



**UNIVERSITÉ  
DE LORRAINE**



**Nancy-Brabois**  
Génie Électrique et Informatique Industrielle

## **Licence Professionnelle EGC-ITEB**

**Energie - Génie Climatique:**

**Intelligence Technique et Energétique du Bâtiment**

**Module EC2b**

**Programmation et Automatisation  
Bâtiment**

par Pr. **Eddy BAJIC**

IUT Nancy Brabois  
Génie Electrique & Informatique Industrielle  
Université de Lorraine

<b>1</b>	<b>AUTOMATISMES INDUSTRIELS.....</b>	<b>3</b>
<b>2</b>	<b>AUTOMATISATION DU BATIMENT (<i>BUILDING AUTOMATION</i>) .....</b>	<b>5</b>
<b>3</b>	<b>EXEMPLES D'ARCHITECTURES DE GTB .....</b>	<b>7</b>
<b>4</b>	<b>DESCRIPTION D'UN SYSTEME AUTOMATISE .....</b>	<b>9</b>
4.1	Partie Opérative - Partie Commande - Interface Homme Machine .....	9
4.2	Un carrefour vu comme un système automatisé .....	10
4.3	Cycle de Vie d'un Système Automatisé .....	11
<b>5</b>	<b>DIFFERENTS TYPES DE PARTIE COMMANDE.....</b>	<b>12</b>
5.1	Comparatif des différentes techniques .....	12
5.2	Système programmable Vs Système Câblé.....	13
5.3	Exemples de API/Contrôleur utilise en GTB/GTC .....	13
5.3.1	<i>Schneider TAC : Gamme TAC Xentra 100-900 (www.tac.schneider-electric.fr/)</i> .....	13
5.3.2	<i>Contrôleur RIO de NAPAC</i> .....	14
5.3.3	<i>Siemens : Contrôleur modulaire RMH760</i> .....	16
5.3.4	<i>Honeywell : Contrôleur HVAC</i> .....	16
5.3.5	<i>API contrôleurs génériques</i> .....	16
<b>6</b>	<b>STRUCTURE D'UN API.....</b>	<b>17</b>
<b>7</b>	<b>FONCTIONNEMENT D'UN API / CONTROLEUR.....</b>	<b>17</b>
7.1	Cycle API.....	18
7.2	Fonctionnement Temps Réel « lâche » .....	19
7.3	Affectation d'une sortie physique sur un automate.....	20
7.4	Détection d'un front montant sur un automate .....	20
7.5	Codage en Langage à contact de la détection de front montant.....	21
7.6	Variables Mémoire API.....	21
<b>8</b>	<b>LANGAGES DE PROGRAMMATION DES API.....</b>	<b>22</b>
8.1	Langage à contacts (LD : Ladder diagram) : .....	22
8.2	Langage littéral structuré (ST : Structured Text).....	24
8.3	Blocs Fonctionnels (FBD : Function Bloc Diagram) .....	24
8.4	SFC (Sequential Function Chart) .....	24
8.5	Liste d'instructions (IL : Instruction list).....	25
<b>9</b>	<b>PROGRAMME EN LOGIQUE COMBINATOIRE OU PROGRAMMES EN LOGIQUE SEQUENTIELLE.</b>	<b>26</b>
9.1	Exercice de Programme API : Commande de gestion d'un Réservoir.....	26
9.2	Exercice de Programme API : Comptage Parking.....	27

## 1 AUTOMATISMES INDUSTRIELS

Automatisation croissante des procédés industriels :

Qualité ↗, Coût ↘, Délai ↘, Productivité ↗, Disponibilité ↗

Un système automatisé est la coopération de :

- ❑ Machines, Processus Physique
- ❑ Eléments d'Instrumentation et Actionnement
- ❑ Système de Conduite et surveillance
- ❑ Opérateurs Humains

Dans le but de :

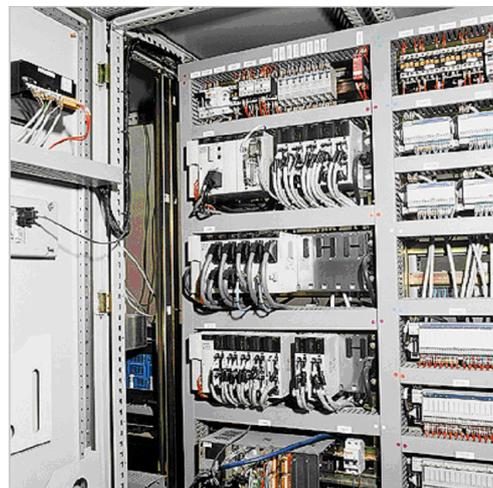
- ⇒ Automatiser le fonctionnement
- ⇒ Assurer la Sécurité
- ⇒ Améliorer la Maintenance
- ⇒ Faciliter les Interventions des opérateurs

### Usine de fabrication de pâtes alimentaires

Système Automatisé



élément vibrant commandé



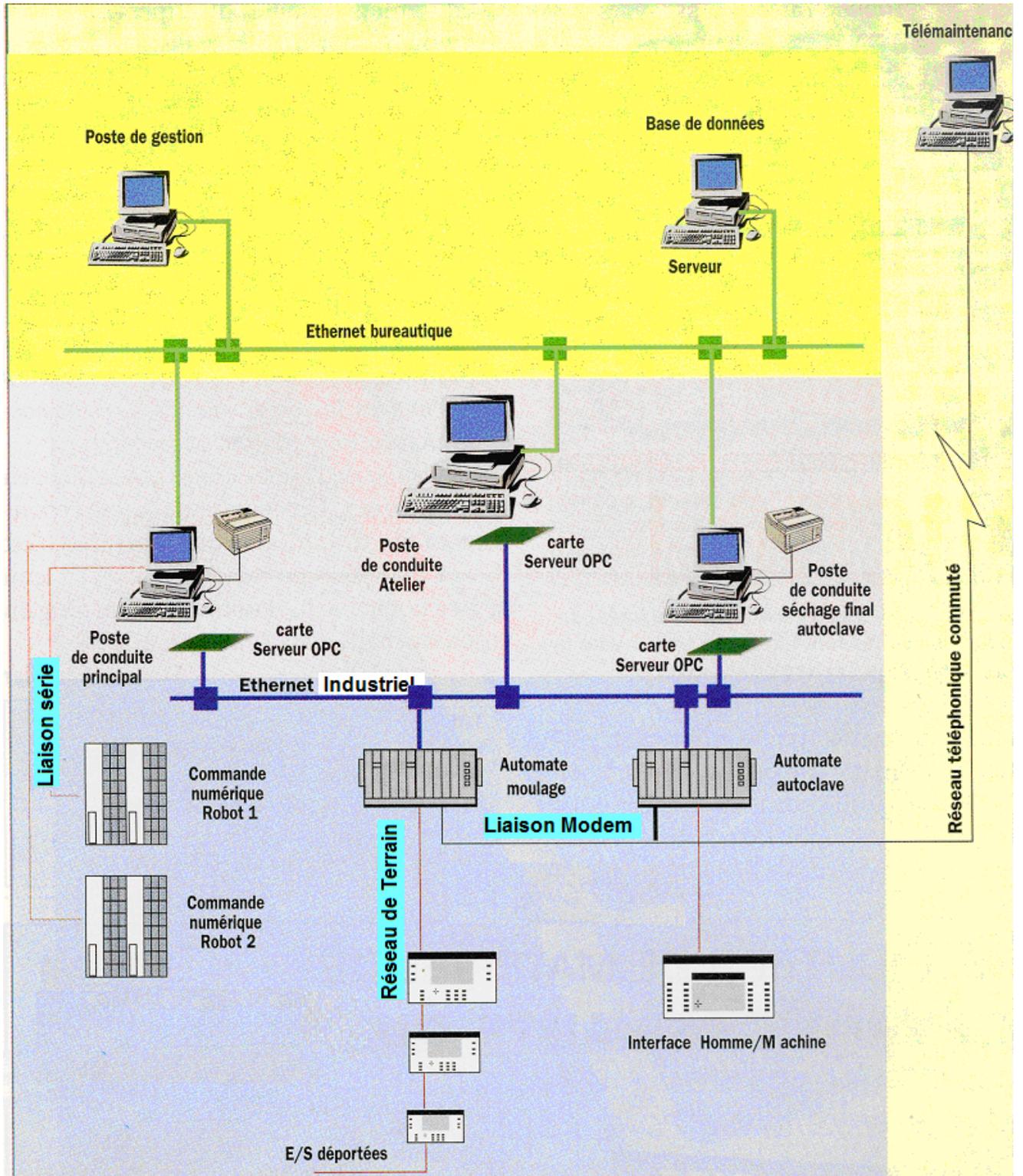


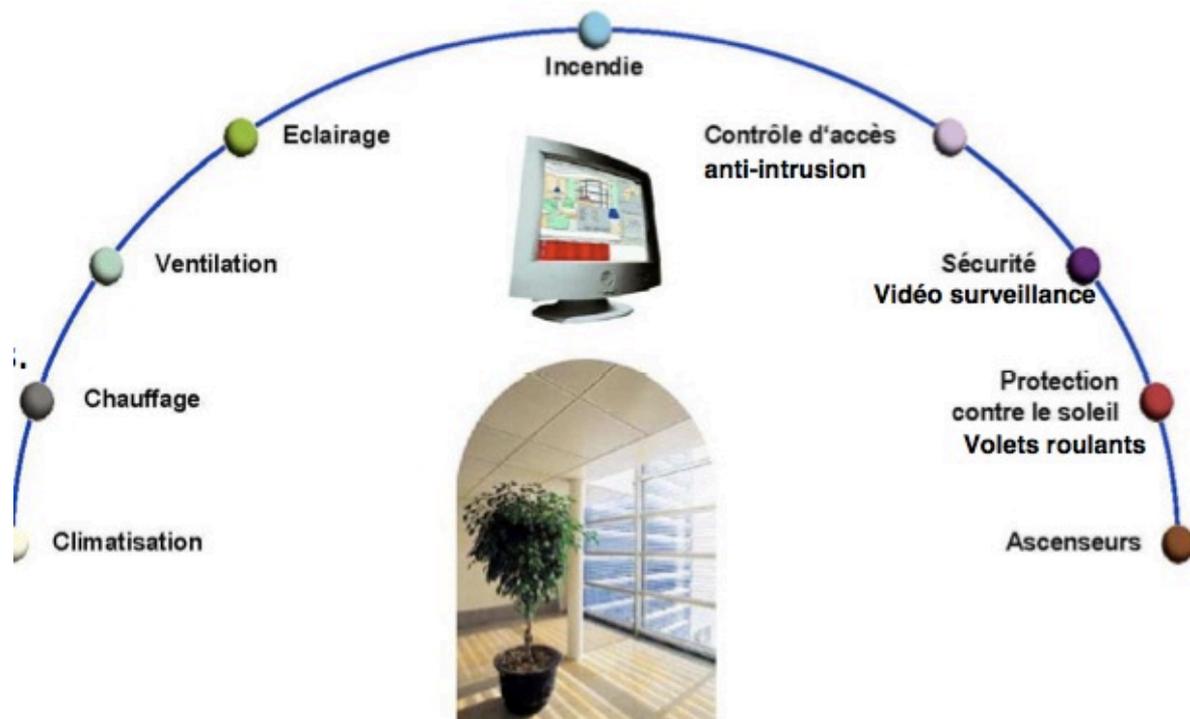
Figure 1 : Architecture d'automatisme distribué

## 2 AUTOMATISATION DU BATIMENT (*BUILDING AUTOMATION*)

« Un bâtiment est un outil de Production : Confortable , Economique , Sûr » (Schneider TAC)

L'automatisation du Bâtiment regroupe les techniques d'automatisme, d'informatique, de télécommunication qui permettent entre autres :

- de gérer intelligemment toutes les fonctionnalités de l'habitat : chauffage, ventilation, électricité, climatisation, ...
- de contrôler l'espace sécurité globalisant la sécurité technique, humaine, effraction, vol (surveillance), incendie
- de communiquer dans l'espace interne et externe de l'habitat.
- de surveiller le bâtiment à distance



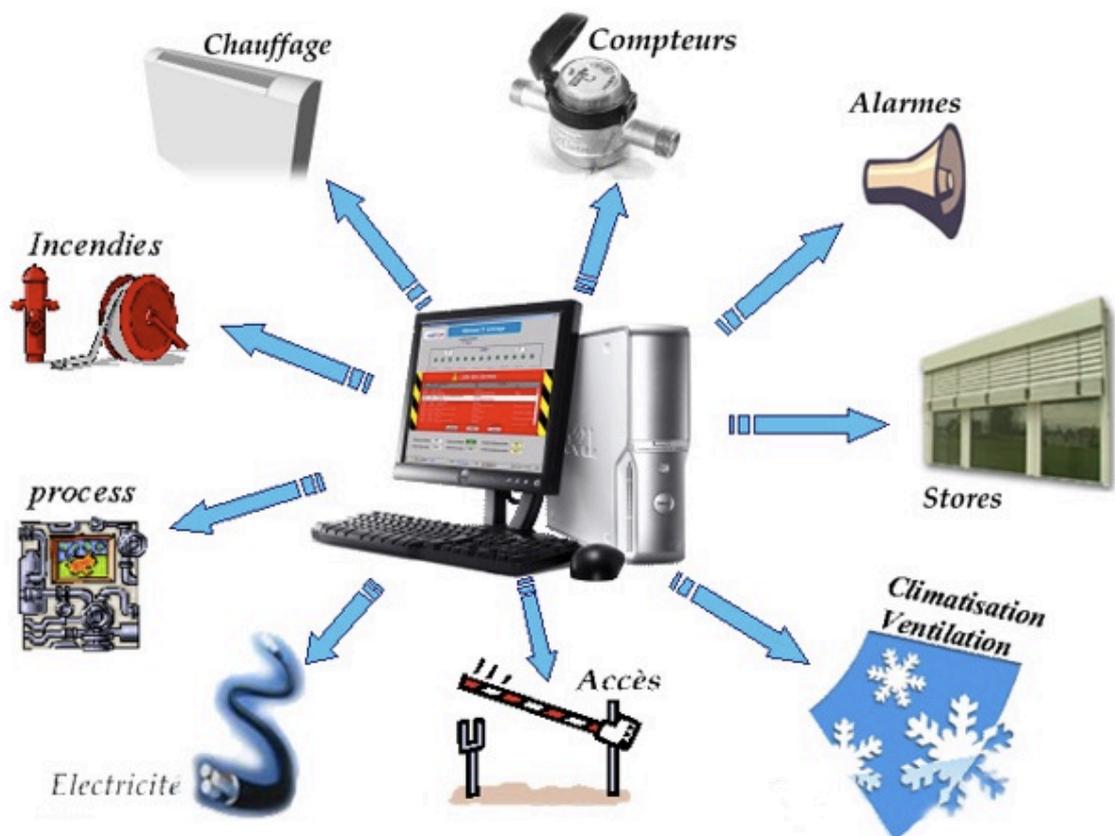
La **Gestion Technique du Bâtiment** dite GTB (ou *BMS en anglais, Building Management System*) est l'ensemble des systèmes de traitement des informations provenant d'un même site et permet de "connecter" les équipements techniques du bâtiment de confort, de chauffage, de sécurité, ..., dans les objectifs de:

→ **SECURISER - MAITRISER - ECONOMISER – GERER**

Gérer les Services Usagers ↗, Economie Energie ↗,  
Confort & Bien-être ↗, Sécurité ↗

**La GTB est un concept communicant numérique et informatique, permettant:**

- La surveillance des installations
- Réduire les dépenses énergétiques
- Maintenir la température et le confort
- Fiabiliser les installations - fonctionnement sans discontinuité
- Surveiller 24h/24 7j/7
- Réduire des déplacements et intervenir très rapidement à distance
- Assurer une gestion énergétique optimum du bâtiment



**La GTB est un concept communicant numérique et informatique dans le bâtiment**

**Building Automation** terme anglais pour GTB

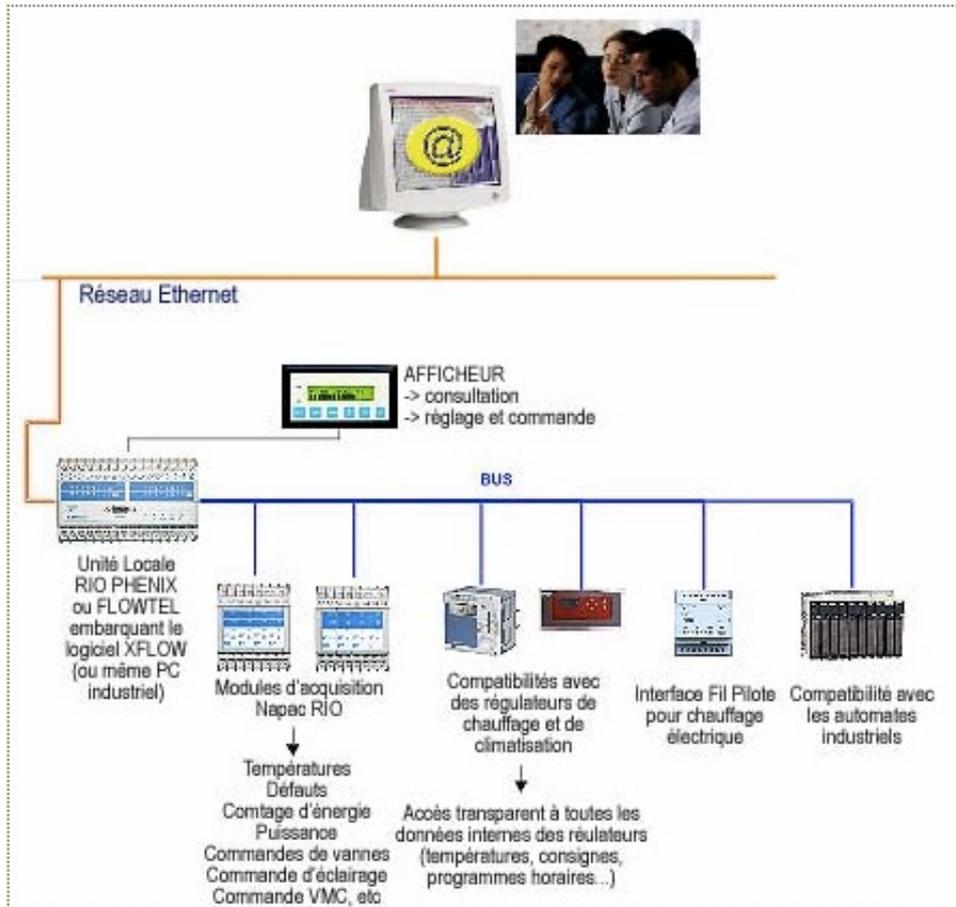
**Domotique** est le terme utilisé dans le domaine de l'habitat particulier.

**HVAC** Heating , Ventilation and Air Conditioning

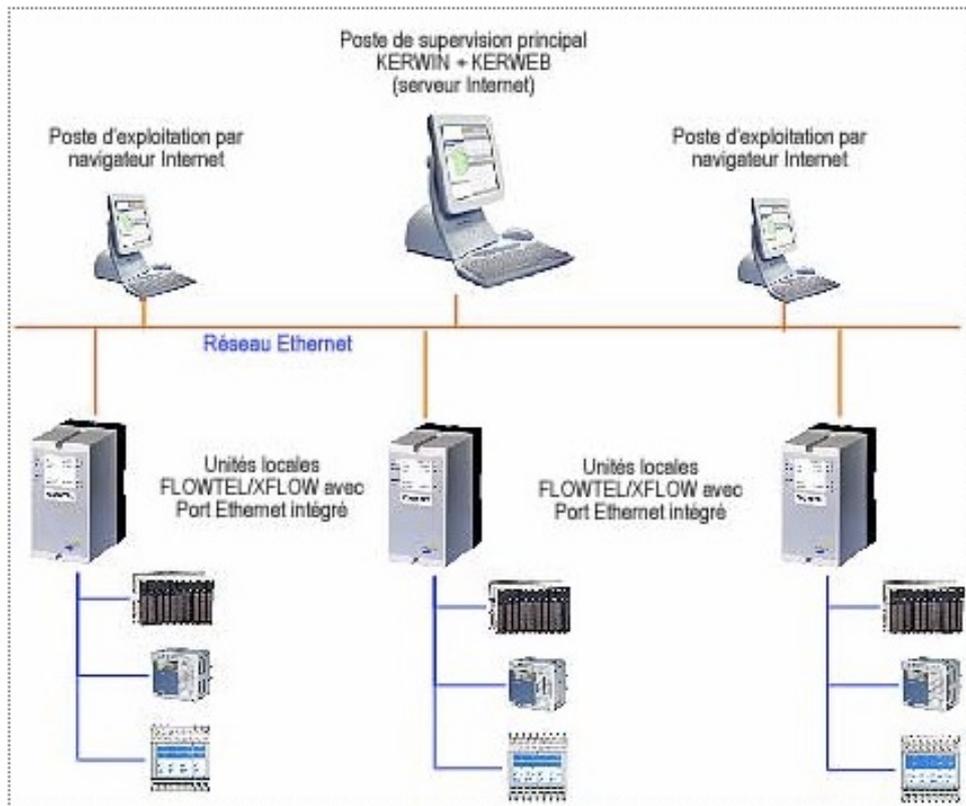
**CVC** : Climatisation, Ventilation, Chauffage

**GTC** : Gestion Technique Centralisée (limitée à un domaine : élec ou clim etc ...)

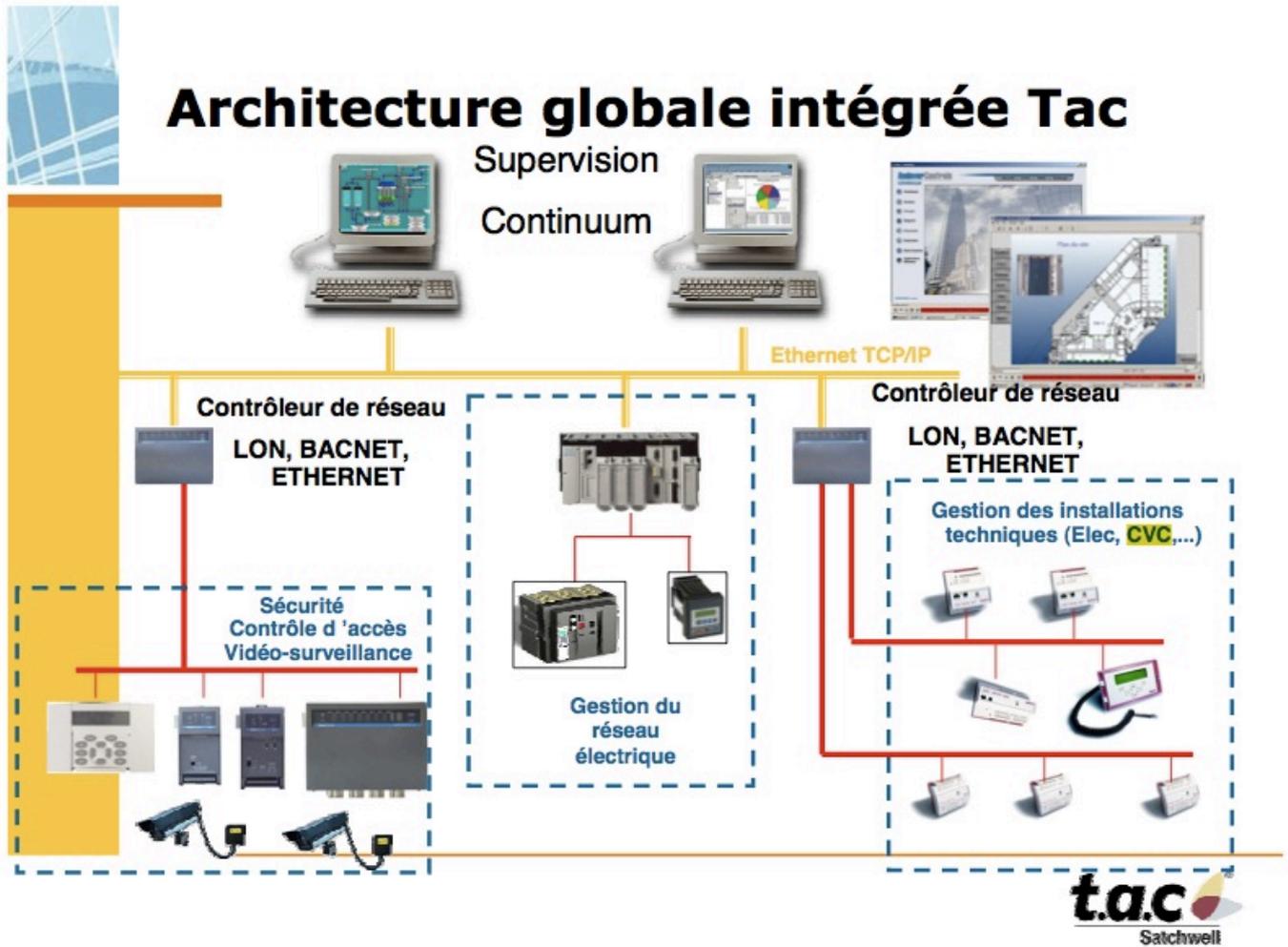
### 3 EXEMPLES D'ARCHITECTURES DE GTB



**Exemple d'architecture GTB simple**



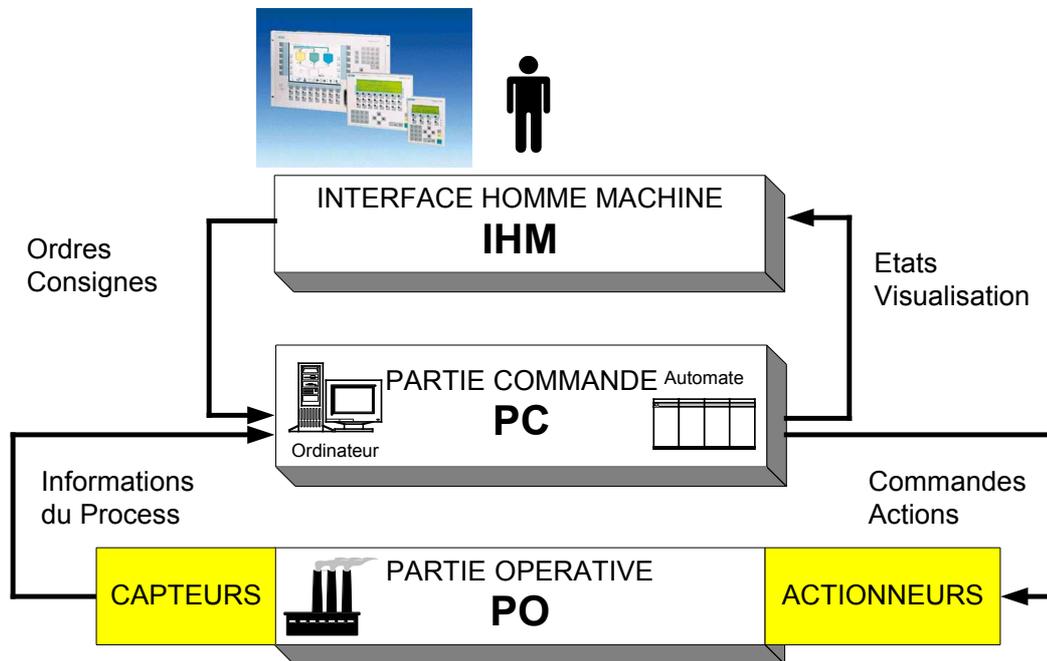
**Exemple d'architecture GTB multipostes**



## 4 DESCRIPTION D'UN SYSTEME AUTOMATISE

### 4.1 Partie Opérative - Partie Commande - Interface Homme Machine

Un système automatisé se décompose en trois parties indépendantes :



-La **Partie Opérative PO**, est le processus physique à automatiser

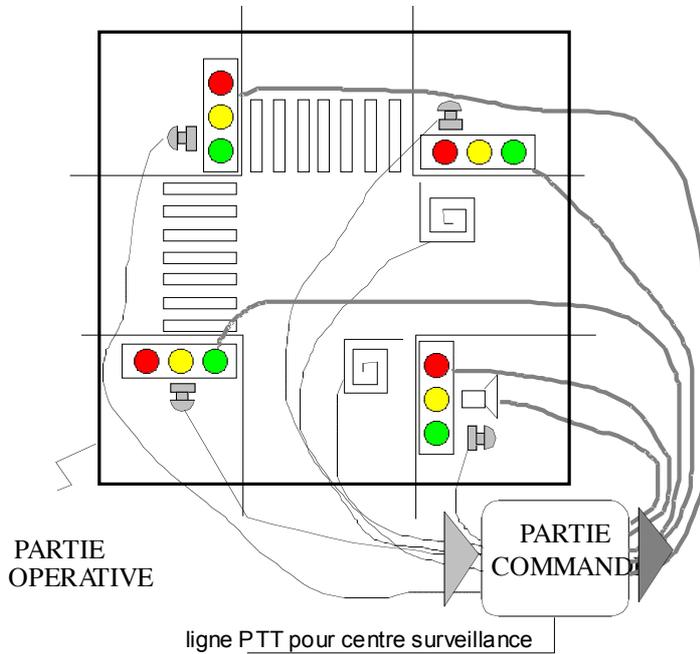
-La **Partie Commande PC**, est l'automatisme qui, en fonction des "ENTREES" (informations externes venant de la partie opérative, consignes extérieures, etc ...) élabore en "SORTIE" des ordres externes destinés à la partie opérative ou à des éléments extérieurs.

-L'**Interface Homme Machine IHM**, est le système de présentation des informations du système automatisé permettant la visualisation du procédé, la prise en compte des consignes et ordres des opérateurs humains sur le procédé.

## 4.2 Un carrefour vu comme un système automatisé

Analysons l'exemple d'un carrefour routier qui représente un système automatisé dans lequel on décompose la PO et la PC.

La majorité des carrefours urbains sont gérés comme des systèmes automatisés pilotés depuis un central de surveillance urbain.



### Entrées (capteurs)

- BP appel piéton
- Boucle Magnétique Sol détection voiture
- ligne PTT pour centre surveillance

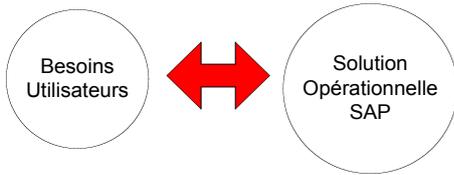
### Sorties (actionneurs)

- Feux tricolores
- Haut Parleur Signal Sonore Malentendants
- Ligne PTT pour centre surveillance

L'analyse et la commande des systèmes automatisés nécessite la **définition préliminaire d'un cahier des charges (CdC)** qui doit conduire à une réalisation sûre et sans équivoque, de l'automatisme.

### 4.3 Cycle de Vie d'un Système Automatisé

Le développement d'un Système Automatisé est un processus long et complexe.



Il est nécessaire de découper en tâches, le processus complexe qui transforme les besoins d'un utilisateur en une solution implantée et opérationnelle.

Plusieurs approches industrielles existent dont l'approche "Cycle de Vie en V".

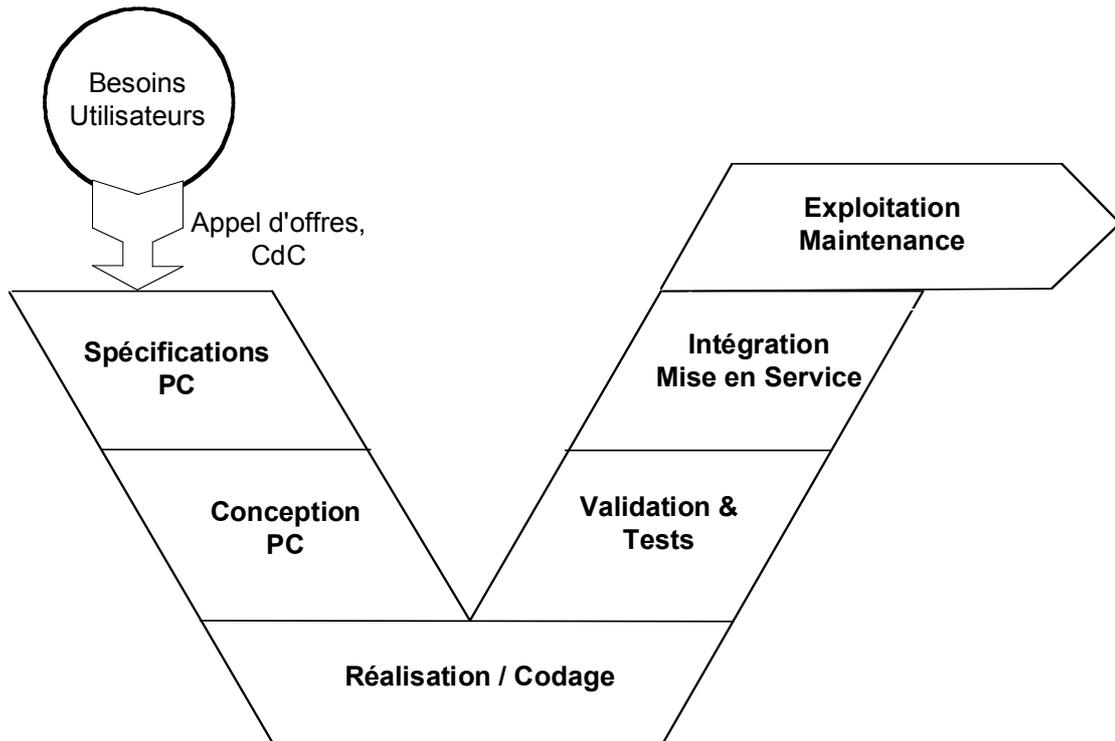
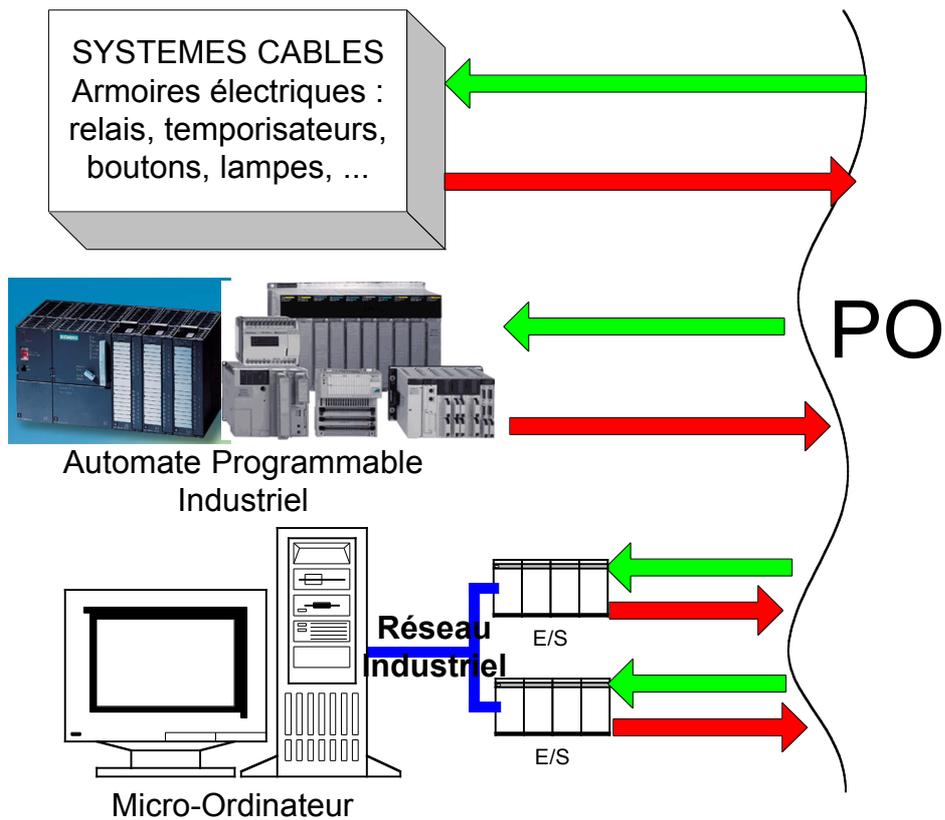


Figure 2 : Cycle en vie d'un automatisme industriel

L'approche « cycle en V » d'un SA, permet un enrichissement progressif durant son développement.

La notion de cycle s'applique de façon identique pour PC et PO, dont les réalisations sont souvent menées en parallèle.

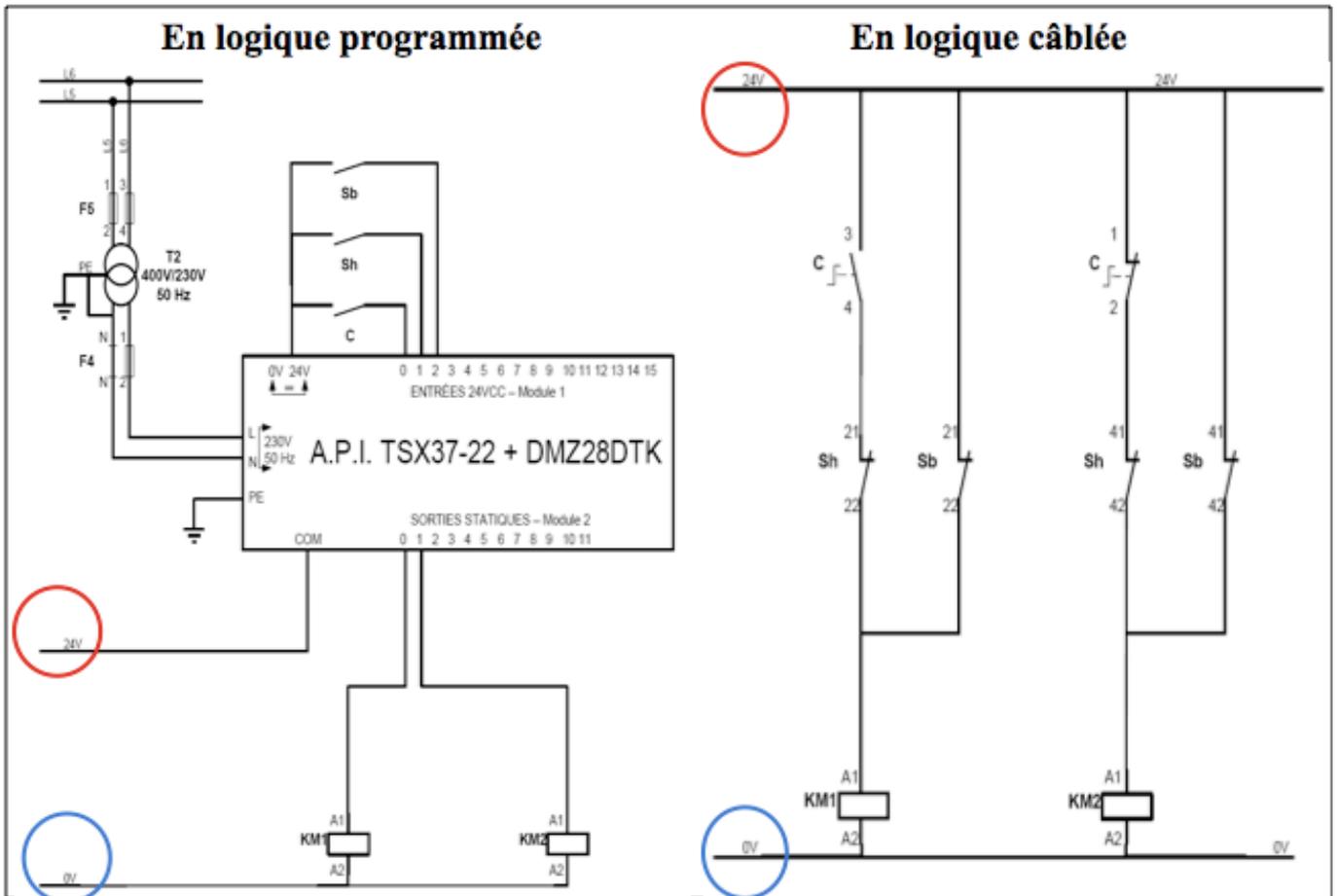
## 5 DIFFERENTS TYPES DE PARTIE COMMANDE



### 5.1 Comparatif des différentes technologies

Systèmes Câblés	API / Contrôleur	Micro-Ordinateur
Système Figé	Evolutif	Idem Automate/Contrôleur
Maintenance difficile	Autodiagnostic Maintenance Intégrée et programmée	Idem Automate/Contrôleur
Spécialisé	Flexible Programmable	Idem Automate/Contrôleur
Coût et temps de câblage importants	Coût matériel élevé	Très grande Versatilité Coût Revient bas
Fonctionnement Indépendant	Fonctionnement en réseau	Idem Automate/Contrôleur

## 5.2 Système programmable Vs Système Câblé



## 5.3 Exemples de API/Contrôleur utilise en GTB/GTC

### 5.3.1 Schneider TAC : Gamme TAC Xentra 100-900 ([www.tac.schneider-electric.fr/](http://www.tac.schneider-electric.fr/))



- Design compact et modulaire
- Programmable
- Extensible avec 10 modules E/S
- Adapté aux systèmes décentralisés
- Conçu pour des applications **CVC** et autres (Elec. Etc..)
- Réduction des coûts d'installation
- Certifié LonMark® pour une intégration facile avec d'autres équipements LonMark®

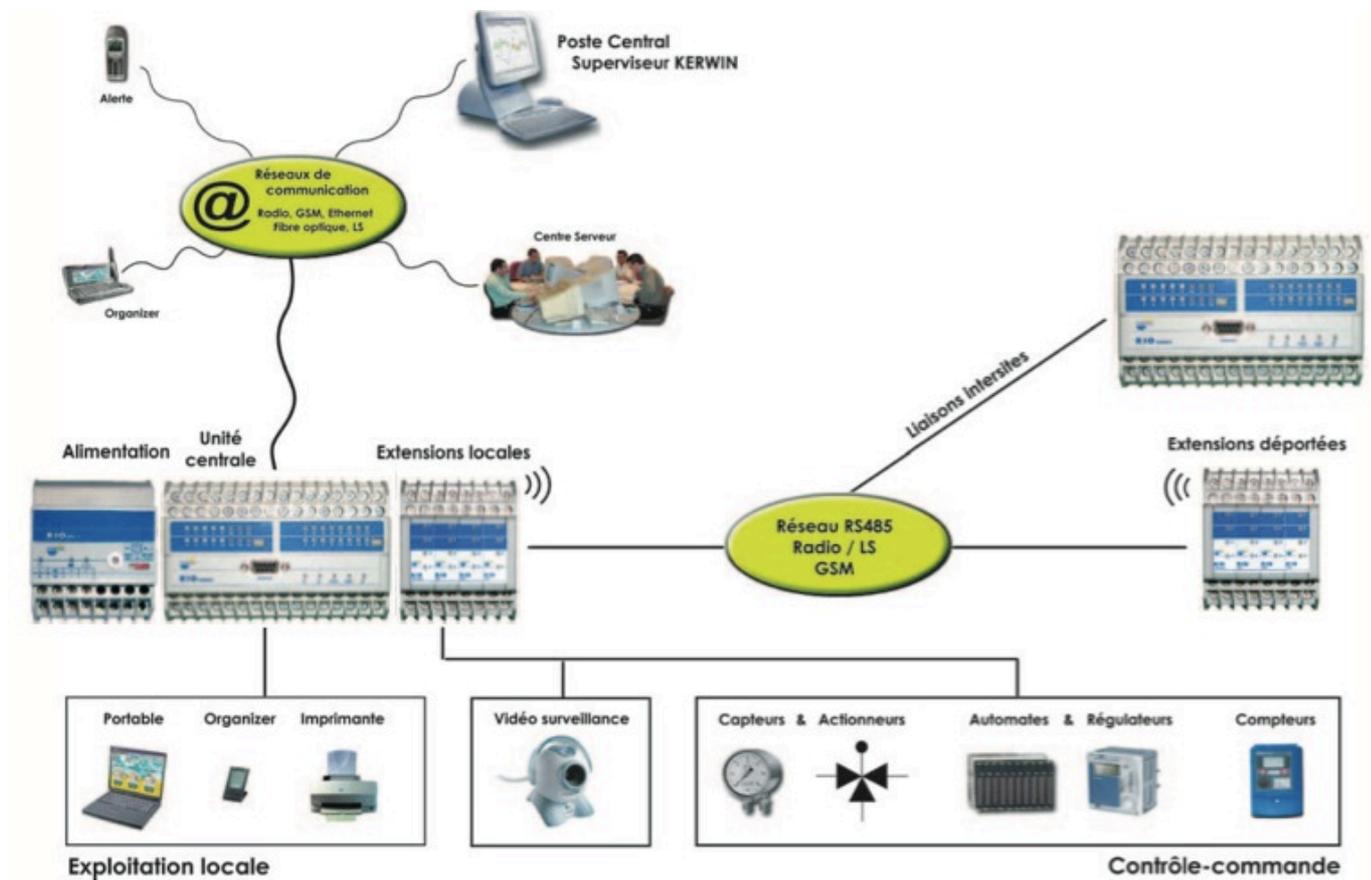


### 5.3.2 Contrôleur RIO de NAPAC

Automate de trélagestion ([www.napac.fr/pdf/fr/pdf/FP\\_RIO.pdf](http://www.napac.fr/pdf/fr/pdf/FP_RIO.pdf))



RIO repose sur une technologie 32 bits avec capacité d'archivage de données horodatées (alarmes, mesures, jusqu'à 125.000 enregistrements). Possibilités de transmission d'alarmes ou de données par SMS, Fax ou E-mail, la gamme RIO dispose d'un serveur Web embarqué permettant une exploitation par navigateur et une intégration du système dans une architecture Intranet / Internet.



# RIO



## CARACTERISTIQUES MATERIELLES

### CARACTÉRISTIQUES GÉNÉRALES

#### Boîtier RIO :

- PVC UL94-V0, format modulaire MULTI 9°
- Montage sur rail DIN
- Dim. : 95x70x75 mm (EXTENSION) et 95x70x140 mm (PHENIX)
- Raccordements sur bornier à vis, électronique séparable
- Leds de contrôle système et tous signaux d'E/S

#### Cartes électroniques :

- T1 : 38,7 x 60 mm
- T2 : 60 x 62 mm
- T3 : 60 x 136 mm

**Conditions d'utilisation :** Température 0 à 55°C, jusqu'à 80% d'humidité

#### Alimentation :

- 230 VAC par modules RIO ALIM ou 24 VDC lorsque l'on utilise la carte 8 AIS

#### Réglementation (agréments en cours) :

- CEM : industrie lourde EN 50082-2 et EN 50081-2
- Foudre : EN 61000-4 et -5
- Sécurité électrique : EN 60950
- Norme téléphonique européenne CTR 21
- Modem PCMCIA pour Flowtel, agréments tous pays

### CARACTÉRISTIQUES DES CPU

#### Carte CPU Phenix (T3) :

- Processeur NEC V850 E/MS1, 32 bits, 30 MHz
- Programme en Flash Eprom : 1 Mo
- Modem 33,6 Kb/s, fonction Identification de l'appelant
- Capacité de stockage : 512 Ko SRAM, 8 Mo DRAM
- 2 liaisons réseau RS485 multiprotocoles (jusqu'à 115 kBauds)
- 1 liaison console de paramétrage et d'exploitation (SUBD9 RS232)

**Carte CPU Flowtel** (format RIO T3 en cours de développement, existe au format "TELEFLO") :

- Processeur 32 bits, 100 MHz - Flash Eprom : 10 Mo
- Capacité de stockage : 1 Mo SRAM - 16 Mo DRAM
- Port PCMCIA pour modem RTC V90 (56 kBauds) ou GSM, RNIS, etc.
- Port Ethernet 10 Base-T
- 2 liaisons réseau RS232 / 485 multiprotocoles (115 kBauds)
- 1 liaison console de paramétrage et d'exploitation (SUBD9 RS232)



### CARACTÉRISTIQUES DES CARTES FILLES

#### Cartes verticales 4 voies (T1) :

- RIO1 4DI, 4 entrées logiques (alim interne ou externe 5 à 30 VDC, 4 voies au même potentiel)
- RIO1 4DO, 4 sorties logiques électromécaniques (Relais 60 V / 1A, 4 voies indépendantes)
- RIO1 4DO-S, 4 sorties logiques statiques (4 voies indépendantes, 60 V / 100 mA)

- RIO1 4AI, 4 entrées analogiques universelles (Pt100, Pt1000, 0-1 V, 0-10 V, 4-20 mA, 4 voies au même potentiel)
- RIO1 4AI-C, 4 entrées analogiques 4-20 mA (auto-alimentation des capteurs, 4 voies indépendantes)
- RIO1 4AIO-C, 2 entrées + 2 sorties analogiques 0-20 mA / 4-20 mA (auto-alimentation des capteurs et actionneurs, 2 entrées au même potentiel, 2 sorties au même potentiel)
- RIO1 4AIO-V, 2 entrées + 2 sorties analogiques 0-10 V (auto-alimentation des capteurs et actionneurs, 2 entrées au même potentiel, 2 sorties au même potentiel)
- RIO1 2AI-V1, 2 entrées +/- 100 V (2 entrées au même potentiel)
- RIO1 2AI-V2, 1 entrée +/- 100 V et 1 entrée +/- 100 mV (2 entrées au même potentiel)
- RIO1 2AI-V3, 2 entrées +/- 10 V (2 entrées au même potentiel)
- RIO1 CPU (pour module d'extension, alimentation 12 ou 24 VDC, isolation du bus et extension d'adressage)

#### Cartes horizontales 16 voies (T2) :

- RIO2 16AI, 16 entrées analogiques universelles (4-20 mA, PT100, PT1000, 2 Kohm, tension 0 à 1 V ou 0 à 10 V, 16 voies au même potentiel)
- RIO2 16DI, 16 entrées logiques et comptage (entrées autoalimentées, comptage 32 bits 100 Hz, 16 voies au même potentiel)
- RIO2 16DIO, 12 entrées logiques et 4 sorties logiques (4 sorties 60 V / 2 A isolées entre elles, comptage 32 bits 100 Hz, 12 entrées au même potentiel)
- RIO2 8AIS, 8 entrées universelles et alimentation (8 entrées universelles, dont 2 entrées TOR comptage 3 Hz, 1 sortie logique 40 V / 1 A et une alimentation 400 mA avec chargeur de batterie permettant un raccordement direct en 24 VDC sans module d'alimentation)

#### Cartes de communication :

- RIO1 COM, Communication série isolée (RS485 ou RS232, isolation 1 KV)
- RIO1 GSM, Communication GSM
- RIO1 AGR, Communication radio 10 mW (fonction récepteur pour balises autonomes AGR ou routeur bidirectionnel radio)
- RIO1 PL, Modem LS (routeur multipoints bidirectionnel pour lignes spécialisées)
- RIO1 OF, Interface Fibre Optique (connectique ST, fonction répéteur)
- RIO2 COM1, multi-format (RS232, RS485, collecteur ouvert)
- RIO2 COM2, bus multiprotocole

### MODULES D'ALIMENTATION

- RIO2 UPS-1, Alimentation 230 VAC / 750 mA avec chargeur de batterie 12 V, autosurveillance de l'alimentation secteur, 1 sortie RS485 isolée

#### Nota :

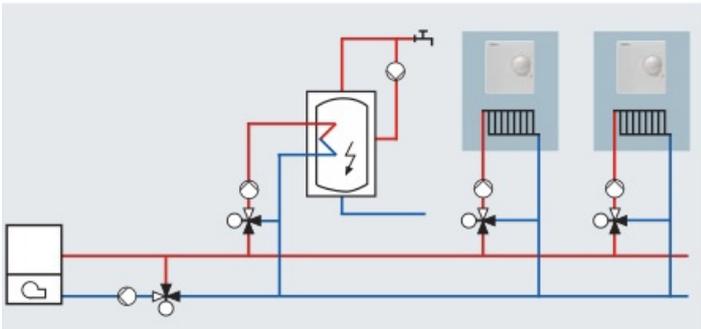
Ce module d'alimentation n'est pas nécessaire lorsque l'on utilise une carte 8 AIS qui permet une alimentation directe en 24 VDC.



33, bd du général Martial Valin  
75015 PARIS  
tel : 01 44 25 20 40  
fax : 01 44 25 20 50  
mail : [napac@napac.fr](mailto:napac@napac.fr)  
web : [www.napac.fr](http://www.napac.fr)

### 5.3.3 Siemens : Contrôleur modulaire RMH760

[www.siemens.com/hvac](http://www.siemens.com/hvac)



- Modular heating controller RMH760**
- Two controlled heating circuits
  - Four 7-day programs and yearly calendar
  - Control of DHW storage tank charging
  - Control of circulating pump
  - Boiler temperature control including multistage or modulating burner control

### 5.3.4 Honeywell : Contrôleur HVAC

MiniRégulateur 22 E/S pour applications HVAC , doté de fonction d ecommunication LON ou M-Bus



Contrôle software Carte configurable par logiciel avec des centaines d'applications possibles. Personnalisation des paramètres en fonction des installations.

Entrée physique

- Entrées analogiques (AI) pour sondes NTC20k
- Entrées digitales (DI) pour contact libre de potentiel ou 24 Vdc

Sortie physique

- Sorties analogiques (AO) 0..10 Vdc
- Sorties digitales (DO) 24 Vac

### 5.3.5 API contrôleurs génériques



API Schneider Premium



micro-API Schneider Twido



API Siemens S7 300



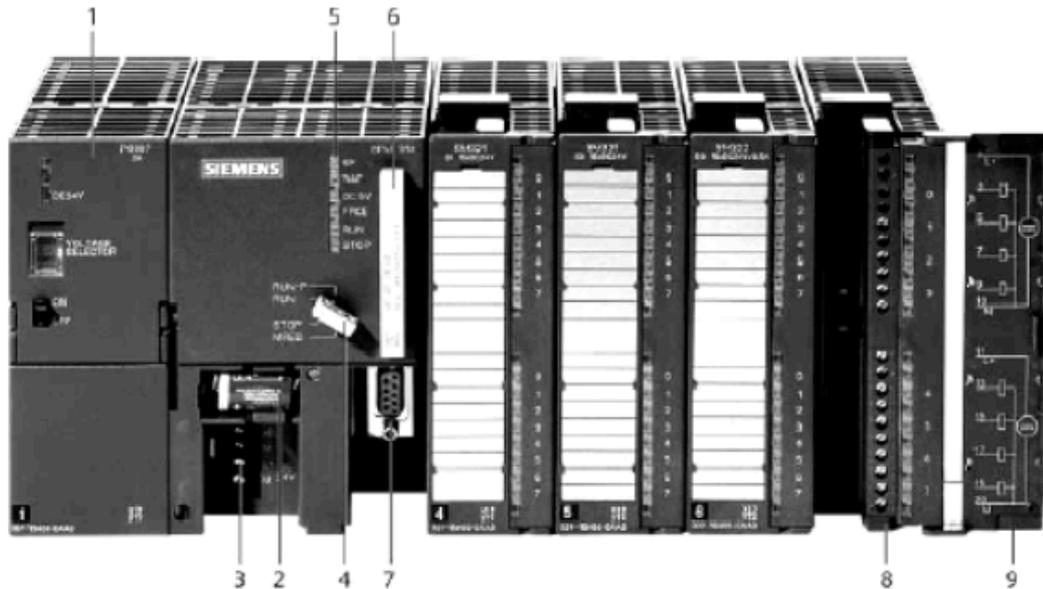
micro-API Siemens LOGO

## 6 STRUCTURE D'UN API

Les API ont une structure identique et ils se différencient par :

- leur taille (nano, micro, standard)
- Le nombre d'E/S maximum qu'ils peuvent gérer
- La rapidité de leur processeur central

### API SIEMENS série S7-300



Automate modulaire (Siemens)

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

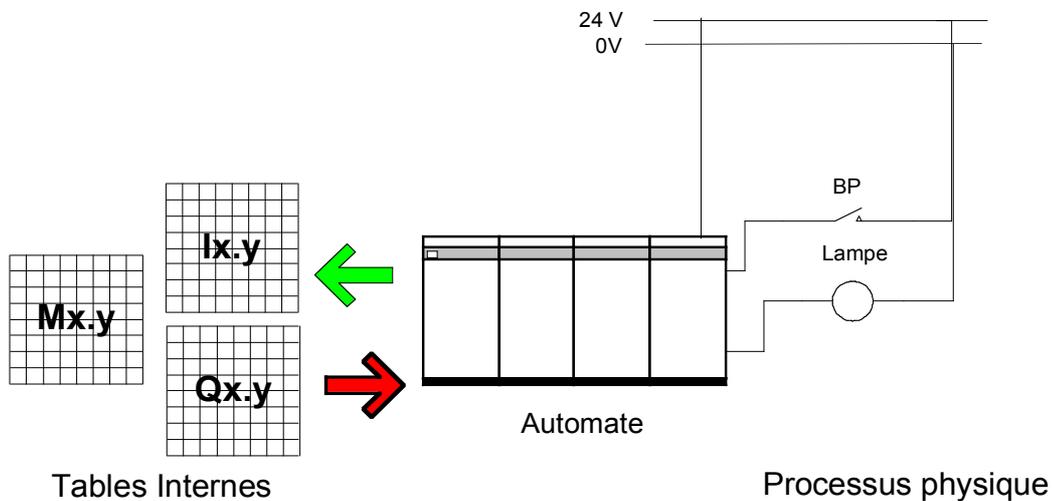
## 7 FONCTIONNEMENT D'UN API / CONTROLEUR

Les machines cibles d'exécution de programme d'automatisme sont appelées des **Automates** :

- [Automate Programmable Industriels \(API\)](#)
- [Automate logiciel](#) sur micro-ordinateur PC ([Soft PLC](#)).

## 7.1 Cycle API

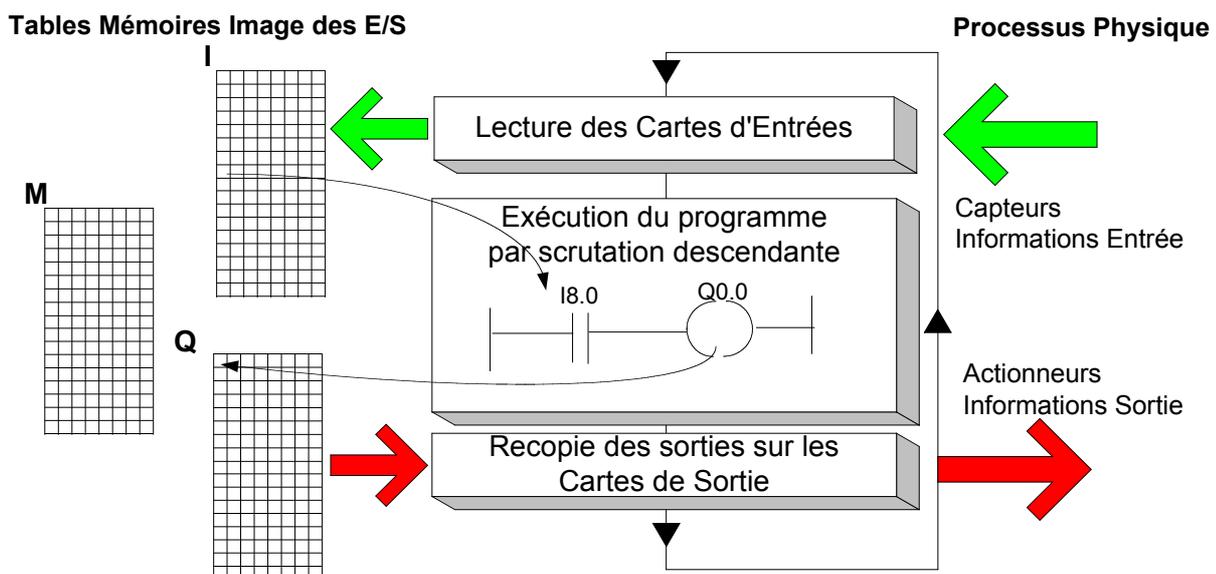
Un automate est raccordé à des capteurs/Actionneurs physique qu'il pilote selon un programme informatique en mémoire.



L'automate exécute un programme en répétant éternellement le même cycle :

1. Lecture des Entrées et recopie en Mémoire Image E (**MIE**)
2. Exécution (Scrutation) du programme,
3. Recopie de la Mémoire Image S (**MIS**) vers les Cartes de sortie.

**Un API ne « regarde » les entrées qu'en début de cycle et ignore tout changement pouvant subvenir durant un cycle.**

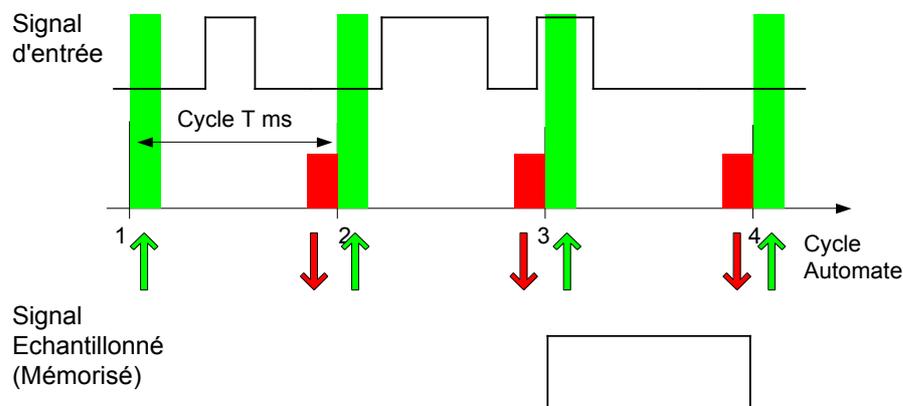


## La durée d'un cycle Automate Tcy doit :

- ⇒ être la plus petite possible pour assurer un **fonctionnement temps réel**.
- ⇒ Tcy doit être très inférieur à la dynamique du processus commandé (**temps typique 20 ms**).
- ⇒ être surveillée pour ne pas dériver, par une technique de **chien de garde (WatchDog)** paramétrable (en général 100 ms)

## 7.2 Fonctionnement Temps Réel « tâche »

Cycle d'échantillonnage des entrées sur un API



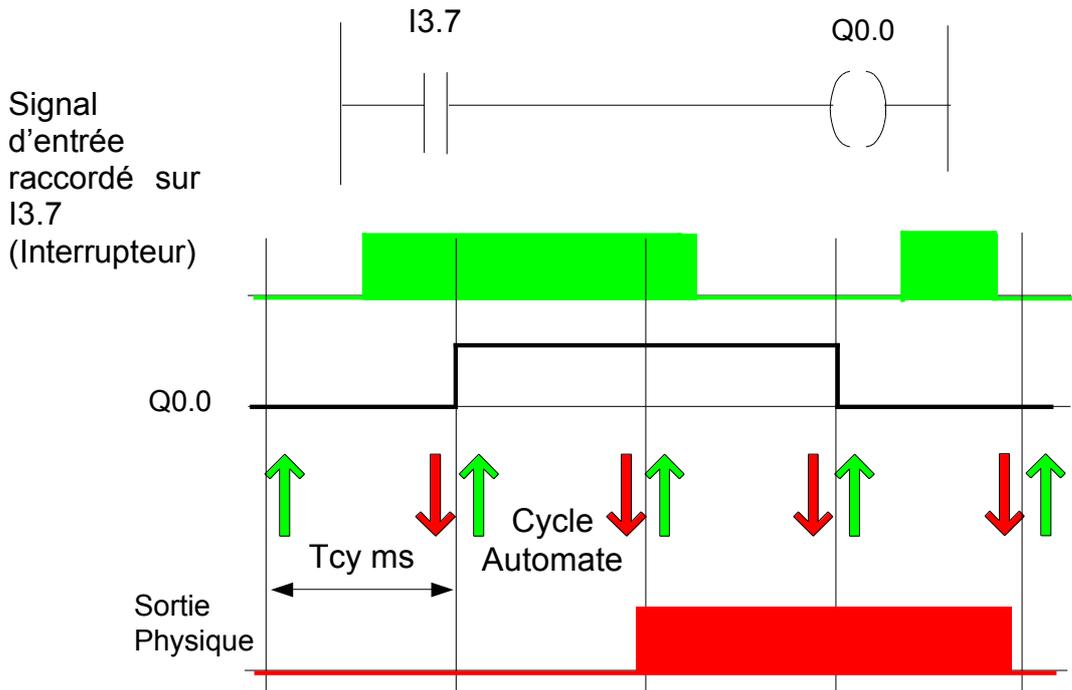
UN changement sur une entrée qui survient après le début de cycle et se termine avant le début du cycle suivant ne sera pas détecté par l'automate : **l'automate est « aveugle » durant son cycle !**

La durée d'un cycle Automate Tcy doit :

- être la plus petite possible pour assurer un **fonctionnement temps réel**.  
Tcy doit être très inférieur à la dynamique du processus commandé (général 20 ms).
- être surveillée pour ne pas dériver, par une technique de **chien de garde** paramétrable ( en général 100 ms)

### 7.3 Affectation d'une sortie physique sur un automate

Les sorties d'un automate (Qx.y) sont récupérées physiquement sur les cartes de sorties en fin de cycle à partir de la table interne.

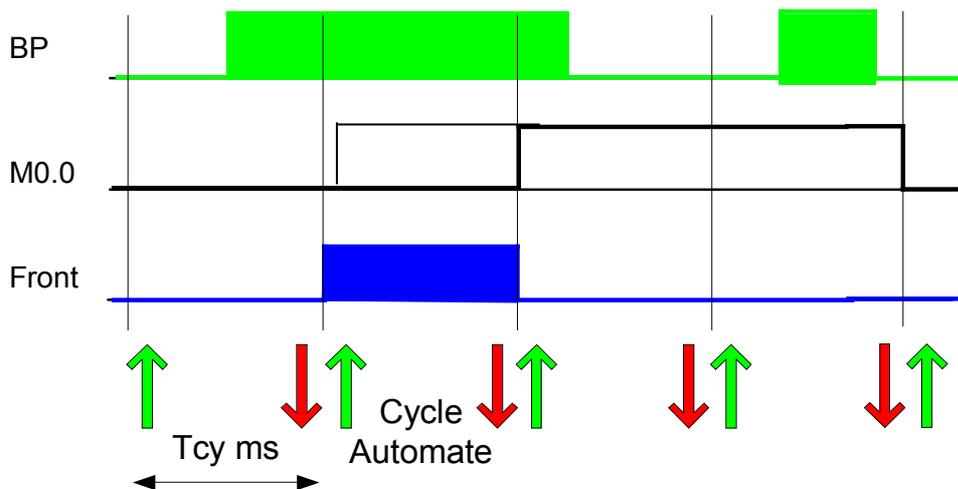


La sortie physique (par ex. une lampe) commandée par Q0.0 ne sera mise à jour qu'à la fin du cycle Automate, soit avec un décalage d'un cycle minimum (voire presque 2) sur la mise à 1 de la variable.

### 7.4 Détection d'un front montant sur un automate

Problème de détection d'un front montant sur une entrée BP (passage de 0 à 1)

Au sens Automate, un front montant sera détecté quand **BP=1 au cycle courant** et que **BP=0 au cycle précédent**



Il faut donc mémoriser dans un bit interne (M0.0) l'état de BP pour en tenir compte au cycle suivant, comme étant l'état de BP au cycle précédent;

## 7.5 Codage en Langage à contact de la détection de front montant

On peut employer deux méthodes pour détecter un front montant par programme API.



L'automate indiquera la détection d'un front montant dans une variable interne **Front** qui sera vraie pendant 1 cycle automate.

## 7.6 Variables Mémoire API

Les entrées de la PO sont recopiées dans des Bits d'Entrée **I x.y**

Les Sorties de la PO sont recopiées dans des Bits de Sortie **Q x.y**

**X** : Rang de la carte sur l'API (0 à 15)

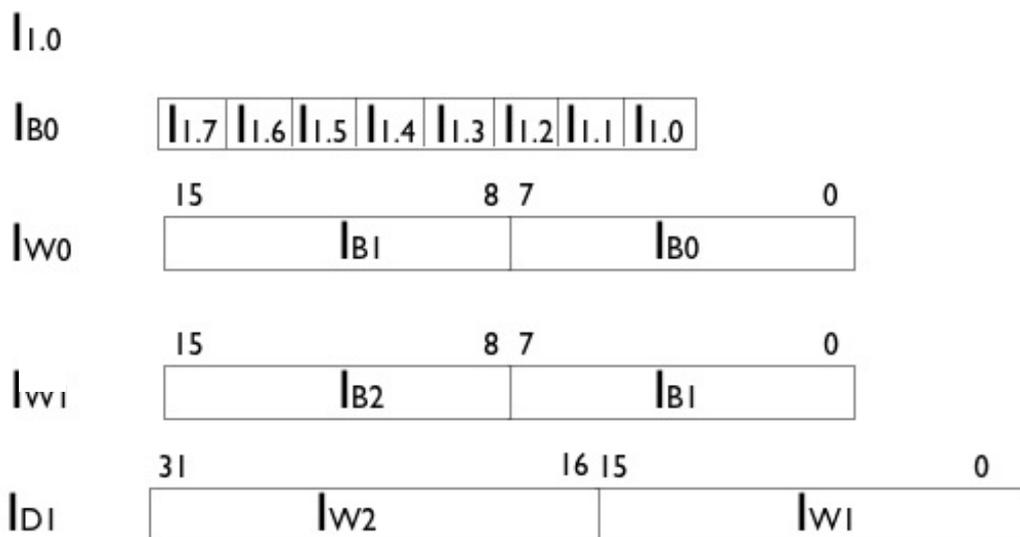
**Y** : N° de l'entrée (sortie) sur la carte (0 à 7)

Des Bits Internes **M x.y**

**X** : N° du l'octet mémoire dans l'API (0 à 255)

**Y** : N° du bit dans l'octet (0 à 7)

Les Bits sont regroupés en Octets puis en Mots puis en Double Mot



### Exercices :

a) A quels bits image correspondent :

- Entrée N° 5 de la carte de rang 0
- Entrée N° 16 de la carte de rang 4
- Sortie N° 0 de la carte de rang 1

b) Quels sont les bits image contenus dans IB2

c) Quels sont les bits internes contenus dans MW92

## 8 LANGAGES DE PROGRAMMATION DES API

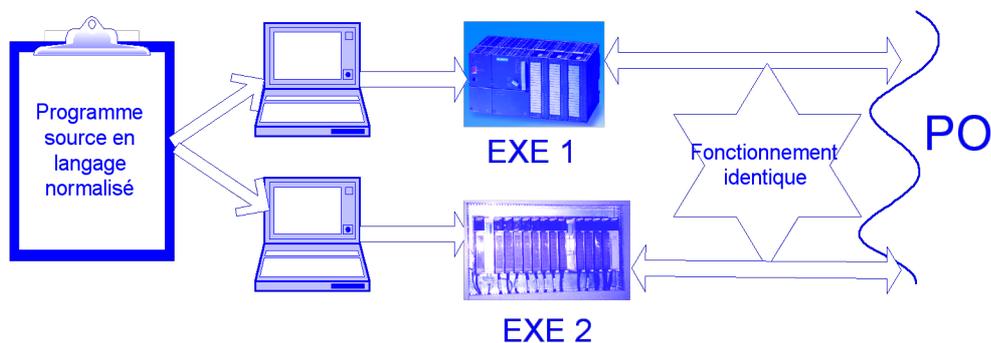
### Constat

- Grande disparité de langages ⇔ effet constructeur
- Langages différents d'un matériel à l'autre
- Langages « standards de fait » car très utilisés : contacts, mnémonique, ...

### Effet de Normalisation (Novembre 1993)

#### Norme IEC 61131 appelée communément : norme IEC 1131

- Part 1 : Présentation générale
- Part 2 : Spécifications techniques et matérielles des API et interfaces
- Part 3 : Langages de programmation (203 pages) norme / langages IEC 1131-3
- Part 4 : Guide utilisateur
- Part 5 : Communication

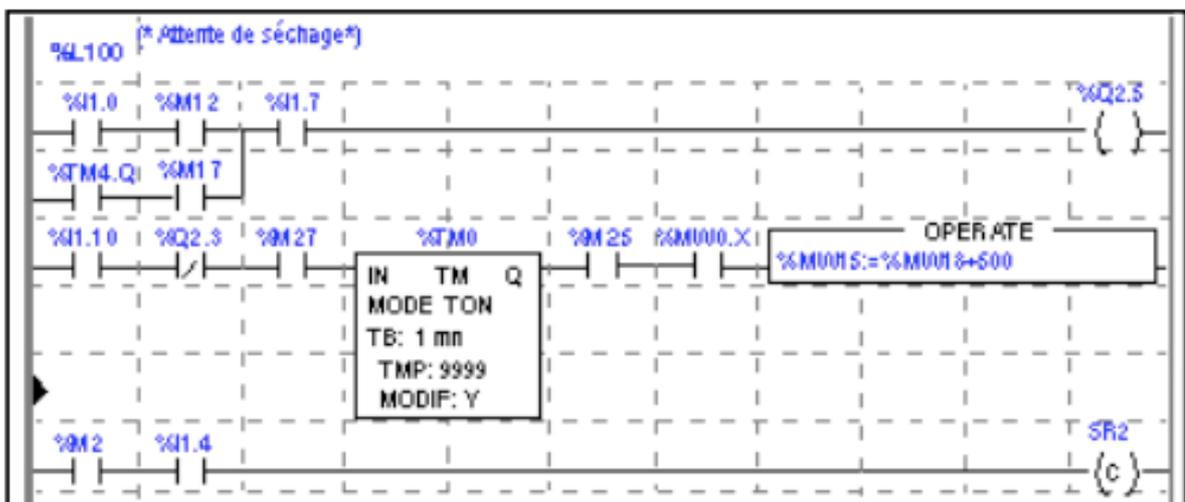


Il existe 5 langages de programmation des automates qui sont normalisés au plan mondial par la norme CEI 61131-3.

Chaque automate se programme via une console de programmation propriétaire ou par un ordinateur équipé du logiciel constructeur spécifique : ATELIER LOGICIEL (Siemens STEP7, SCHNEIDER UNITY ou PL7 Pro, WAGO CodeSys)

### 8.1 Langage à contacts (LD : Ladder diagram) :

Langage graphique développé pour les électriciens. Il utilise les symboles tels que : contacts, relais et blocs fonctionnels et s'organise en réseaux (labels).





## IV - Logique des contacts

C'est une présentation électrique des règles de l'algèbre de Boole par des contacts (interrupteurs) et des relais (bobines)

- Une variable logique tout comme un **contact** électrique peut prendre deux états : 0 : Ouvert 1 : fermé



- Une fonction logique correspond à une variable logique "de sortie" tout comme un **relais** (bobine) prend deux états : 0 : Inactivée 1 : Activée



Informatique Industrielle – Logique – Cours E. BAJIC – IUT Nancy Brabois

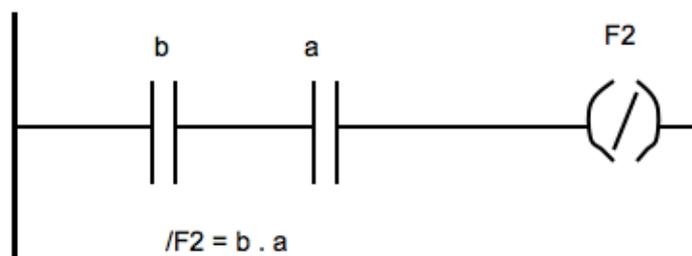
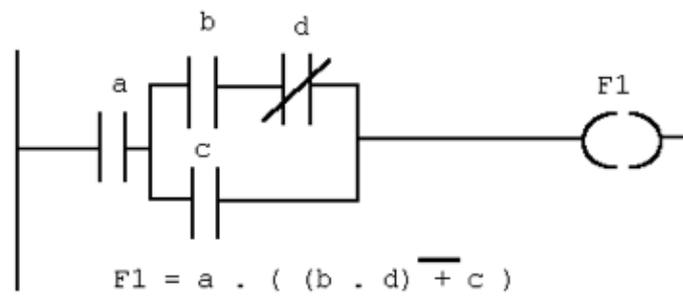


### IV.1 – Schéma à contacts

Un **Schéma à contacts** traduit électriquement les lois de composition + et . à partir de branchement Parallèle et Série.

Programme en

- Echelle à relais
- Schéma à Contact
- Ladder



## 8.2 Langage littéral structuré (ST : Structured Text)

Langage informatique de même nature que le Pascal, il utilise les fonctions comme *if ... then ... else ...* (si ... alors ... sinon ...)

Utilisé par les automaticiens.

```

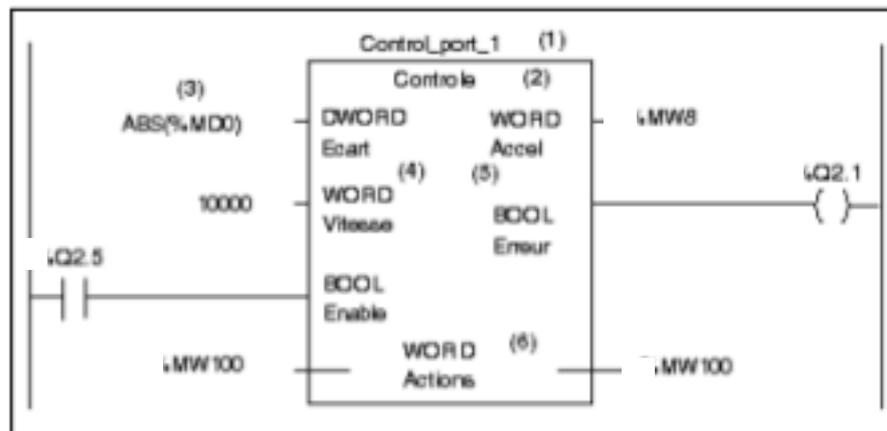
IF %M0 THEN
  FOR %MW99 := 0 TO $1 DO
    IF %MW100 [%MW99] > 0 THEN
      %MW110 := %MW100 [%MW99];
      %MW111 := %MW99;
      %M1 := TRUE;
      EXIT;          (*Sortie de la boucle FOR*)
    ELSE
      %M1 := FALSE;
    END_IF;
  END_FOR;
ELSE
  %M1 := FALSE;
END_IF;

```

## 8.3 Blocs Fonctionnels (FBD : Function Bloc Diagram)

Langage graphique par assemblage de blocs fonctionnels figurés par des boîtes. Les blocs sont programmés (bibliothèque) ou programmables. Les blocs sont programmés (bibliothèque) ou programmables.

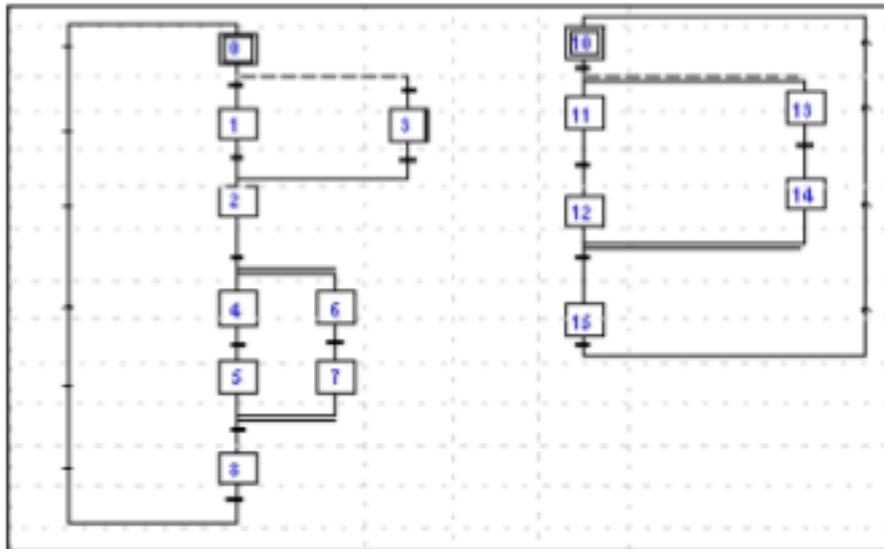
Utilisé par les automaticiens.



## 8.4 SFC (Sequential Function Chart)

La norme IEC 848 est nommée mais le terme GRAFCET n'est jamais utilisé, pourtant les analogies entre SFC et GRAFCET sont très grandes.

Le **SFC n'est pas décrit comme un langage** mais comme un moyen de structuration permettant d'effectuer des commandes séquentielles.



### 8.5 Liste d'instructions (IL : Instruction list)

Langage textuel de même nature que l'assembleur (programmation des microcontrôleurs).  
Très peu utilisé par les automatismes.

```

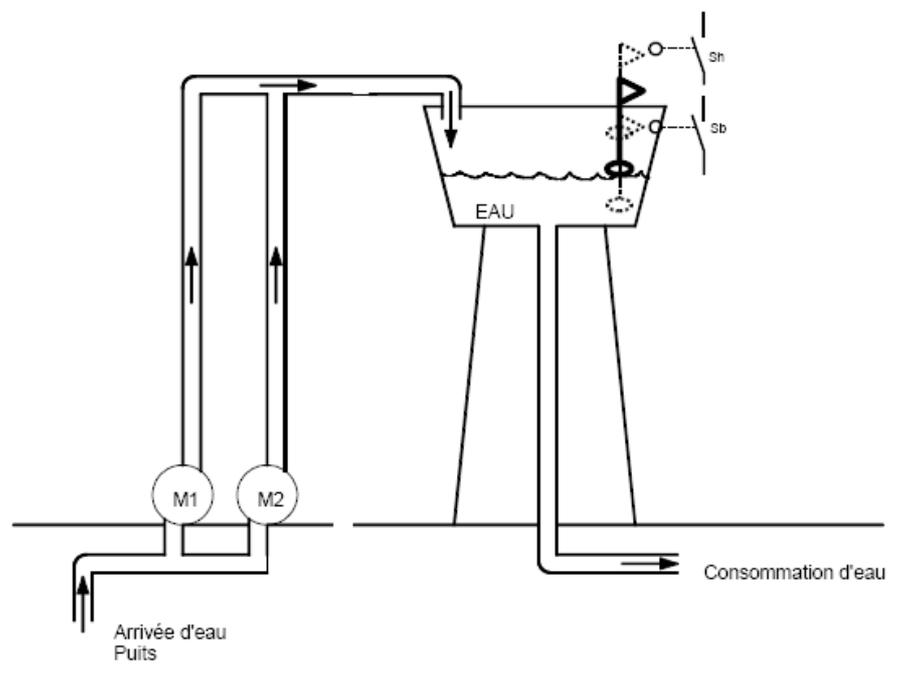
! %L0 : LD      %I1.0
        ANDN   %M12
        OR (   %TM4.Q
        AND   %M17
        )
        AND   %I1.7
        ST    %Q2.5
! %L5 : LD      %I1.10
        ANDN   %Q2.5
        ANDN   %M27
        IN    %TM0
        LD    %TM0.Q
        AND   %M25
        AND   %M00.X5
        %M00.S := [%M00.S+500]
    
```

## 9 PROGRAMME EN LOGIQUE COMBINATOIRE OU PROGRAMMES EN LOGIQUE SEQUENTIELLE

Les langages de programmation supportés par les API permettent de résoudre des problèmes de nature :

- Combinatoire : décrit par des équations logiques : LD
- Algorithmique : décrits par un algorithme ou organigramme : ST, IL
- Séquentiel : Analyse réalisée par grafcet et traduite en SFC ou LD.

### 9.1 Exercice de Programme API : Commande de gestion d'un Réservoir



Commander 2 pompes de remplissage d'un réservoir de telle sorte que :

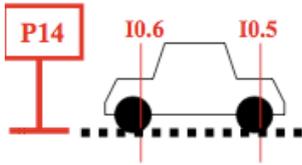
- Quand la cuve est pleine ( $Sh=1$  et  $Sb=1$ ), aucune pompe ne fonctionne.
- Quand la cuve est vide ( $Sh=0$  et  $Sb=0$ ), les 2 pompes fonctionnent
- Quand la cuve est à moitié vide (ou pleine..) ( $Sh=0$  et  $Sb=1$ ), une seule pompe fonctionne. Le choix se fait à l'aide d'un commutateur  $C=1$  alors la pompe M1 fonctionne.

**Il vient intuitivement**

$$\begin{cases} KM1 = \overline{Sh} \times \overline{Sb} + C \times \overline{Sh} \\ KM2 = \overline{Sh} \times \overline{Sb} + \overline{C} \times \overline{Sh} \end{cases}$$

## 9.2 Exercice de Programme API : Comptage Parking

### Comptage de visiteurs dans un parking: énoncé/solution



Un passage contrôle, grâce à 2 cellules (I0.5 et I0.6), les entrées et les sorties des véhicules.  
 L'ordre dans lequel elles sont occultées indique le sens.  
 Une lumière (Q0.2) avertit que le parking est plein (100 places)

