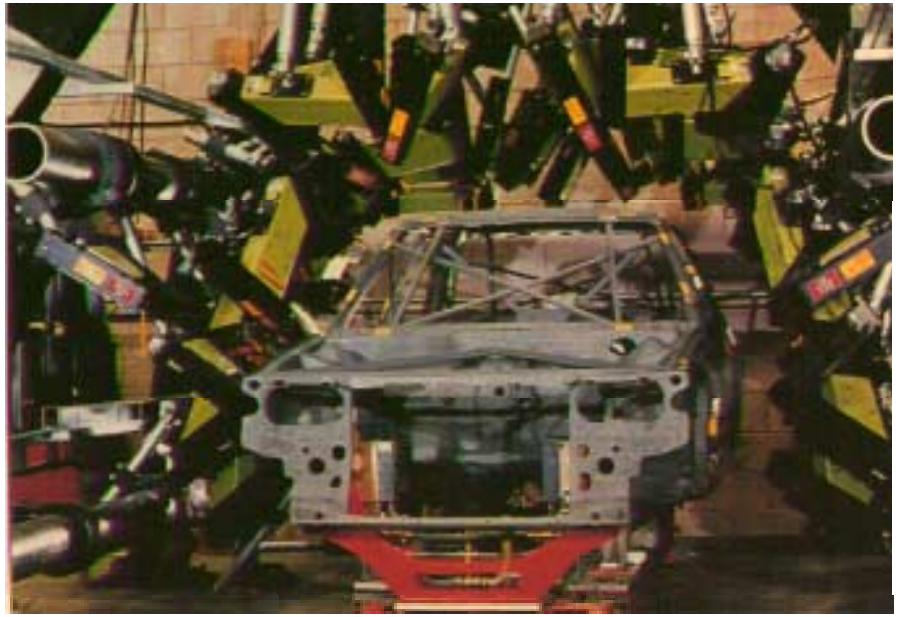
Les Automates Programmables Industriels (API)

PLC : programmable logic controller

Philippe RAYMOND - révision mai 2004

Historique



Ambiance industrielle

bruit électrique

poussière

température

humidité

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.

Cahier des charges de l’époque

Solution pour un coût acceptable

utilisable par le personnel en place

programmation facile

Supporter l’ambiance industrielle

Adaptation aux nombreuses variétés E/S

Simplicité de mise en œuvre

L’ordinateur en 1969

coût “astronomique”

utilisation complexe

nécessite un environnement particulier

Juillet 1969

La Mission Apollo XI dépose N. Armstrong sur la lune ! Deux GE 635 ont contrôlé le vol.

The first PLC, model 084, was invented by Dick Morley in 1969

The “084” - Details

The **“084”** consisted of three major components mounted on two vertical rails, one of which was hinged to allow for service access to the front and back.

Ladder Logic:

The use of **Ladder Logic** was significant in the rapid acceptance of the **“084”** because the very same engineers and electricians who designed and maintained Factory Automation Systems could also program an **“084”**. Ladder Logic was simply an electronic version of the elementary electrical diagram that they already used -- not the case for other types of control systems being designed at the time.

The “084” - Details

Input/Output Rack (top)

Two I/O Racks could be mounted on top of the **“084”** for a total capacity of **256 I/O Points** (only one mounted on the unit shown).

CPU (middle)

The middle unit contained the **CPU**. The “**084”** had ***1K x 16 Bit Core Memory***, which included both the **operating system memory** as well as the **User Program**.

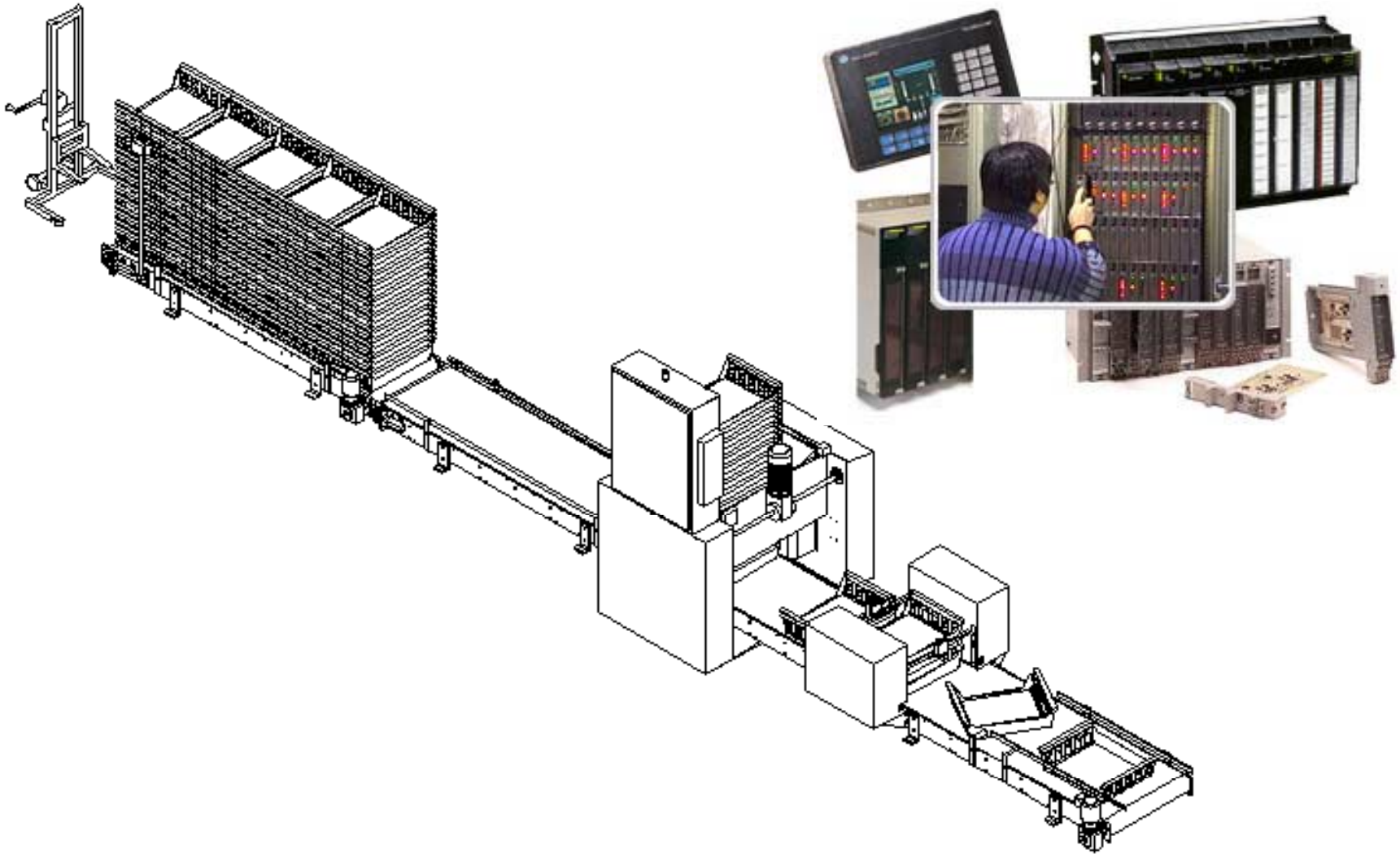
Power Supply (bottom)

A **Single Phase 115V Line** was connected to the front of the Power Supply Module, which supplied **DC Power** to the rest of the unit.

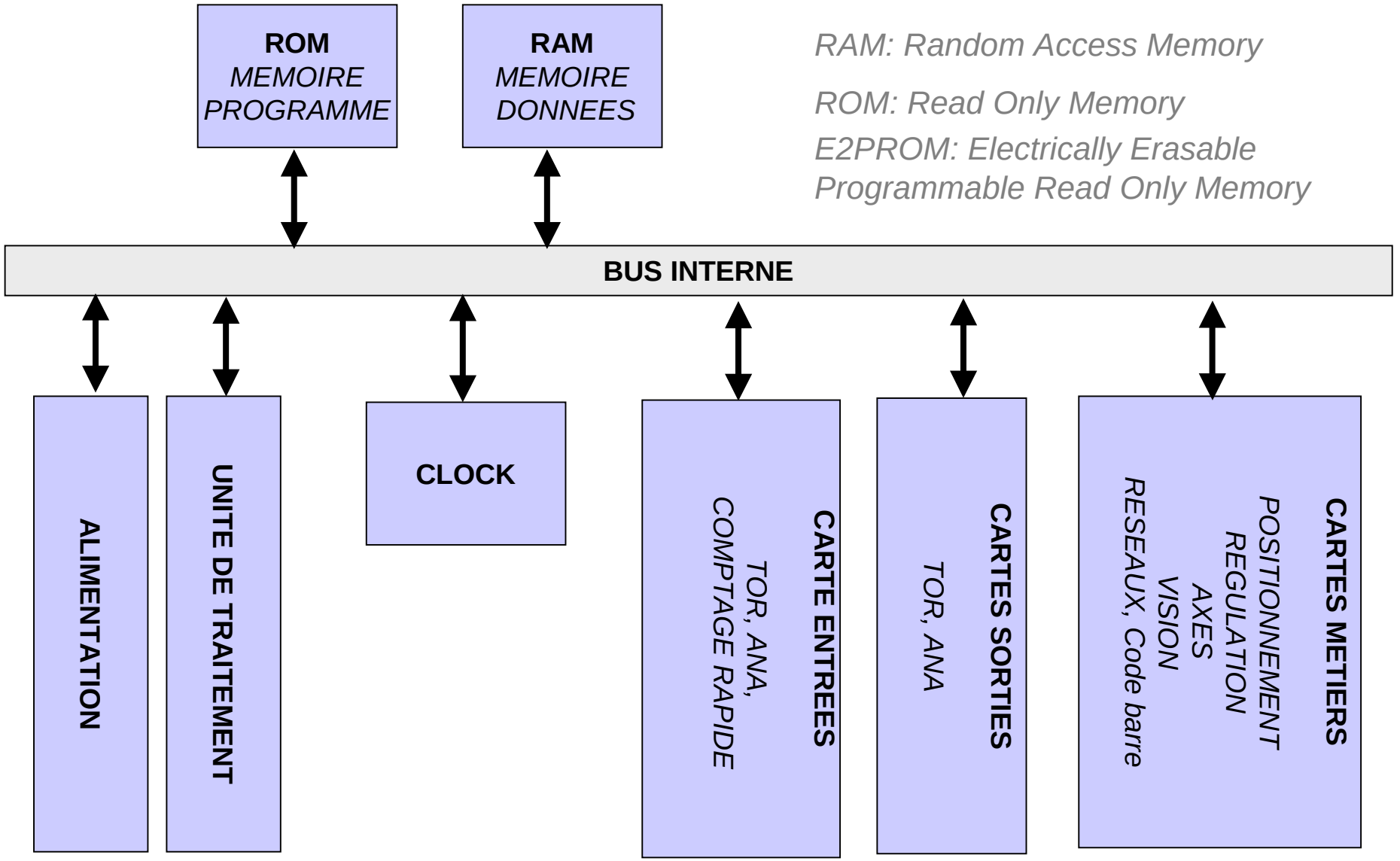
Options:

Timer
Counter
Portable Programming Panel
Magnetic Tape Cartridge Program Loader

Les API d’aujourd’hui



Architecture interne

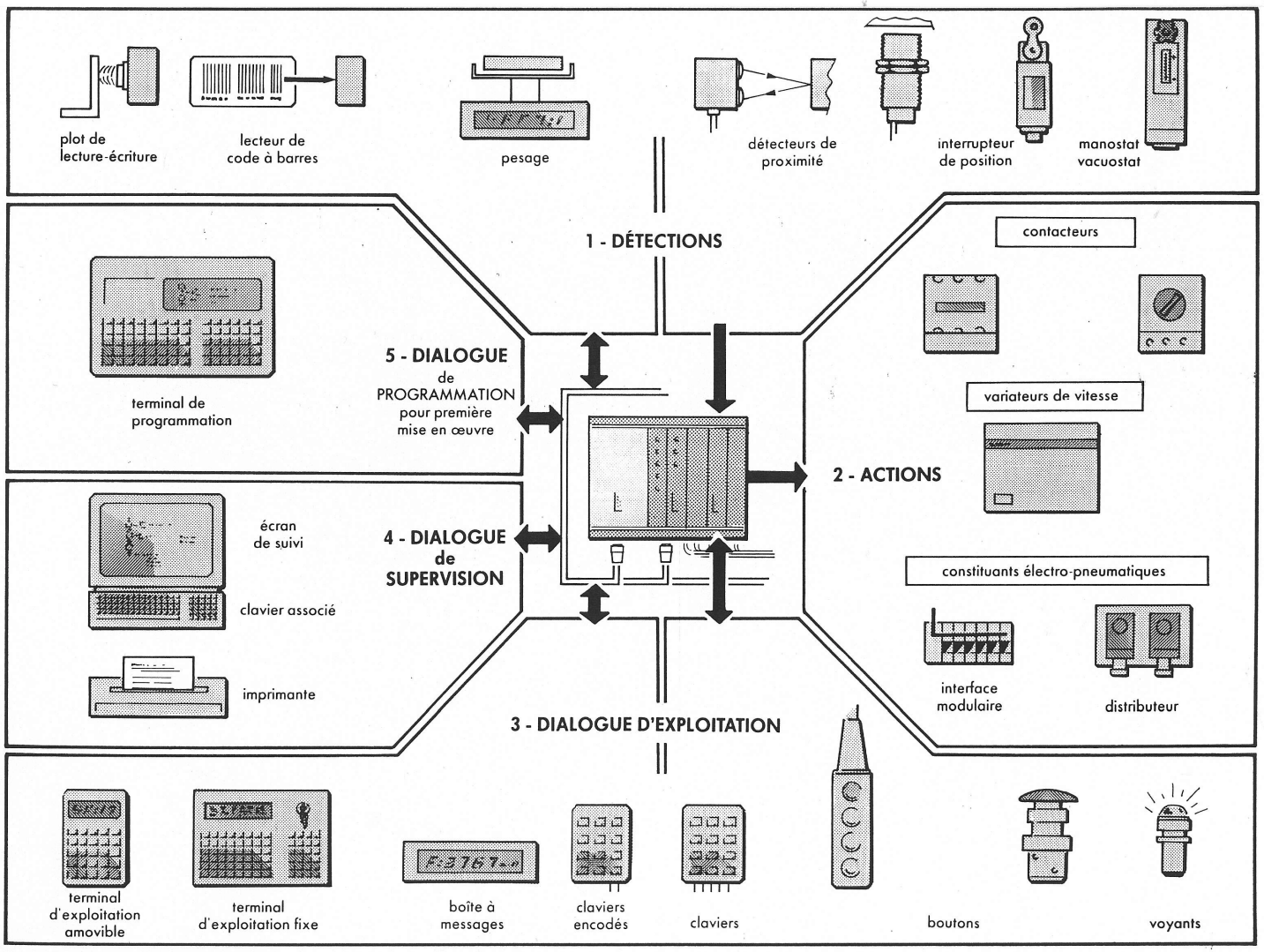


RAM: Random Access Memory

ROM: Read Only Memory

E2PROM: Electrically Erasable Programmable Read Only Memory

Les fonctions



Essai de classification

nombre
d'équipement

Cartes dédiées

?

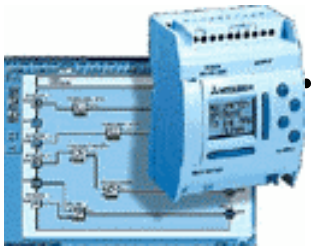
automate compact

automate modulaire

(réseaux et métiers)

SOFT PLC : Pc industriel et logiciel de contrôle commande

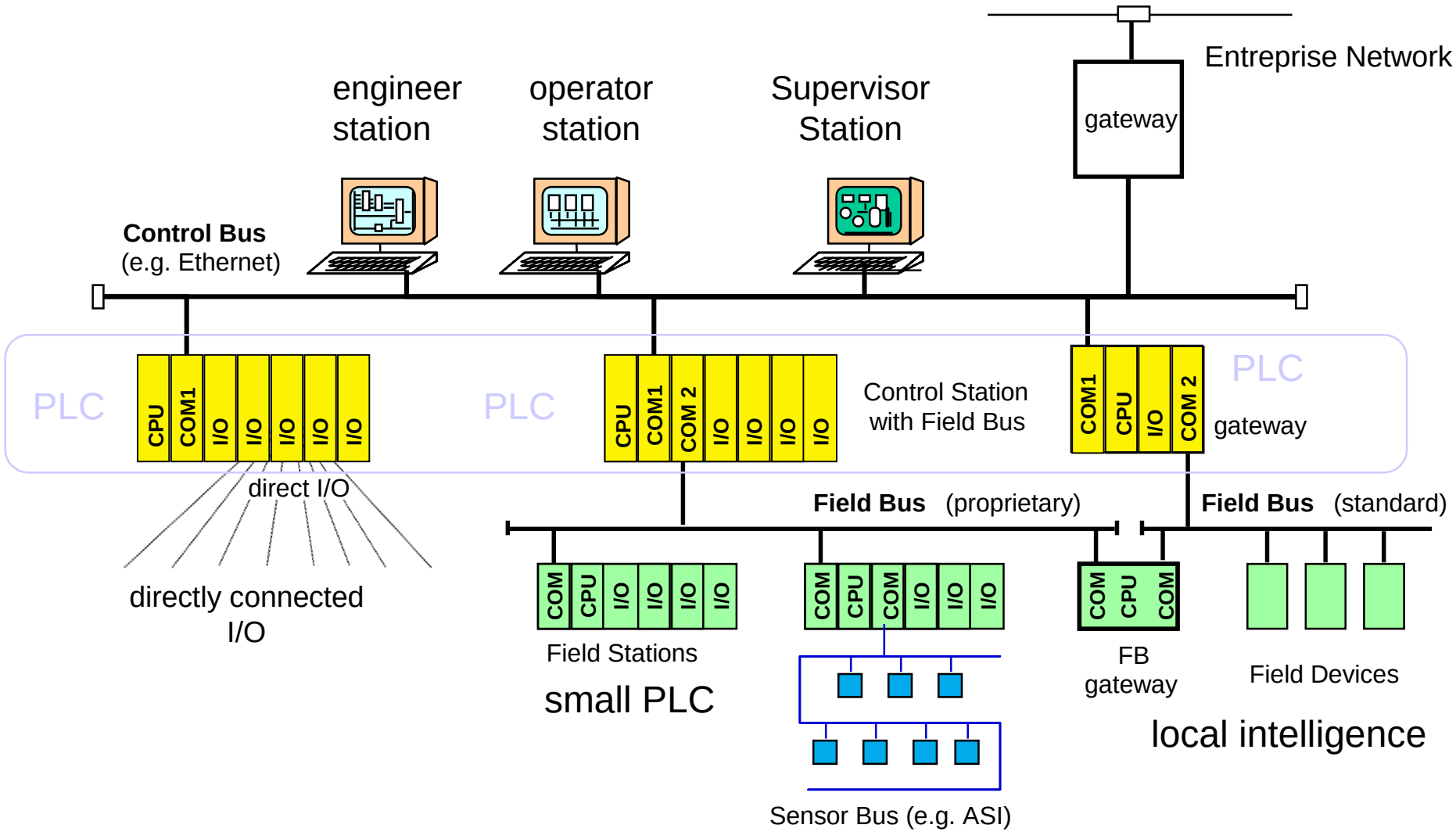
Logique cablée



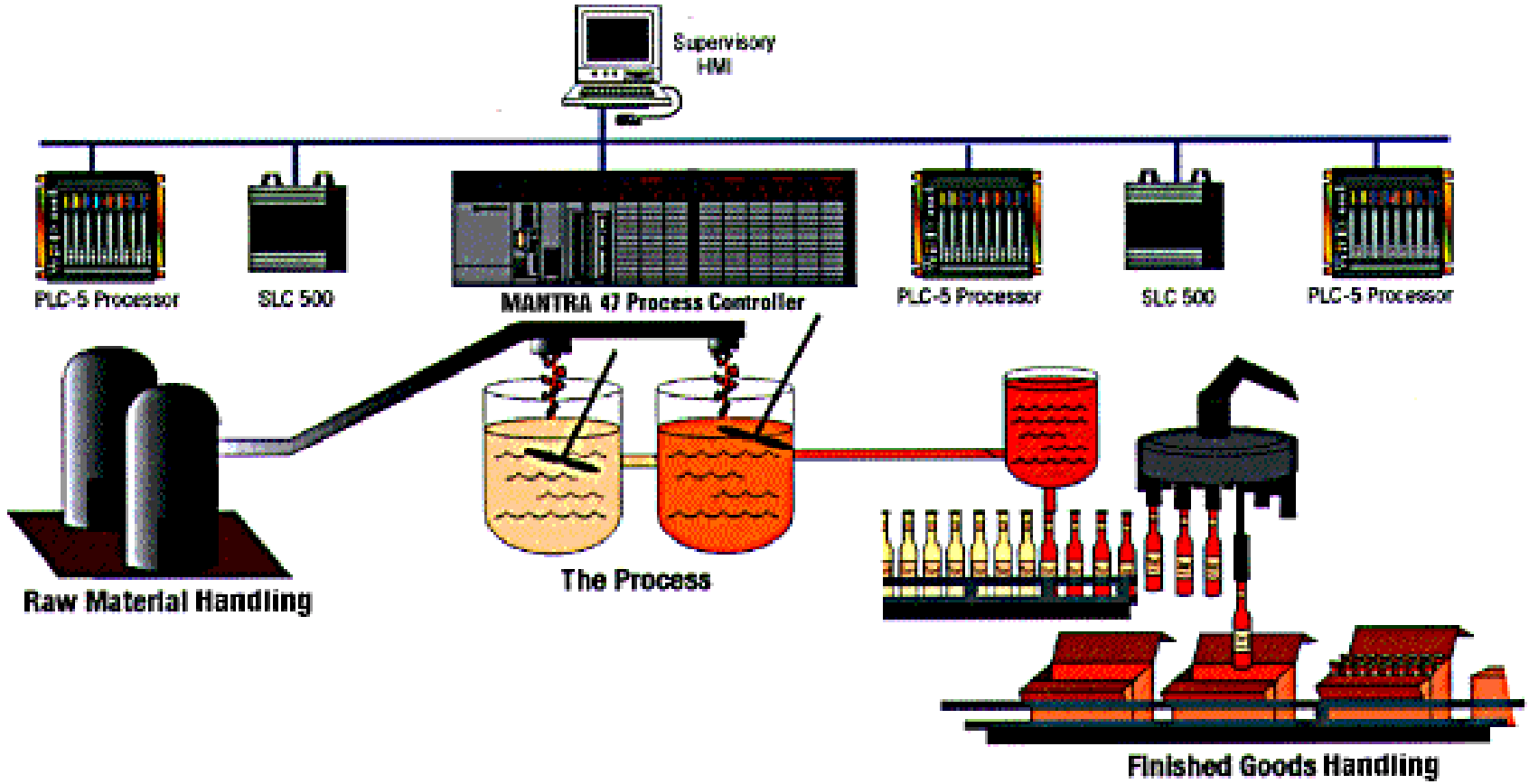
Relais programmable

Volume & niveau
d'automatisme

Architecture réseau



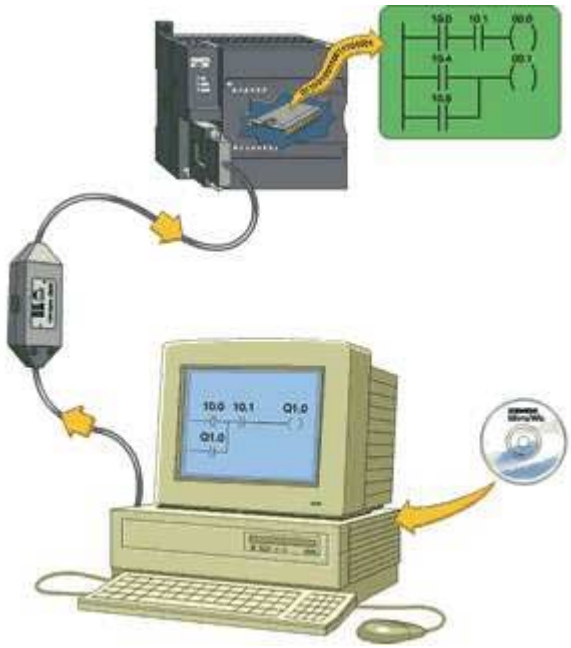
Exemple


Supervisory
HMI
PLC-5 Processor
SLC 500
MANTRA 47 Process Controller
PLC-5 Processor
SLC 500
PLC-5 Processor
Raw Material Handling
The Process
Finished Goods Handling

Aspect programmation

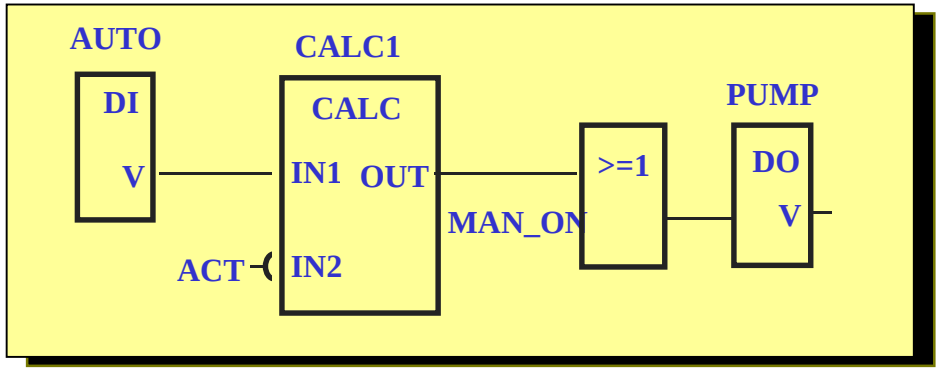
FONCTIONS DES ATELIERS LOGICIEL

- *gestion des projets*
- *configuration API*
- *édition, compilation*
- *Transfert console <> API*
- *Mise au point dynamique*

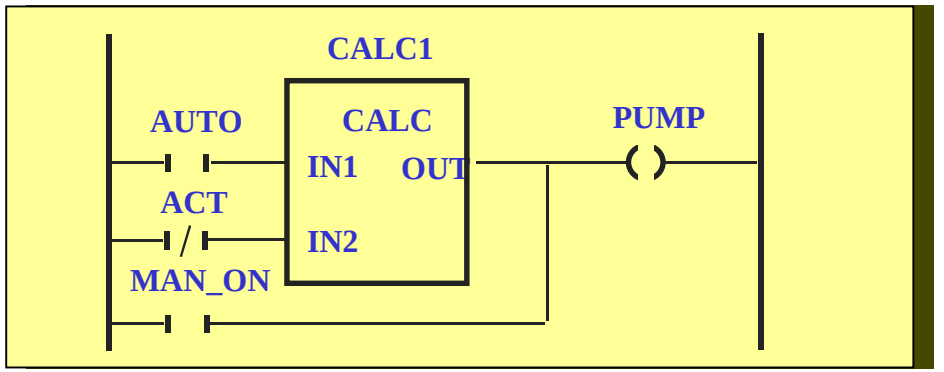


Les langages IEC1131

Function Block Diagram (FBD)



Ladder Diagram (LD)



Les langages IEC1131

Structured Text (ST)

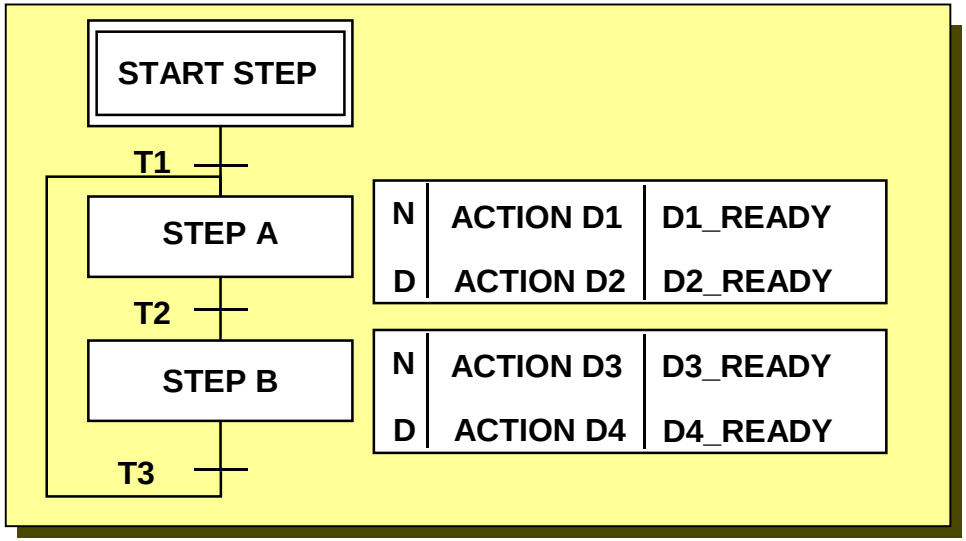
```
VAR CONSTANT X : REAL := 53.8 ;
Z : REAL; END_VAR
VAR aFB, bFB :  FB_type; END_VAR

bFB(A:=1, B:='OK');
Z := X - INT_TO_REAL (bFB.OUT1);
IF Z>57.0 THEN aFB(A:=0, B:="ERR");
ELSE  aFB(A:=1, B:="Z is OK");
END_IF
```

Instruction List (IL)

```
A: LD   %IX1 (* PUSH BUTTON *)
   ANDN %MX5 (* NOT INHIBITED *)
   ST   %QX2 (* FAN ON *)
```

Les langages IEC1131

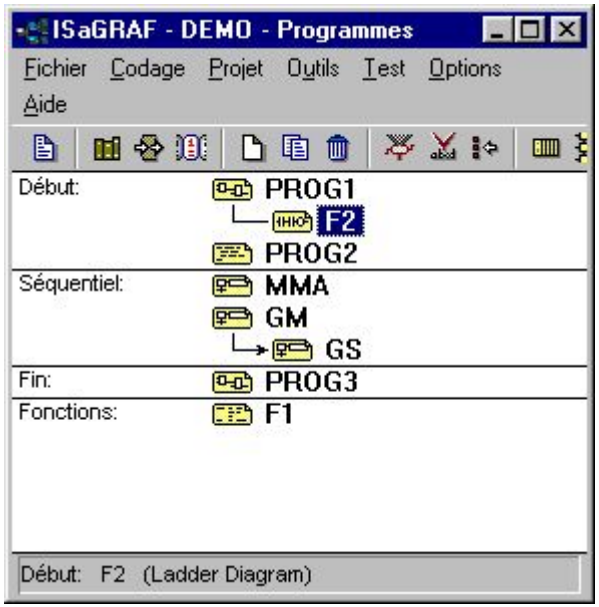
Sequential Flow Chart (SFC)



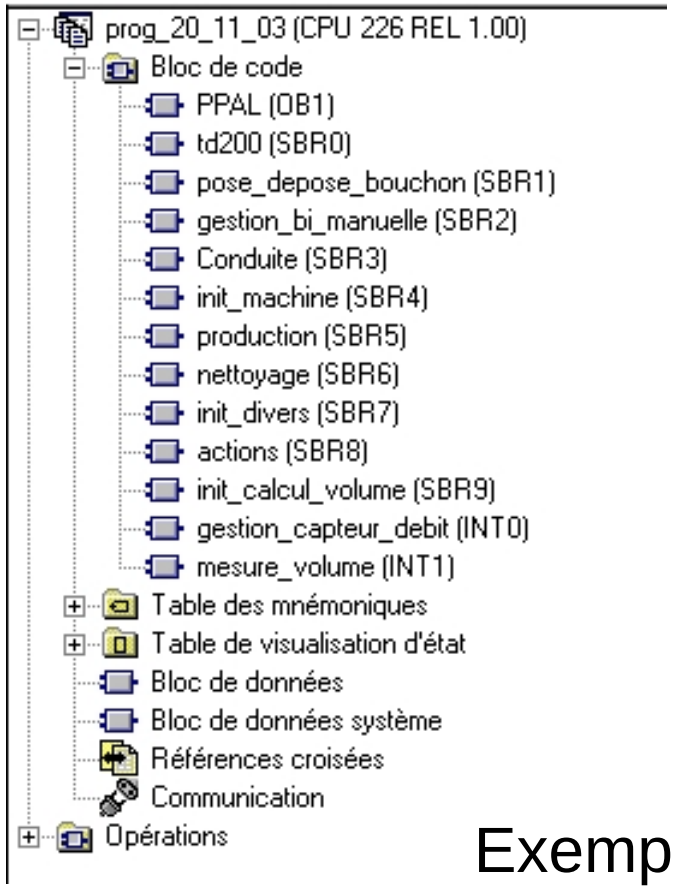
Comparaison des langages

LANGAGE	AVANTAGES	INCONVENIENTS
LD	facile à lire et à comprendre par la majorité des électriciens langage de base de tout PLC	suppose une programmation bien structurée
FBD	Très visuel et facile à lire	Peut devenir très lourd lorsque les équations se compliquent
ST	Langage de haut niveau (langage pascal) Pour faire de l'algorithmique	Pas toujours disponible dans les ateliers logiciels
IL	langage de base de tout PLC type assembleur	très lourd et difficile à suivre si le programme est complexe Pas visuel.
SFC	Description du fonctionnement (séquentiel) de l 'automatisme. Gestion des modes de marches Pas toujours accepté dans l'industrie…	Peu flexible

Multi-langages, multi-programmes !



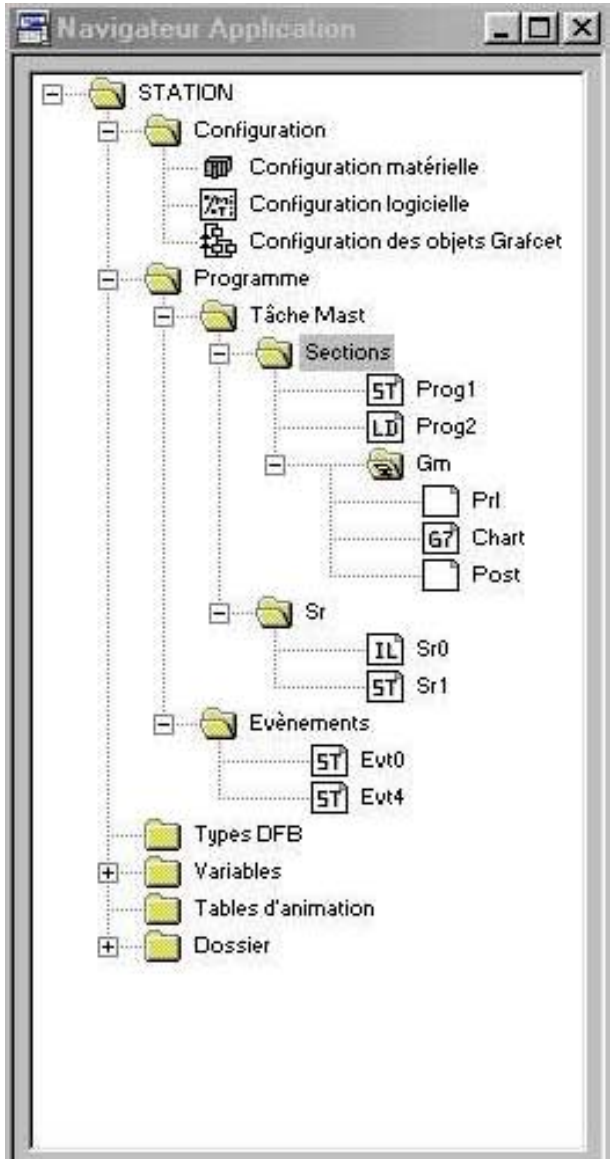
Exemple
Isagraf



Exemple
Siemens

Multi-langages, multi-programmes !

Exemple
Schneider



Avantages des API

évolutivité	très favorable au évolution. très utilisé en reconstruction d'armoire.
fonctions	assure les fonctions Conduites, Dialogue, Communication et Sûreté.
taille des applications	gamme importante d'automate
vitesse	temps de cycle de quelque ms
modularité	haute modularité. présentation en rack

Avantages des API

développement d'une application et documentation	très facile avec des outils de programmation de plus en plus puissant
architecture de commande	centralisée ou décentralisée avec l'apparition d'une offre importante en choix de réseaux , bus de terrain, blocs E/S déportées.
mise en oeuvre	mise au point rendu plus facile avec l'apparition des outils de simulation de PO
maintenance	échange standards et aide au diagnostique intégrée
portabilité d'une application	norme IEC 1131

Exemple (tendance micro)

FPO by MATSUSHITA AUTOMATION CONTROLS

vitesse de 0,9 µs/pas - scrutation cyclique

(possibilité en scrutation périodique)

programmation en langage LD et FDB et GRAFCET

EEPROM programme 5,4Ko (2720 pas) ou 10Ko (5000 pas)

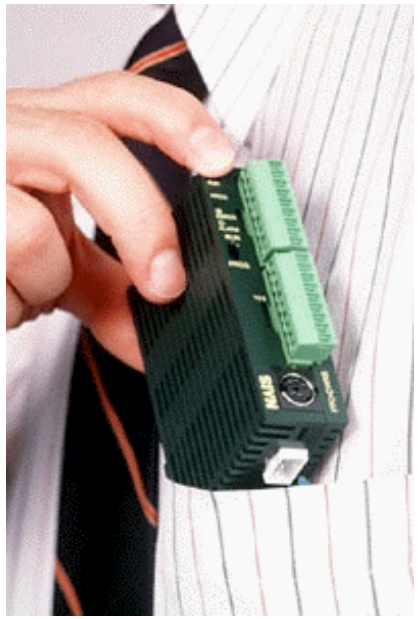
chien de garde - mise en réseau possible avec adaptateur - programmable en IEC 1131

E/S 6/4 8/6 8/8 16/16 et bientôt 128 – Analogique disponible

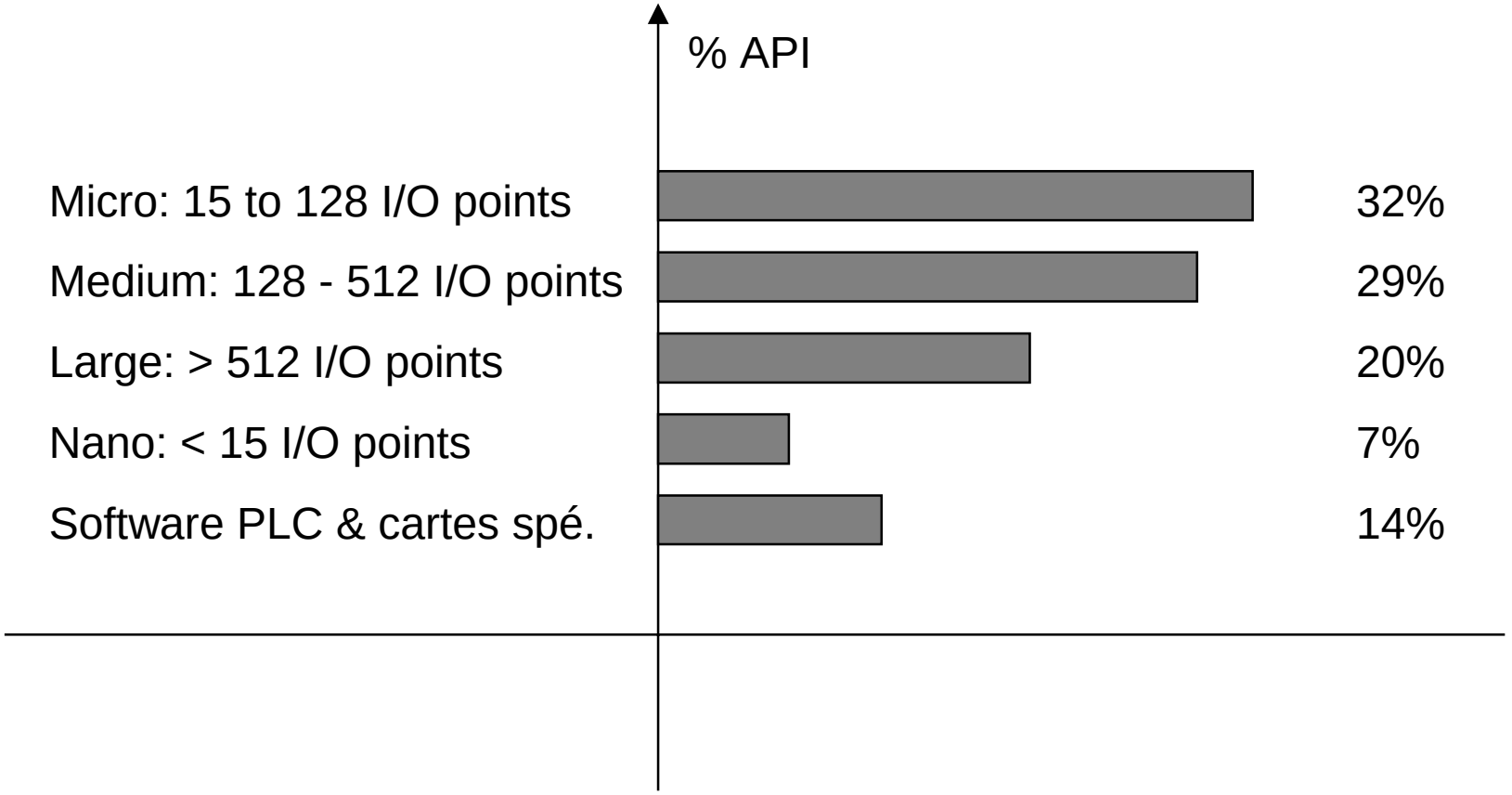
capture d'impulsion

interruption périodique programmable de 0,5 ms à 30 s

90x25x60 mm et moins de 100g !



Etat du marché



Source: Control Engineering, Reed Research, 2002-09

Les constructeurs

www.schneider-electric.com

SIEMENS

PHOENIX CONTACT

Honeywell

Allen-Bradley

Rockwell Automation

When good enough just isn't good enough

Weidmüller

HITACHI

Inspire the Next

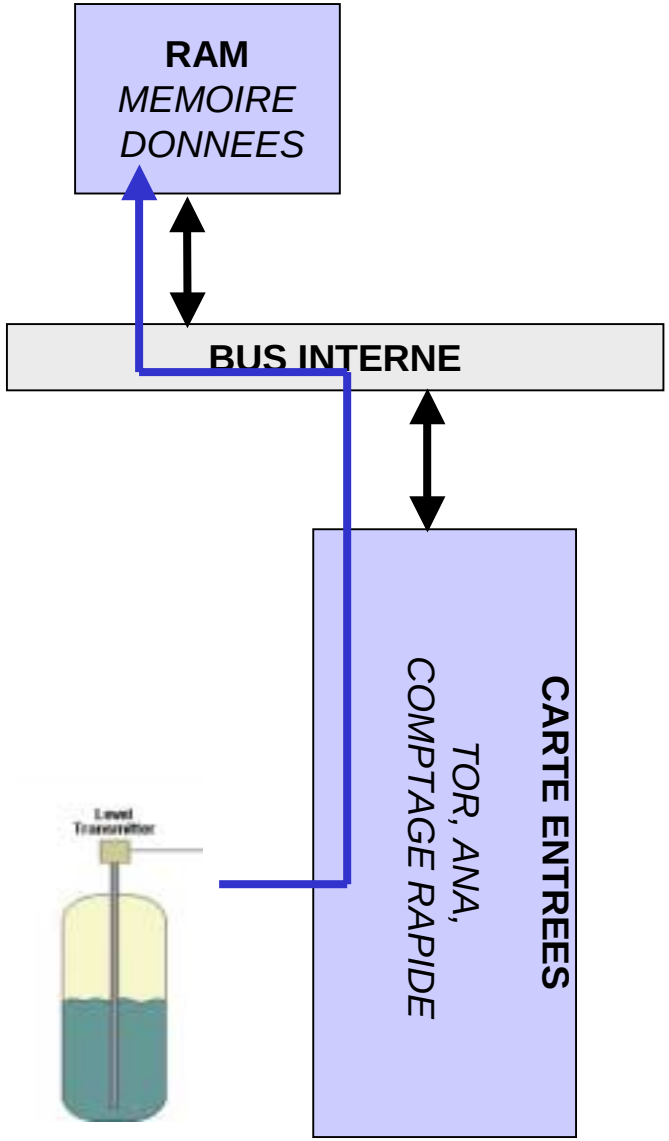
NAiS Matsushita Electric Works-Automation Controls Company

Acquisition des entrées

Acquisition des entrées

E

écriture en mémoire de l'état des informations présentes sur les entrées (réalise une image du monde extérieur)

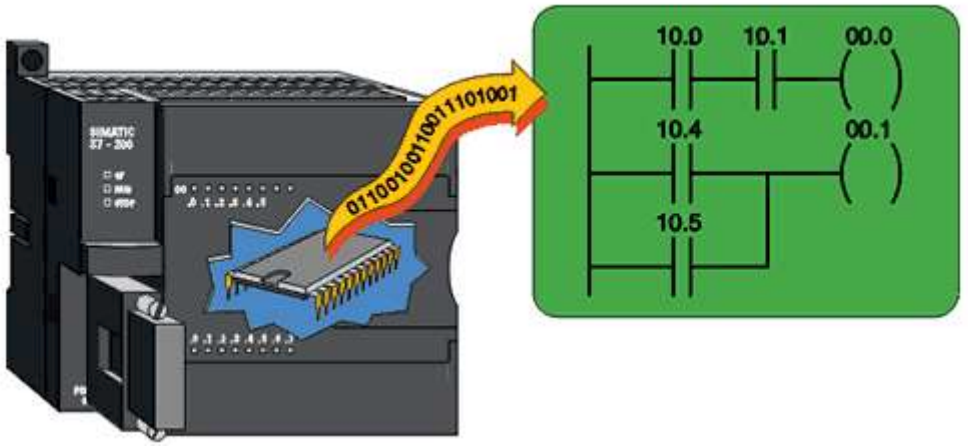


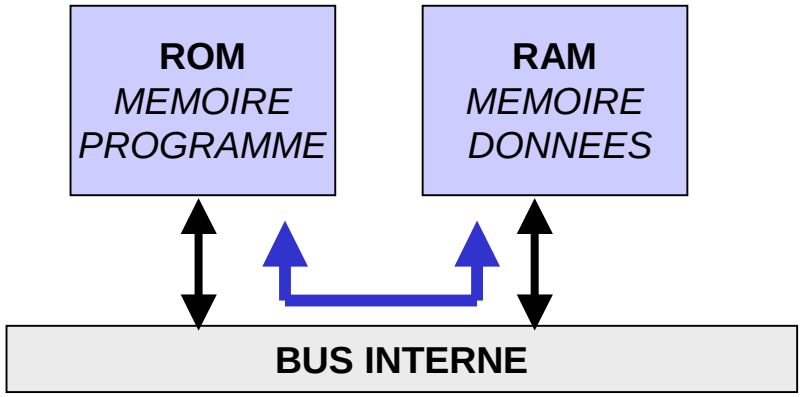
Traitement

Traitement du programme

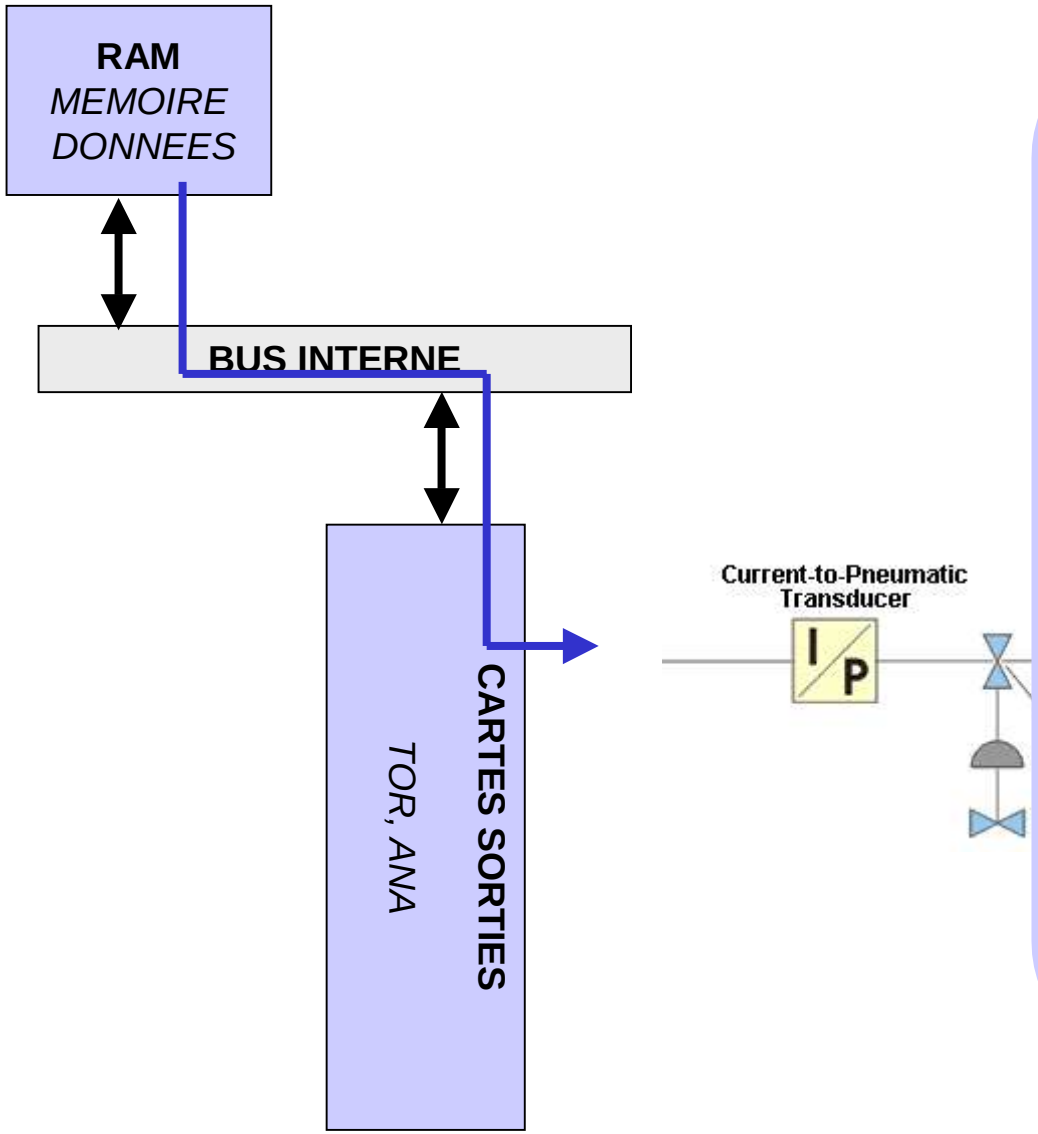
T

exécution du programme application, écrit par l'utilisateur.





Mise a jour des sorties

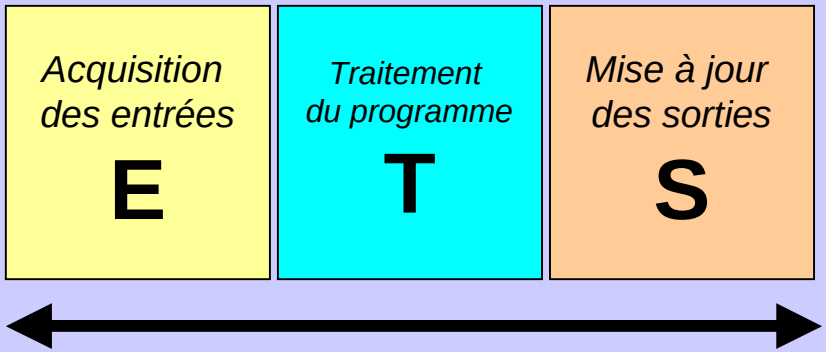


Mise à jour des sorties

S

écriture des bits ou des mots de sorties associés aux modules TOR et métier selon l'état défini par le programme application.

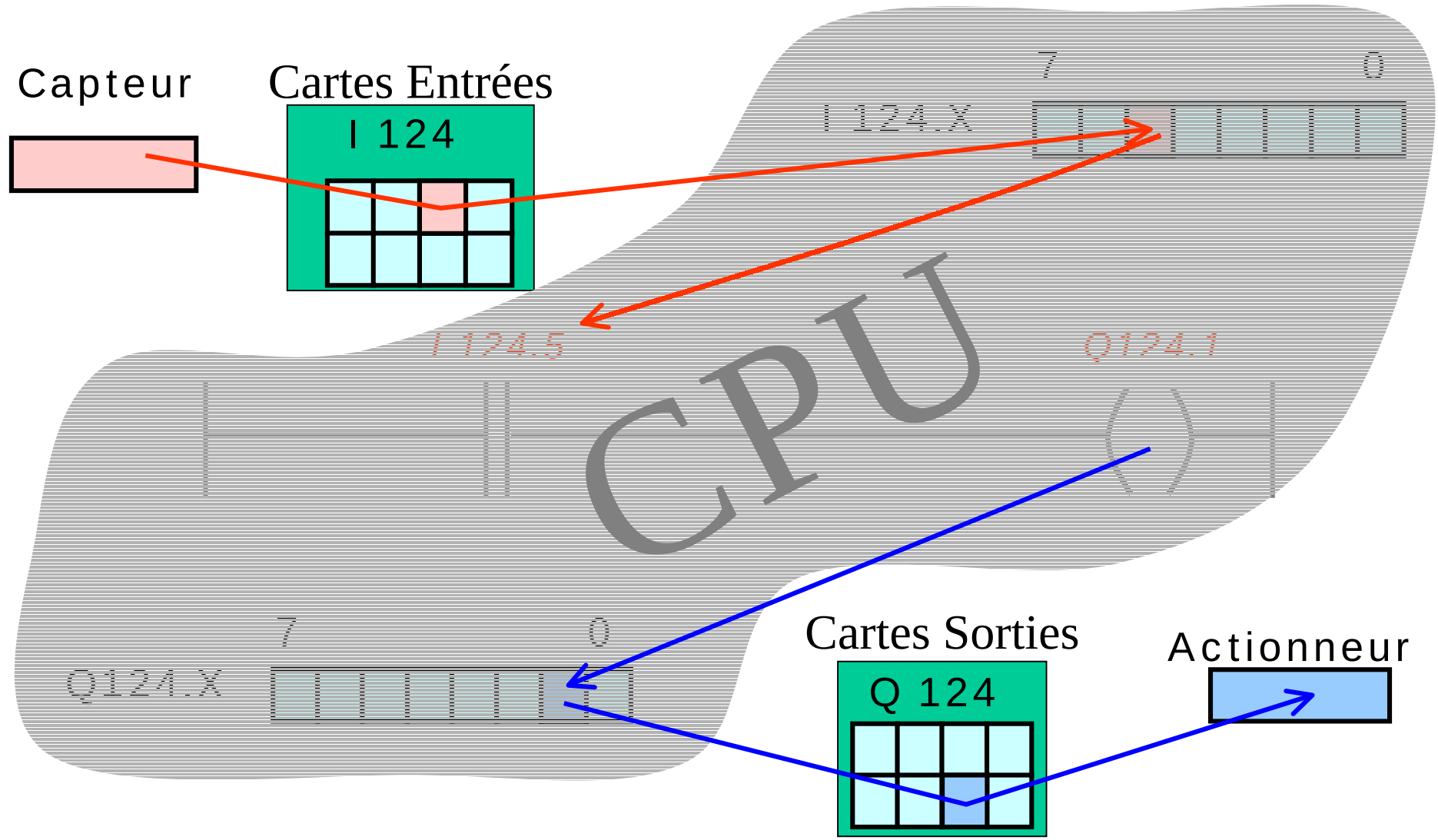
Tâche Automate



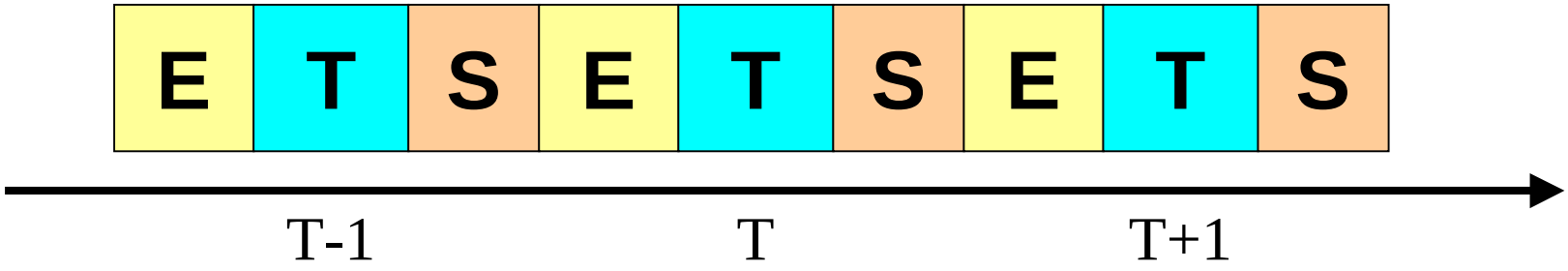
Temps de cycle

$T_C = T_E + T_T + T_S$

Exemple

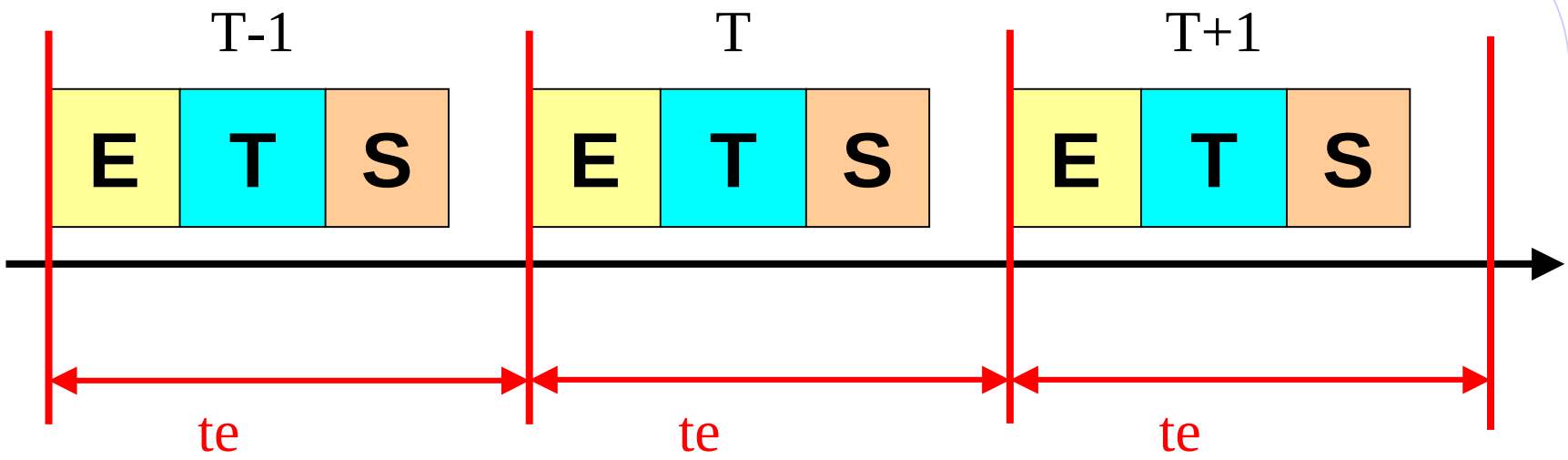

Capteur
Cartes Entrées
I 124
7
0
I 124.X
I 124.5
Q124.1
CPU
7
0
Q124.X
Cartes Sorties
Q 124
Actionneur

fonctionnement mono-tâche cyclique (asynchrone)



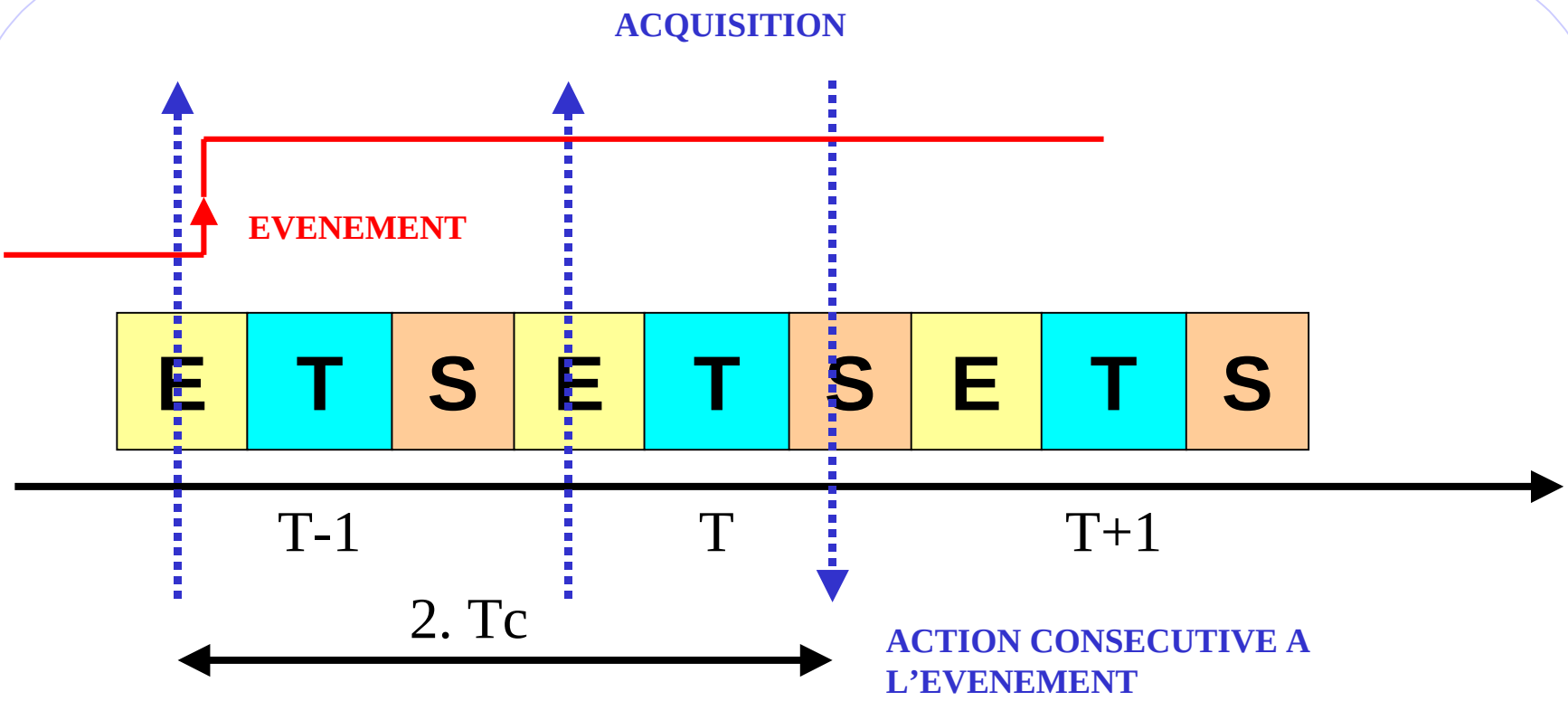
Ce type de fonctionnement consiste à enchaîner les cycles les uns après les autres.

fonctionnement mono-tâche périodique (synchrone)



Dans ce mode de fonctionnement, l'acquisition des entrées, le traitement du programme et la mise à jour des sorties s'effectue de façon périodique te ms selon un temps défini par configuration API .

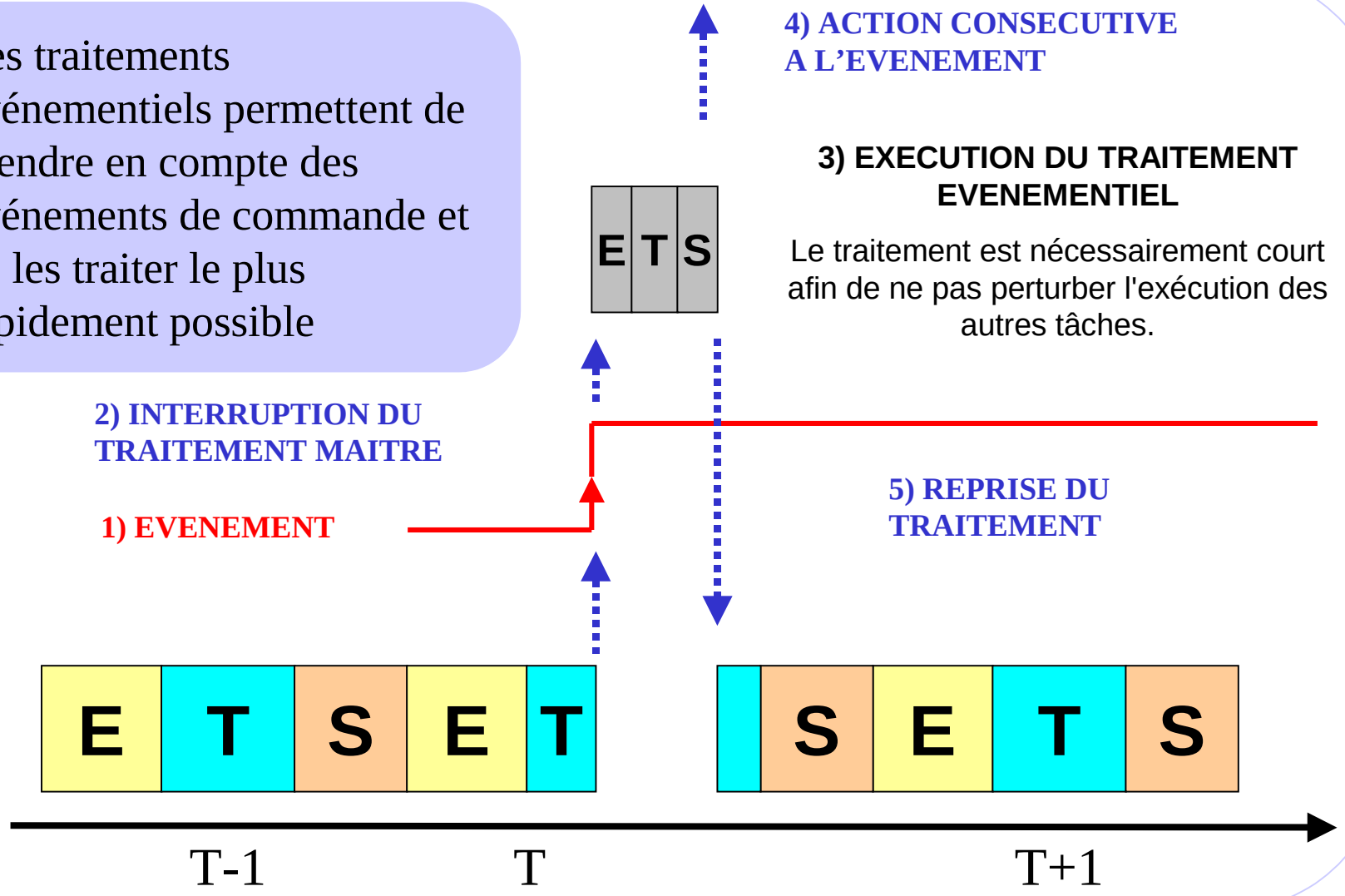
Retard dans le traitement de l'événement



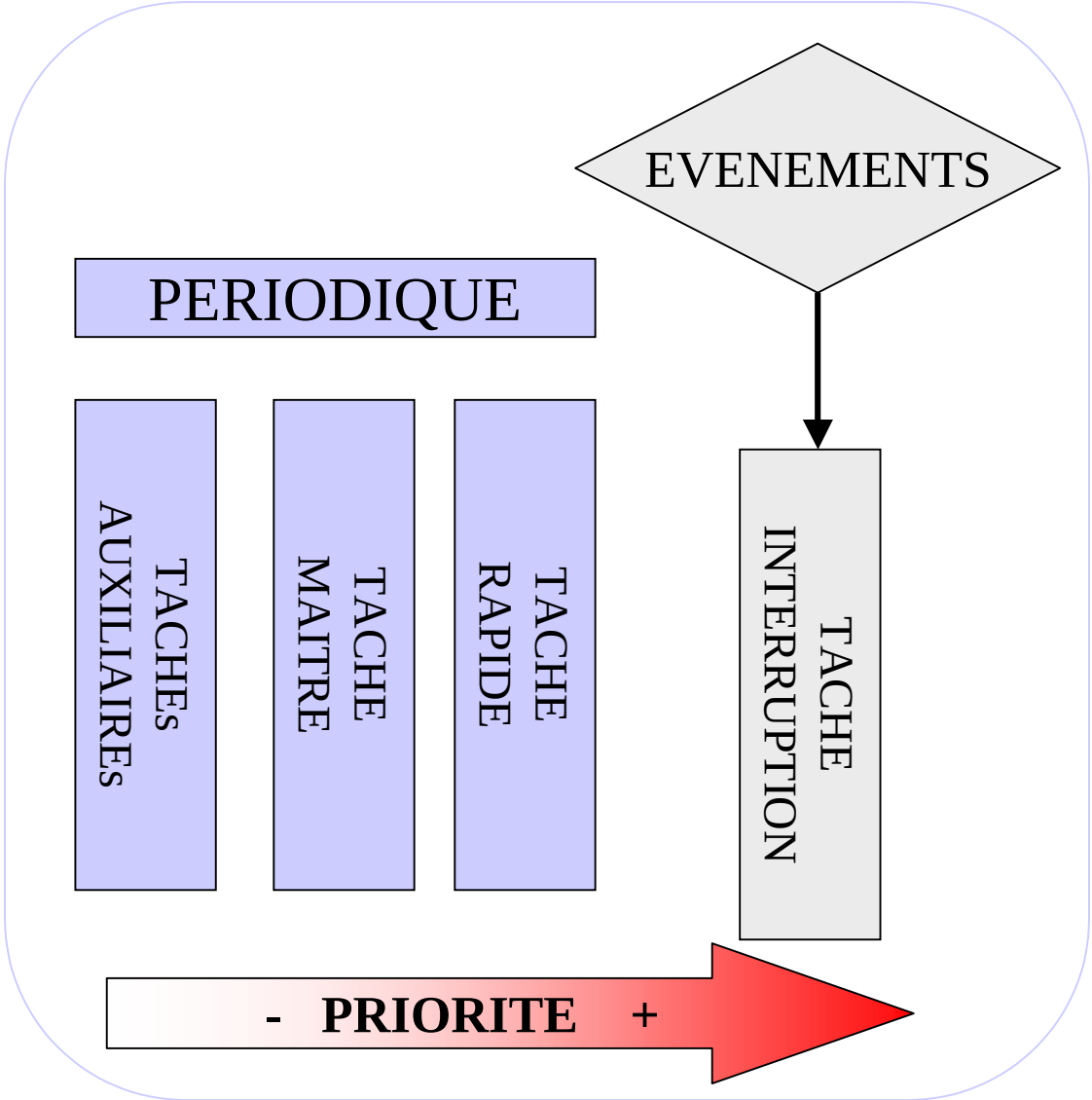
Les deux modes de traitements (cyclique ou périodique) sont appelé différé. Dans le pire des cas, il peut s 'écouler à peu prés 2 fois le temps de cycle moyen avant que l'UT réagisse à l'apparition d'un événement

Les interruptions

Les traitements événementiels permettent de prendre en compte des événements de commande et de les traiter le plus rapidement possible



Traitement multitâches



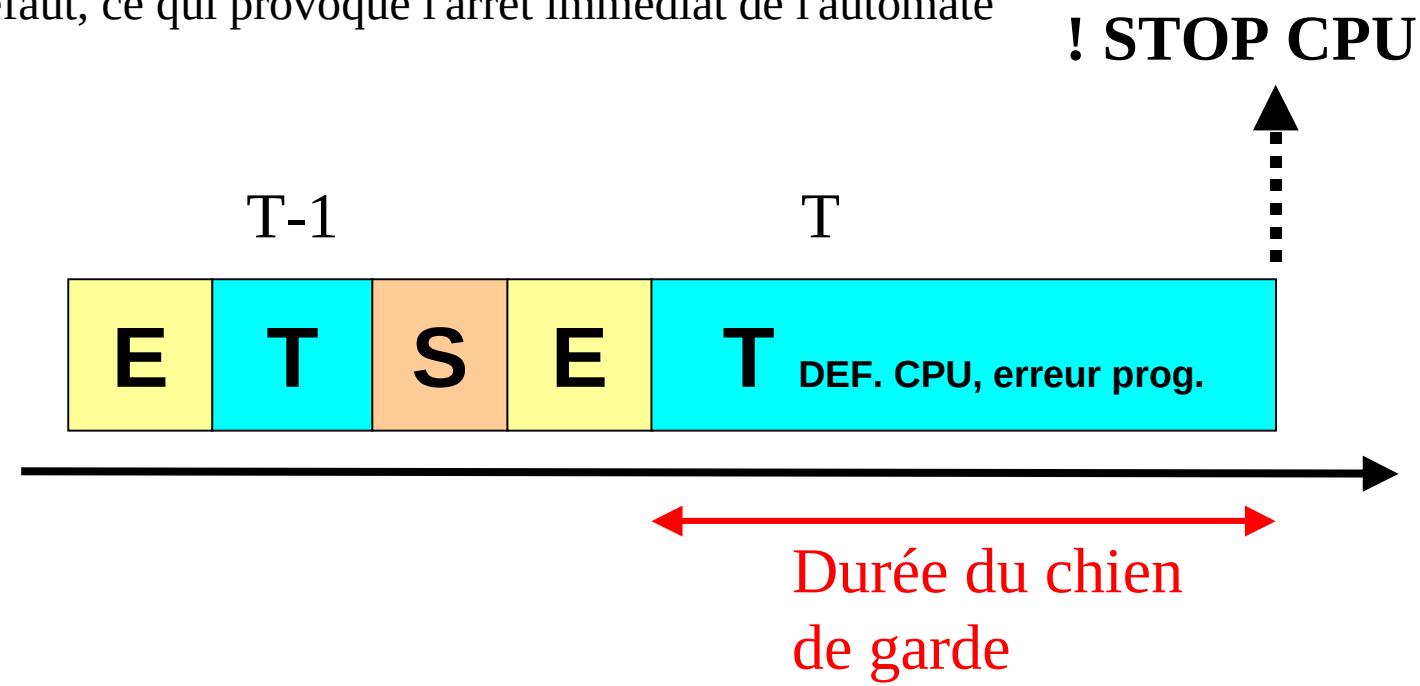
La tâche rapide permet d'effectuer des traitements courts avec une priorité plus élevée que dans la tâche maître

Le traitement est nécessairement court afin de ne pas perturber l'exécution des autres tâches

Chien de garde (watchdog)

La durée d'exécution de la tâche maître, en fonctionnement cyclique ou périodique, est contrôlée par l'automate (chien de garde) et ne doit pas dépasser la valeur définie en configuration

Dans le cas de débordement, l'application est déclarée en défaut, ce qui provoque l'arrêt immédiat de l'automate



Un contact est associé au watchdog