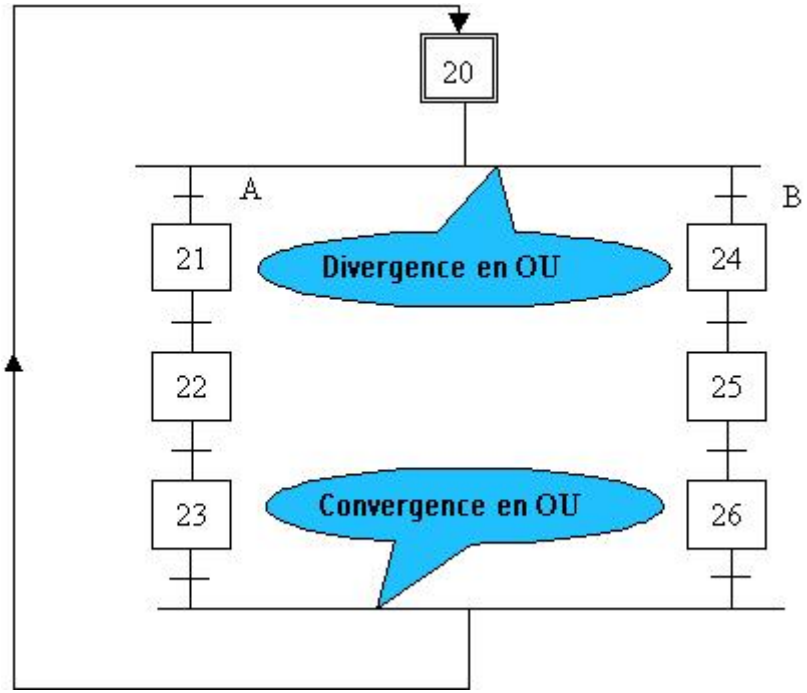
- Exemple 2



GRAFCET

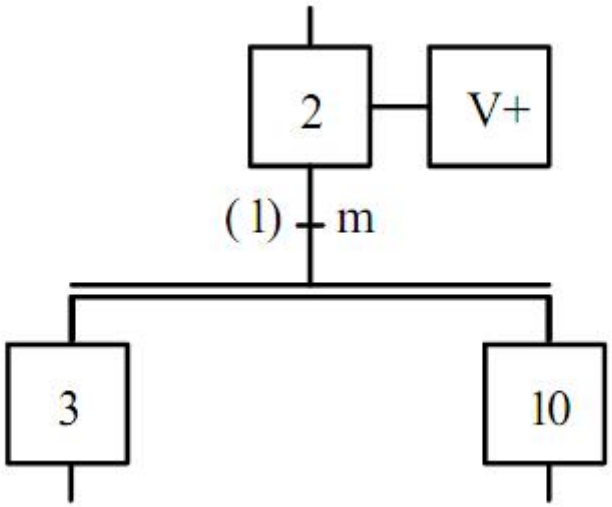
Divergence/convergence en OU

Divergence en OU : l'évolution du système vers une branche dépend des réceptivités A et B associées aux transitions.
Convergence en OU : après l' évolution dans une branche, il y a convergence vers une étape commune.
REMARQUES :
-A et B ne peuvent être vrais simultanément (conflit).
-Après une divergence en OU, on trouve une convergence en OU.
Le nombre de branches peut-être supérieur à 2
-La convergence de toutes les branches ne se fait pas obligatoirement au même endroit.



GRAFCET

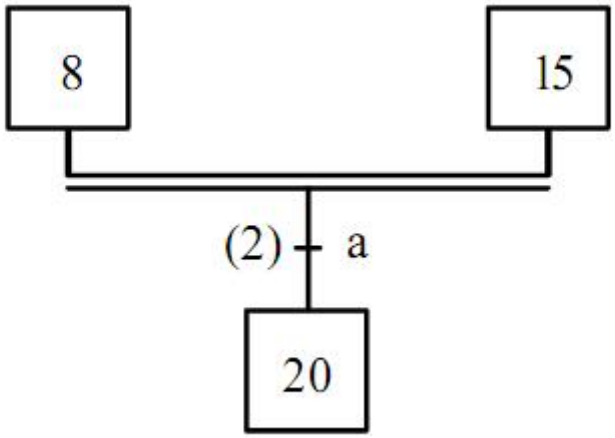
Divergence en ET



Si l'étape 2 est active et que m = 1 alors les étapes 3 et 10 sont activées tandis que l'étape 2 est désactivée.

GRAFCET

Convergence en ET

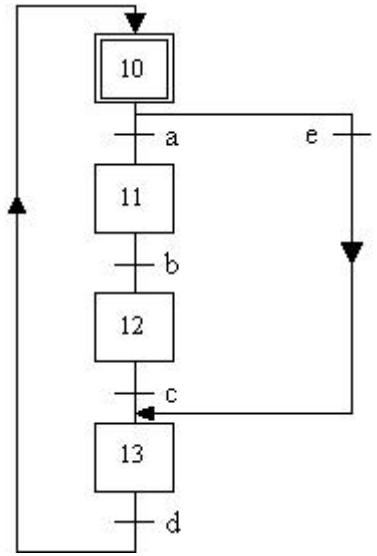


Si les étapes 8 et 15 sont actives et que a = 1 alors l'étape 20 est activée tandis que les étapes 8 et 15 sont désactivées.

GRAFCET

- **Saut en avant (saut de phase)**

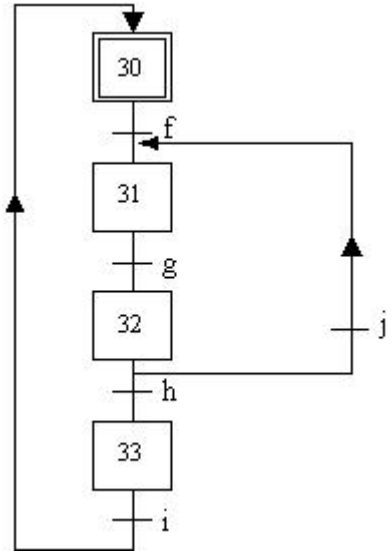
Le saut en avant permet de sauter une ou plusieurs étapes lorsque les actions à réaliser deviennent inutiles.



GRAFCET

- **Saut en arrière (reprise de phase)**

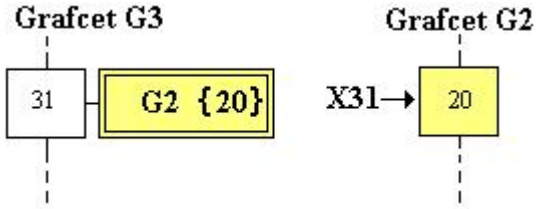
Le saut en arrière permet
de reprendre une séquence
lorsque les actions à réaliser sont répétitives.



GRAFCET

- **FORÇAGE**

A l'étape 31 du Grafcet G3,
il y a **forçage** du Grafcet G2 à l'étape 20.

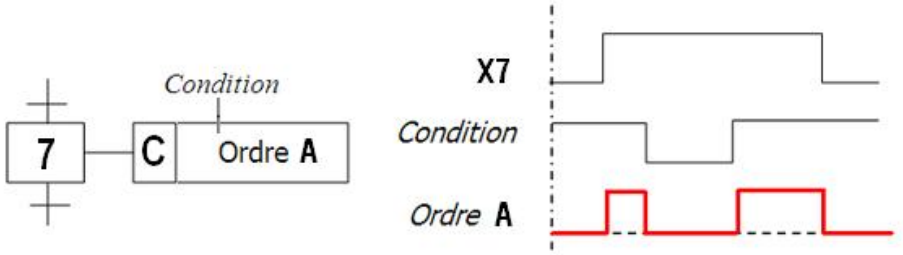


GRAFCET

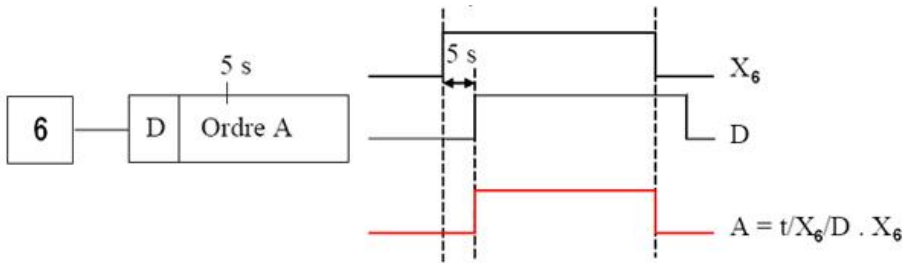
Actions conditionnelles:

- Une action **conditionnelle** n'est exécutée que si l'étape associée est active et si la condition associée est vraie. Elles peuvent être décomposées en 3 cas particuliers:

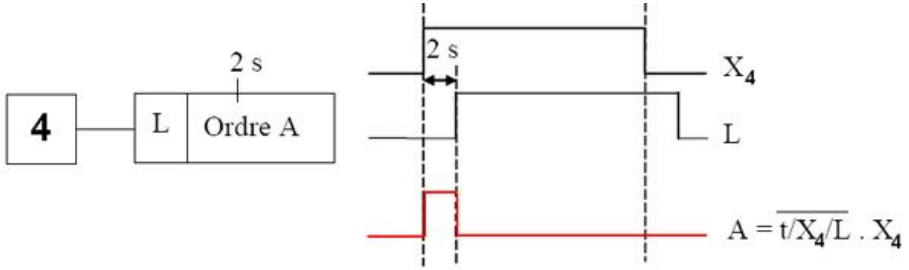
-Action conditionnelle simple : *Type C*



-Action retardée : Type D (delay)

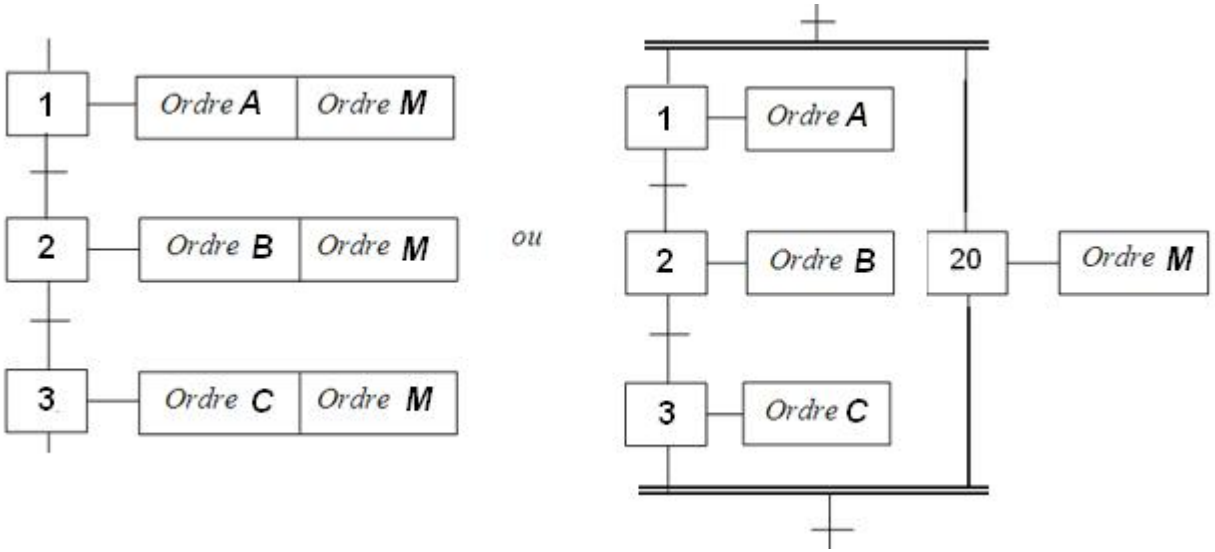


- Action de durée limitée: Type L (limited)



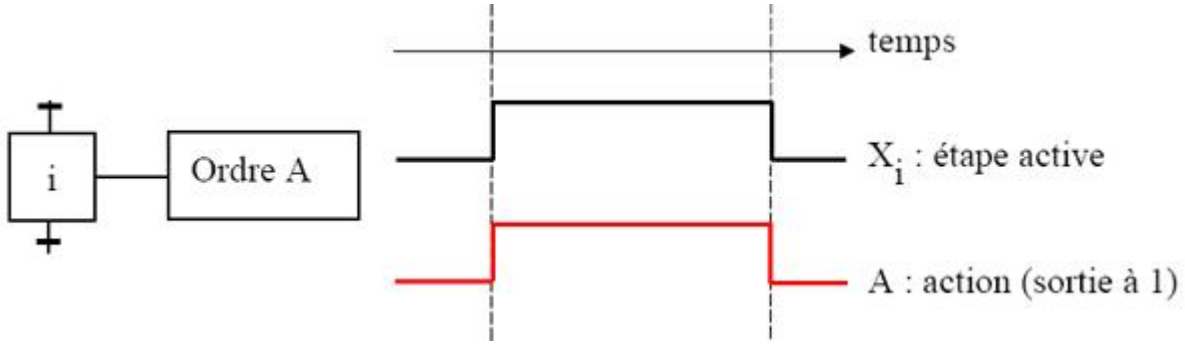
GRAFCET

- **Action maintenue sur plusieurs étapes:**



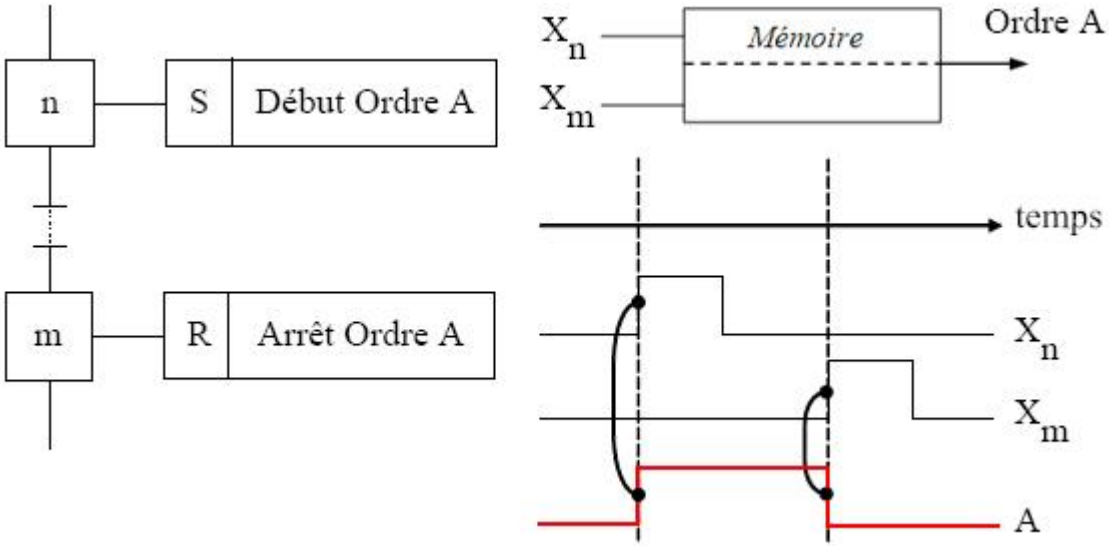
GRAFCET

- **Classification des actions associées aux étapes**
  - **Actions continues**



GRAFCET

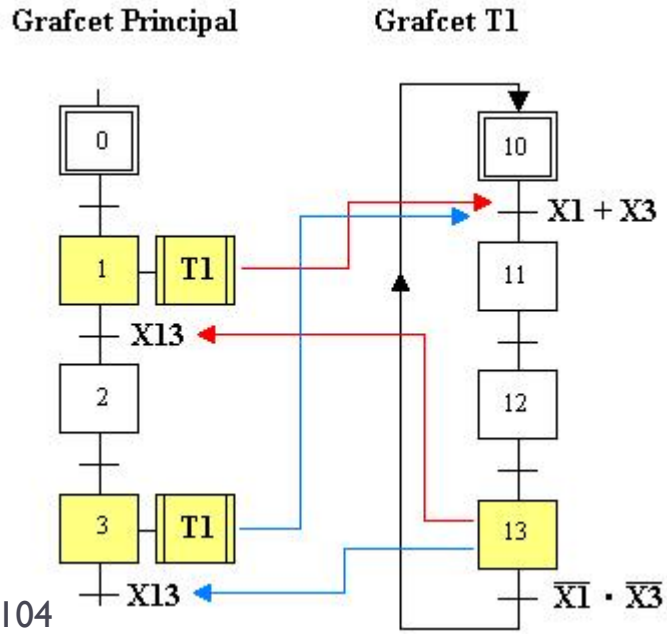
- **Action mémorisée :**

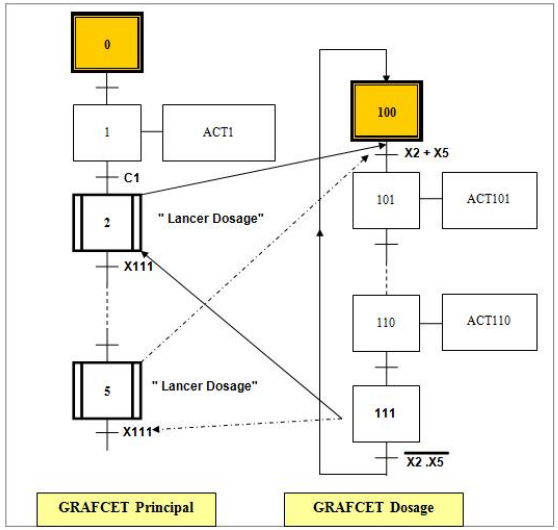


GRAFCET

Taches – Sous–programme

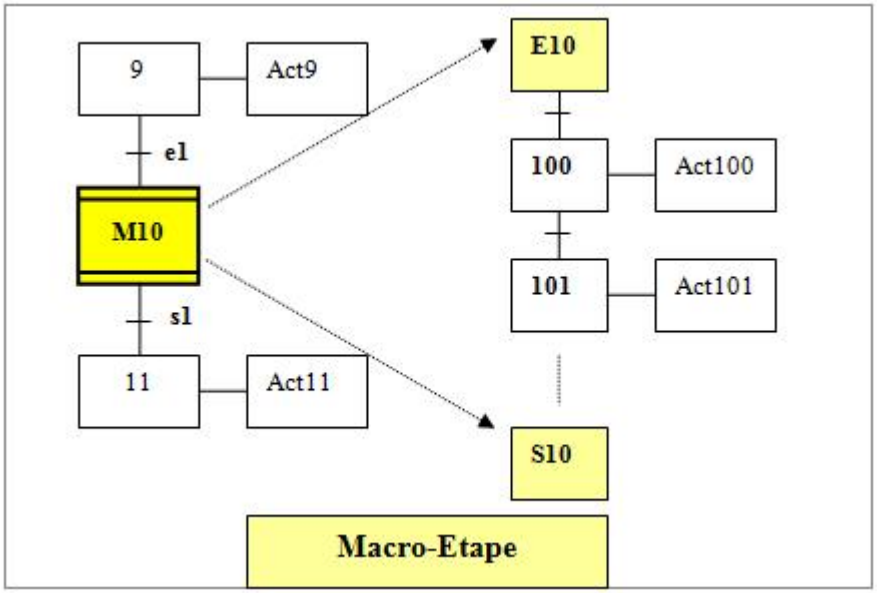
- Dans les automatismes séquentiels, il est fréquent de rencontrer des séquences répétitives dans le même cycle. Une séquence répétitive peut être représentée par un sous-grafcet ou grafcet sous-programme. Cette notion de est empruntée au langage informatique.
- Un grafcet sous-programme est écrit sous la forme d'un grafcet indépendant, connecté au grafcet principal.





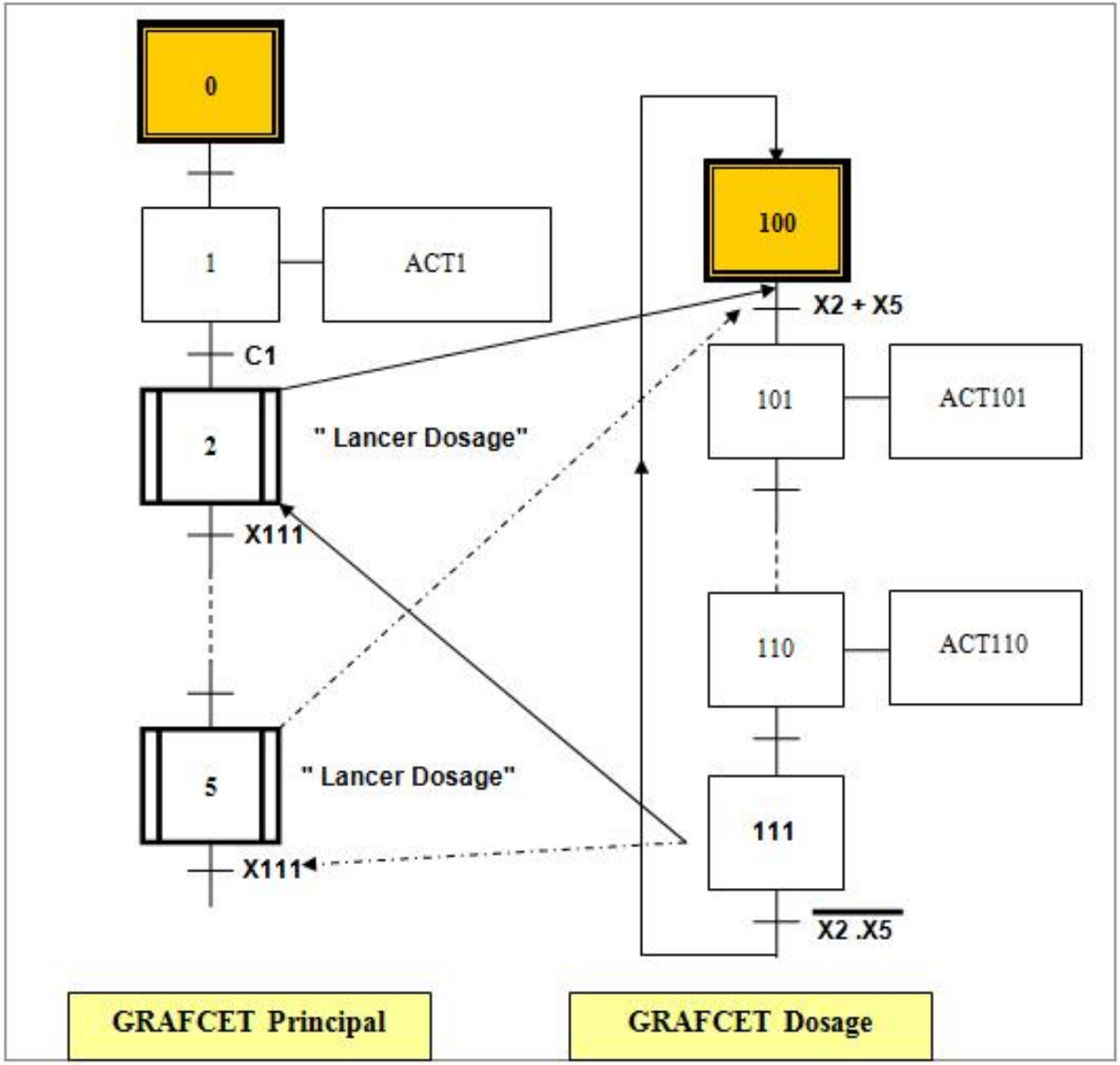
GRAFCET

- Macro-étape



Une macro-étape est une représentation unique d'un ensemble d'étapes et de transitions appelée *expansion* de la macro-étape.

GRAFCET


0
1
ACT1
C1
2
" Lancer Dosage"
X111
5
" Lancer Dosage"
X111
100
X2 + X5
101
ACT101
110
ACT110
111
X2 .X5
GRAFCET Principal
GRAFCET Dosage

GRAFCET

- Les équations logiques d'étapes :

- pour les étapes initiales :

$$X_i = CAX_i + X_i . \overline{CDX_i} + Init$$

- pour les étapes non initiales :

$$X_i = (CAX_i + X_i . \overline{CDX_i}) . Init$$

La condition d'activation de l'étape i (CAXi) :

$$CAX_i = X_{i-1} \cdot t_{i-1}$$

La condition de désactivation de l'étape i (CDXi) :

$$CDX_i = X_{i+1}$$