

LOISIRS TECHNIQUES D'AUJOURD'HUI

**hors série**

# Leed

# MICRO

# PROGRAMMATION

# COURS 2<sup>ème</sup> CYCLE

COURS  
**N°30**

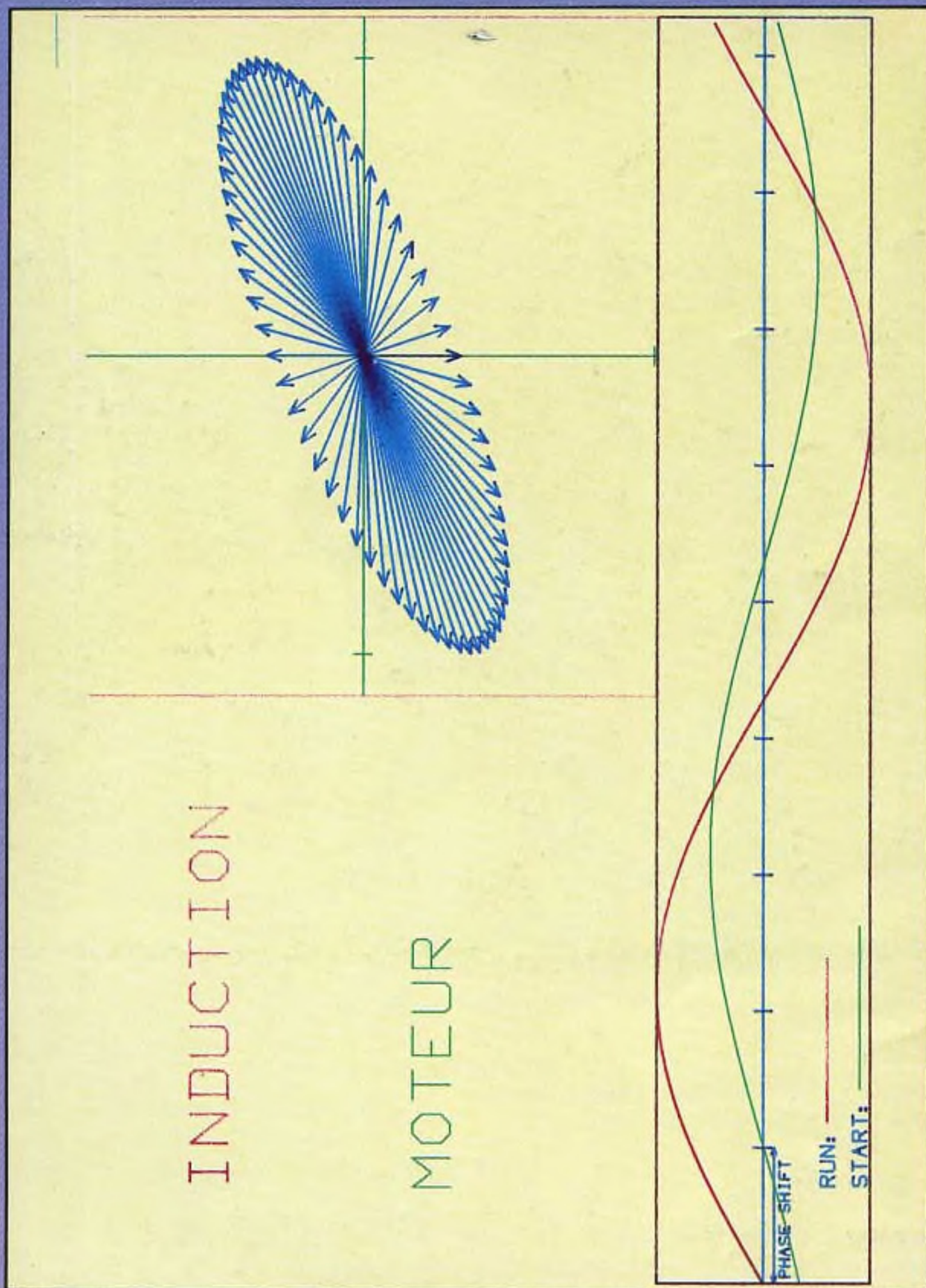
Suite  
2<sup>e</sup> cycle

**N°10**

COURS DE  
PASCAL  
forme générale  
d'un programme

COURS DE  
PROGRAM-  
MATION  
APPROFONDIE :  
les erreurs

COURS DE  
GENIE LOGICIEL :  
de la théorie  
à la pratique



ISSN 0757-6889

# VOYAGE AU COEUR DES MICRO-ORDINATEURS

dans la  
**COLLECTION**  
**«ETUDES»**  
aux  
éditions  
fréquences



**une véritable schémathèque**

- 128 pages
  - 101 schémas
  - 34 tableaux
- Prix : 150 F

Que ce soit pour concevoir des interfaces ou optimiser un programme (utilisation des périphériques, encombrement mémoire...) «un micro-informaticien performant» doit posséder une bonne connaissance de son matériel.

Ce livre s'adresse donc à tous les électroniciens qui désirent découvrir les différents

composants constituant un micro-ordinateur. Articulé autour du microprocesseur Z80, cet ouvrage contient de nombreux schémas (plan mémoire, interfaces série et parallèle, interface clavier, interface vidéo, CAN, CNA...) qui pourraient être le thème... de nouvelles extensions.

En vente chez votre libraire et aux Editions Fréquences

## BON DE COMMANDE

Je désire recevoir l'ouvrage **L'électronique des micro-ordinateurs** au prix de **160 F** (150 F + 10 F de port).

Nom .....

Adresse .....

A adresser aux **EDITIONS FREQUENCES 1 boulevard Ney, 75018 Paris**

Règlement ci-joint :

Par chèque bancaire  par chèque postal  par mandat

*Philippe Faugeras, Docteur-ingénieur en électronique a acquis son expérience dans de grandes entreprises françaises où pendant cinq ans, il a travaillé sur des systèmes d'automatismes à base de microprocesseurs. Philippe Faugeras est responsable de la rubrique «Raconte-moi la micro-informatique» dans la revue LED.*

LO... MOUES D'AUJOURD'HUI

**hors série**

# LED

# MICRO

## PROGRAMMATION COURS 2<sup>e</sup> CYCLE

**Société éditrice :**  
**Editions Fréquences**  
 Siège social :  
 1, bd Ney, 75018 Paris  
 Tél. : (1) 46.07.01.97 +  
 SA au capital de 1 000 000 F  
 Président-Directeur Général :  
 Edouard Pastor

**LED MICRO**  
 (cours 2<sup>e</sup> cycle)  
 Mensuel : 18 F  
 Commission paritaire : 64949  
 Directeur de la publication :  
 Edouard Pastor

Tous droits de reproduction réservés  
 textes et photos pour tous pays  
 LED MICRO est  
 une marque déposée ISSN 0757-6889

**Services Rédaction-Publicité-  
 Abonnements :**  
 1, bd Ney, 75018 Paris  
 Tél. : (1) 46.07.01.97  
 Lignes groupées

**Comité de rédaction :**  
 Dominique Chastagnier  
 Jean-François Coblentz  
 Charles-Henry Delaleu  
 Patrick Gueneau

Secrétaire de Rédaction  
 Chantal Cauchois

**Publicité, à la revue**  
 Tél. : 607.01.97  
 Secrétaire responsable  
 Annie Perbal

**Abonnements**  
 10 numéros par an  
 France : 160 F  
 Etranger : 240 F

**Réalisation**  
 Composition-Photogravure  
 Edi Systèmes  
 Impression  
 Berger-Levrault - Nancy



MAI 86

### COURS DE PASCAL de la page 5 à la page 17

- Une première approche ..... p. 6
- Un peu plus de détails ..... p. 8
  - Déclaration du programme
  - Déclaration des étiquettes
  - Déclaration des constantes
  - Les déclarations de types
  - Les déclarations de variables
  - Conclusion sur les déclarations de paramètres
  - Déclaration des procédures et fonctions
- Conclusion

**Dominique Chastagnier  
 Jean-François Coblentz  
 Patrick Gueneau**

### COURS DE PROGRAMMATION APPROFONDIE

Les erreurs  
 de la page 20 à la page 28

**NOTRE COUVERTURE :** En associant un ordinateur scientifique à différents capteurs, il est possible de réaliser un diagnostic sur les caractéristiques d'un moteur, de manière automatique.

- Votre programme ne tourne pas p. 21
  - Interprétation et compilation
  - Les erreurs immédiates simples
  - Les erreurs en cascade
  - Les erreurs différées
- Préparation aux erreurs ..... p. 23
  - Le routage
  - L'affichage
  - Précautions d'emploi
  - Les tests
- Les messages d'erreurs ..... p. 24
- Exercices d'applications ..... p. 27
  - Les carrés magiques
  - La communauté économique européenne

**Dominique Chastagnier  
 Jean-François Coblentz  
 Patrick Gueneau**

**C'EST ARRIVÉ DEMAIN  
 de la page 32 à la page 34**

### COURS DE GENIE LOGICIEL L'informatique industrielle de la page 35 à la page 49

- Systèmes en informatique industrielle ..... p. 36
- L'informatique industrielle aujourd'hui ..... p. 37
- Les premiers systèmes d'informatique industrielle ..... p. 38
- Les interruptions ..... p. 39
- Conclusion ..... p. 40
- Liste des principaux capteurs .. p. 41
- Informatique industrielle ..... p. 42
- Principe de base en informatique industrielle ..... p. 43
- Calculateur en informatique industrielle ..... p. 44
- Une carte d'entrées-sortie .... p. 46
- Convertisseur analogique-digital ..... p. 48
- Code BCD, afficheurs  
 7 segments ..... p. 49

**Charles-Henry Delaleu**

# Microprocesseurs un cours essentiellement pratique !



Philippe Duquesne, ingénieur électronicien (I.S.E.N.) est chargé du cours de microprocesseurs au C.N.A.M. de Paris. Depuis plus de dix ans, il a pris goût à l'enseignement et il est l'auteur d'un ouvrage didactique sur l'électronique digitale et notamment d'un cours pratique de microprocesseurs. Fervent pratiquant du « dialogue » école/industrie, après avoir exercé les fonctions de chef de département électronique chez Burroughs, second constructeur mondial en informatique, il est actuellement chef du service Etudes Electroniques au sein de la direction technique chez Messier Hispano Bugatti (groupe SNECMA) avec, pour principal objectif l'introduction des microprocesseurs dans les trains d'atterrissage.

**Pour ceux qui veulent aborder la micro-informatique en désirant en connaître les éléments essentiels ; ceux pour qui la « puce » ne doit pas rester un mythe.**



## Electronique digitale ?

**Notre temps aura témoigné d'une nouvelle technique, une autre façon de communiquer avec l'électronique digitale. Philippe Duquesne, professeur chargé de cours au CNAM, a su dans cet ouvrage en expliquer clairement les fondements.**



Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

**Bon de commande** à adresser aux EDITIONS FREQUENCES 1, bd Ney 75018 PARIS

Je désire recevoir le(s) ouvrage(s) suivant(s) :

- INITIATION A L'ELECTRONIQUE DIGITALE au prix de 105 F (95 F + 10 F de port).  
 INITIATION AUX MICROPROCESSEURS au prix de 105 F (95 F + 10 F de port).

Ci-joint mon règlement par :  CCP  Chèque bancaire  Mandat

Nom ..... Prénom .....

Adresse .....

Code postal ..... Ville .....

# COURS DE PASCAL

Dominique Chastagnier  
Jean-François Coblentz  
Patrick Gueneau

Attention, ce cours est dangereux. Si vous commencez à le lire, il est vraisemblable que vous voudrez programmer votre ordinateur en Pascal. Vous mettez ici le doigt dans un engrenage fatal et il ne sera pas possible de vous soigner du virus qui vous guette. Si cette perspective vous inquiète, passez ce cours sans remords. Vous aurez été prévenu.

Si vous restez avec nous voilà, comme promis, le premier cours de programmation en Pascal. Nous allons y décrire la forme générale d'un programme et le maître-mot sera structure. Ne l'oubliez pas, car c'est tout le Pascal en un mot.

## COURS N° 1

### PLAN DU COURS

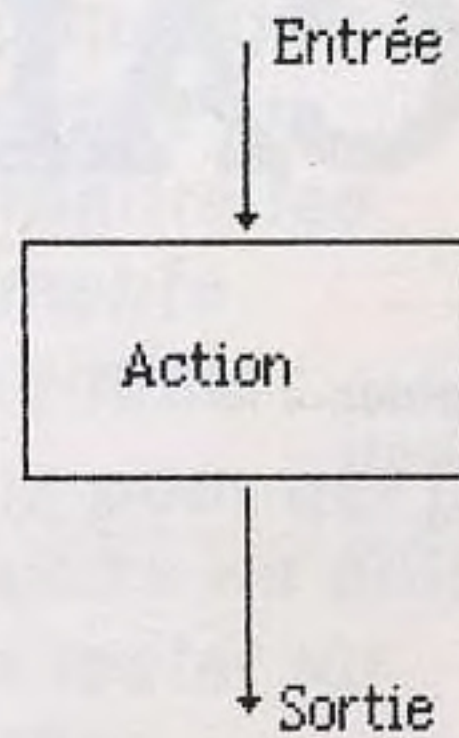
1. Une première approche
2. Un peu de détails
  - 2.1. Déclaration du programme
  - 2.2. Déclaration des étiquettes
  - 2.3. Déclaration des constantes
  - 2.4. Les déclarations de types
  - 2.5. Les déclarations de variables
  - 2.6. Conclusion sur les déclarations de paramètres
  - 2.7. Déclaration des procédures et fonctions
3. Conclusion

## I. UNE PREMIERE APPROCHE

Contrairement au Basic, le Pascal est un langage structuré. Cela signifie que toute partie d'un programme Pascal est incluse dans une structure. Ces structures sont de deux types :

- déclarations
- instructions.

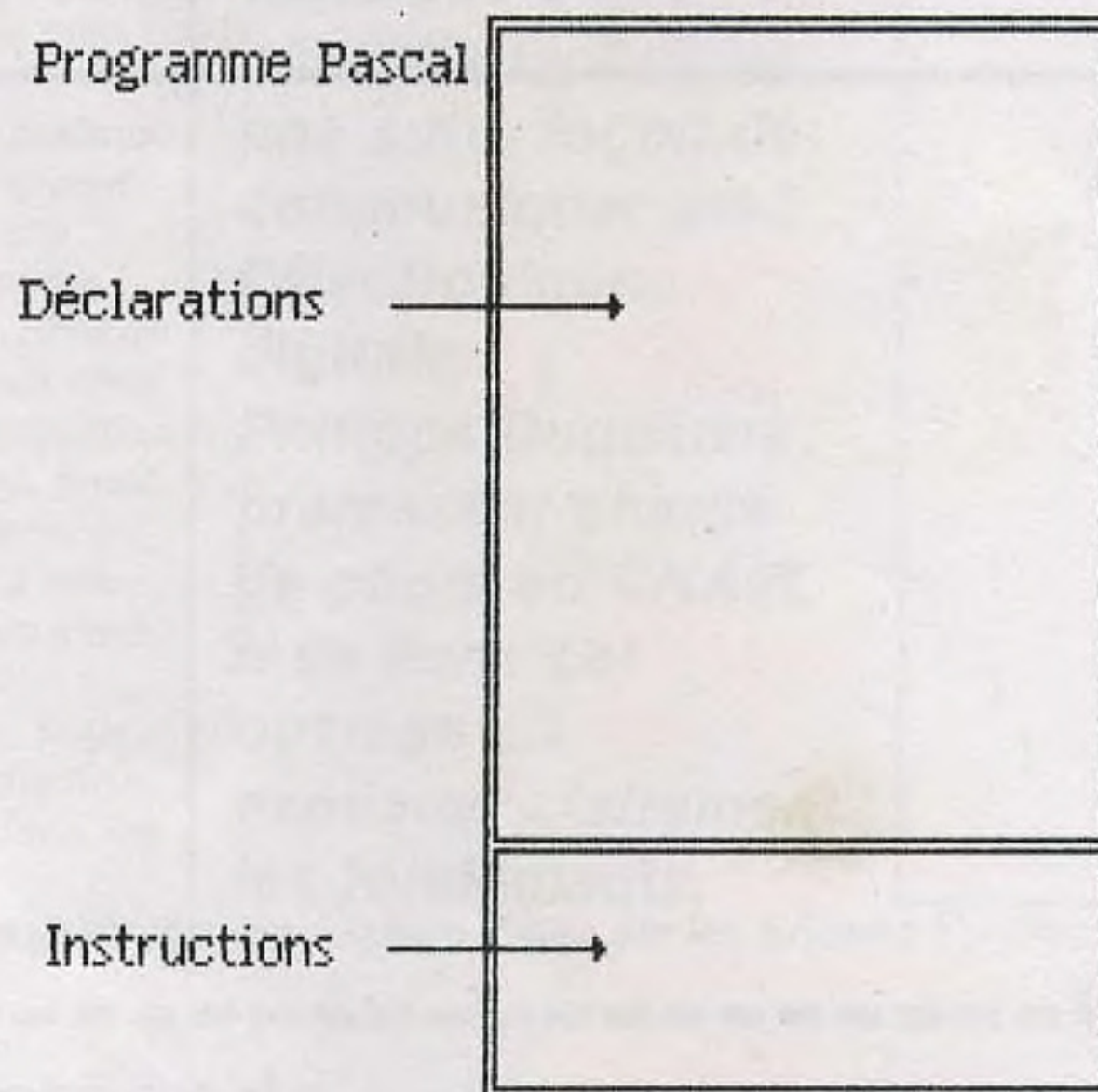
Avec Pascal, on passe d'une structure à une autre en permanence. Ainsi, chaque action est un être à part entière.



La structure la plus générale est la suivante :

1. bloc déclarations
2. bloc instructions.

Cela peut se visualiser de la façon suivante :



Prenons un premier exemple de programme, pour bien voir le principe. Le programme suivant affiche le nombre que vous lui proposez.



- Avant un ELSE, pour des raisons que nous vous expliquerons plus tard dans l'étude des blocs IF.
- Après les mots réservés CONST, TYPE, VAR, BEGIN, LABEL, car ils se suffisent à eux-mêmes, et le programme sait qu'après ces mots, l'instruction se termine.

Avant de passer à la description détaillée des différents blocs, signalons que les commentaires sont situés n'importe où dans le programme, la syntaxe est la suivante :

```

.....
(* inserer votre commentaire ici *)
.....

```

Il peut aussi être à cheval sur plusieurs lignes :

```

.....
(* inserer votre commentaire
ici *)
.....

```

Si votre clavier vous autorise l'accès aux caractères ( et ), ils remplaceront avantageusement les symboles (\* et \*) (respectivement).

## II. UN PEU PLUS DE DETAILS

Nous allons maintenant approfondir la description de la forme du programme. La partie instructions ne demande pas plus de détails, mais la partie déclaration en demande. Une partie déclarative est structurée de la manière suivante :

déclaration du programme
déclaration des étiquettes
déclaration des constantes
déclarations des types
déclaration des variables
déclaration des procédures et fonctions



Reprenons chaque partie maintenant.

### 2.1. Déclaration du programme

Pour l'instant, nous considérerons qu'il s'agit simplement du mot **program** suivi d'un nom de votre choix. Dans le court exemple donné plus haut, il s'agit de la ligne :

**Program Essai;** (\* attention aux ; dans les programmes Pascal \*)

C'est toujours la première ligne (non commentaire) d'un programme. A travers elle, vous indiquez le nom du programme et vous envisagez de lui imposer des entrées (données que vous lui donnerez en cours de programme) ou des sorties (résultats qu'il devra sortir sur écran, sur imprimante, ou autres). Cette dernière série d'informations se résume à ajouter les mots **INPUT** et **OUTPUT** dans la première ligne, de la manière suivante :

**Program Essai(INPUT,OUTPUT);**

Dorénavant, nous ajouterons toujours cette information, car cela représente un faible travail, et facilite la vie en cas de transferts interactifs avec le programme.

Un dernier détail important : le nom du programme doit commencer par une lettre. Pour la suite, c'est à votre discrétion (lettre, chiffre ou le tiret souligné «\_»). Par exemple :

**A1**

**a\_1**

**Ceci\_est\_un\_long\_nom** sont des noms de programmes.

**1\*\_prog**

n'en sont pas.

**&Hello**

Ces règles restent valables pour tous les noms que vous définirez (noms de type, de constantes, de variables, de procédures ou de fonctions, etc.).

### 2.2. Déclaration des étiquettes

Ceux qui ont suivi le cours de Basic savent qu'une étiquette est le symbole, chiffre(s) ou lettre(s), qui permet d'indiquer à un GOTO où se rendre. Or, la première chose à savoir en Pascal est qu'un GOTO est ce qu'il y a de pire, et qu'il est toujours possible de l'éviter. Oubliez donc cette section, car nous ne reparlerons plus jamais de GOTO, sinon pour vous encourager à ne pas l'utiliser, comme nous le faisons à la fin de ce cours.

Nous abordons maintenant un point très important du Pascal.

**Toute variable, paramètre, nom... utilisés doivent être annoncés au programme dans les parties déclaratives qui suivent.**

### 2.3. Déclaration des constantes

Une constante est une quantité qui ne changera pas de valeur, de contenu, lors de l'exécution du programme. (Si nous sommes à la limite du pléonasme, c'est que nous travaillons sans filet !!). Il y a deux façons complémentaires de voir l'utilisation des constantes :

- Si vous voyez dans un programme des entités qui ne changent pas de valeurs, mettez-les en partie constante.

- Si des entités déclarées en constante sont modifiées dans le programme, les résultats seront dépendants de l'implémentation du Pascal sur votre machine (c'est-à-dire de la façon dont le langage a été adapté pour votre ordinateur) et, de toutes façons, imprévisibles.

L'expérience montre que l'utilisation de constantes facilite bien la lecture d'un programme. En effet, lorsque constante est déclarée, vous savez, en lisant le programme, que vous n'avez pas à vous préoccuper de traquer ses changements successifs, ce que vous ne pouvez pas faire en Basic où tout peut changer à tout moment.

Des exemples de constantes :

```
nb_mois=12;          (* à noter la différence entre la déclaration *)
pi=3.1415926;      (* de la valeur d'une constante (signe = )
```

avec l'affectation d'une valeur à une variable (signe := ) \*)

Vous pouvez voir que ces entités ne changent que rarement de valeur, que ce soit le nombre de mois par an ou le nombre PI. Ils sont restés assez stables à Wall Street ces derniers temps.

Prenons un exemple :

```
Program ex_const(output);          (* pas de ; car exception *)
const
  pi=3.1415926;
  deux=2;
begin                                (* pas de ; car exception *)
  writeln(' deux fois PI =',deux*pi) (* pas nécessairement de ; *)
end.                                (* car END suit *)
```

Ici, nous nous contentons de multiplier deux constantes, deux et pi. Output précise qu'il y aura une sortie interactive de message.

#### 2.4. Les déclarations de types

Il s'agit là d'une nouveauté complète pour vous, et peut-être la trouverez-vous quelque peu artificielle. Mais c'est une aide à la programmation qui n'a pas de prix. Il s'agit ni plus ni moins de pouvoir créer toute structure que vous utilisez régulièrement, sans avoir à la redéfinir à chaque fois. Prenons tout de suite un exemple comparé avec le BASIC. Supposons que vous déclariez en Basic trois tableaux 5x5. Vous feriez :

**100 DIM A(5,5) : DIM B(5,5) : DIM C(5,5)**

Maintenant, en Pascal, dans partie type :

**Tableau = array[1..5,1..5] of real;** (\* array signifie tableau \*)

Ceci est un exemple extrêmement limité.

Ici, nous avons décrit un type ami, qui regroupe les amis de quelqu'un. Il sera par la suite possible de travailler sur cet ensemble ou sur des sous-ensembles de cet ensemble.

Pour résumer, dans cette déclaration-type, il est possible de présenter toutes les structures de données qui seront utilisées par le programme.

Ce qu'il faut bien comprendre, c'est que par ces déclarations, vous prévenez le système de vos besoins ultérieurs, lui permettant de travailler le plus efficacement au niveau du stockage de l'information.

Un certain nombre de types sont prédéfinis en Pascal, ce qui signifie que vous pouvez les utiliser sans les décrire. La liste ci-dessous ne présente que le standard Pascal ; suivant les compléments que votre version du langage peut apporter, d'autres types peuvent être prédéfinis (par exemple LONGINT pour des entiers longs, BYTE pour les octets, STRING pour les chaînes de caractères, etc.) :

- integer les entiers numériques tels que vous les avez pratiqués déjà (entre - 32 768 et + 32 767 le plus souvent),
- real les nombres réels (dont la précision équivaut à la simple précision en BASIC),
- char type représentant les lettres, et tous les autres caractères,
- boolean les variables booléennes sont du type VRAI, FAUX (**true, false**).

A partir de ces types, libre à vous d'en décrire d'autres, selon vos besoins et votre fantaisie. Par exemple, un type interval peut être :

**interval = 1..5;**

qui est un type décrivant les entiers entre 1 et 5 (le signe .. indique que l'on a affaire à un intervalle).

Un autre exemple :

**jours = (lundi,mardi,mercredi,jeudi,vendredi,samedi,dimanche)**

qui décrit les jours de la semaine (la virgule servant de simple séparateur) et à partir duquel nous proposons des sous-types :

**certains\_jours = (mardi,samedi);**

**autres\_jours = lundi..jeudi;**

**jour\_ouvrable=lundi..vendredi;**

**week\_end=samedi..dimanche;**

Le premier sous-type est formé par les deux jours indiqués, alors que le second est formé par les quatre jours allant de lundi à jeudi, les deux autres exemples se comprennent aisément. Cette façon de faire est très performante.

Ce dont il faut se souvenir, c'est que toute variable et toute constante en Pascal dépendent d'un type. Il y a quelques exceptions, mais très rares.

## 2.5. Les déclarations de variables

Une variable est un paramètre susceptible de changer de valeur à tout moment. Ce paramètre doit être déclaré, et doit appartenir à l'un des types décrits par le programme, ou prédéfinis par Pascal. Un exemple reprenant les types décrits précédemment entre autres :

### type

```
jours=(lundi,mardi,mercredi,jeudi,vendredi,samedi,dimanche);
certains_jours = (mardi,samedi);
autres_jours = lundi..jeudi;
```

### var

```
x,y,z : integer;
j1 : autres_jours;
```

Nous avons indiqué au programme que x, y et z sont des entiers, et que j1 est une variable prenant ses valeurs dans l'ensemble JOURS. Si par la suite, vous essayez de lui faire prendre une valeur comme SAMEDI, le programme sera stoppé par une erreur de type dépassement d'intervalle (OUT OF RANGE). On n'est plus respecté !!!

## 2.6. Conclusion sur les déclarations de paramètres.

2.6.1. Nous avons vu jusque là comment préciser au programme ce à quoi il doit s'attendre. Voilà un exemple complet d'une partie déclarative de paramètres :

```
Program calendrier(input,output);
```

### const

```
nb_jours=7;
nb_mois=12;
nb_heures=24;
```

### type

```
tab_jours = array [1..nb_jours] of integer;
jours= (lundi,mardi,mercredi,jeudi,vendredi,samedi,dimanche);
certains_jours = (mardi,samedi);
autres_jours = lundi..jeudi;
```

### var

```
x,y,z : integer;
j1 : autres_jours;
```

Vous pouvez remarquer que nous avons combiné des déclarations, dans le cas où nous nous servons de la constante nb\_jours pour déclarer tab\_jours, et dans le cas où nous utilisons autres\_jours pour j1.

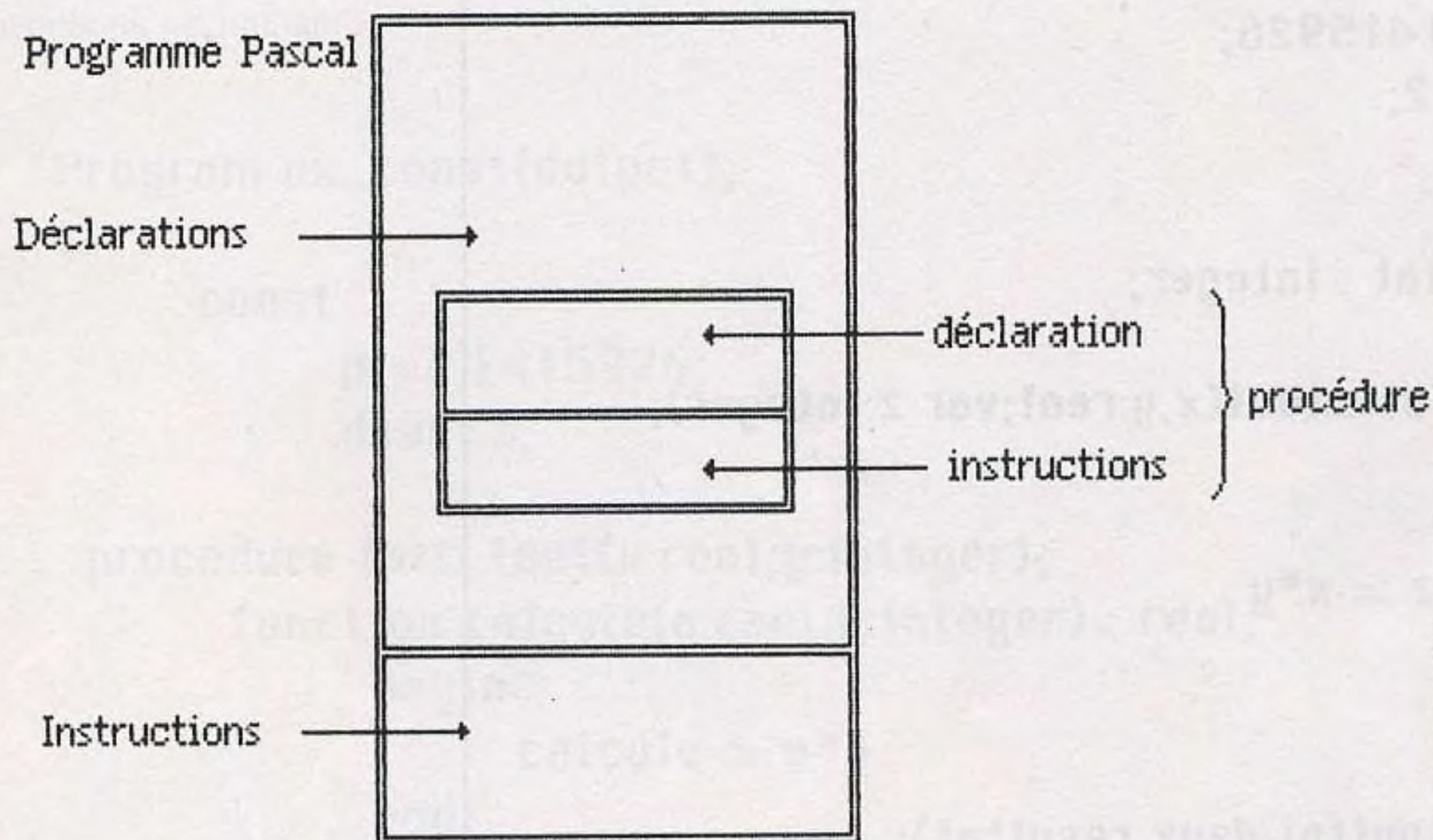
**2.6.2.** Contrairement au Basic, en Pascal les noms des paramètres contiennent plusieurs lettres significatives. En général, il y en a huit minimum, certains systèmes en donnant autant que vous en voulez, c'est-à-dire que si vous donnez un nom de 100 lettres, les 100 sont conservées. Attention à la place de stockage.

## 2.7. Déclaration des procédures et fonctions

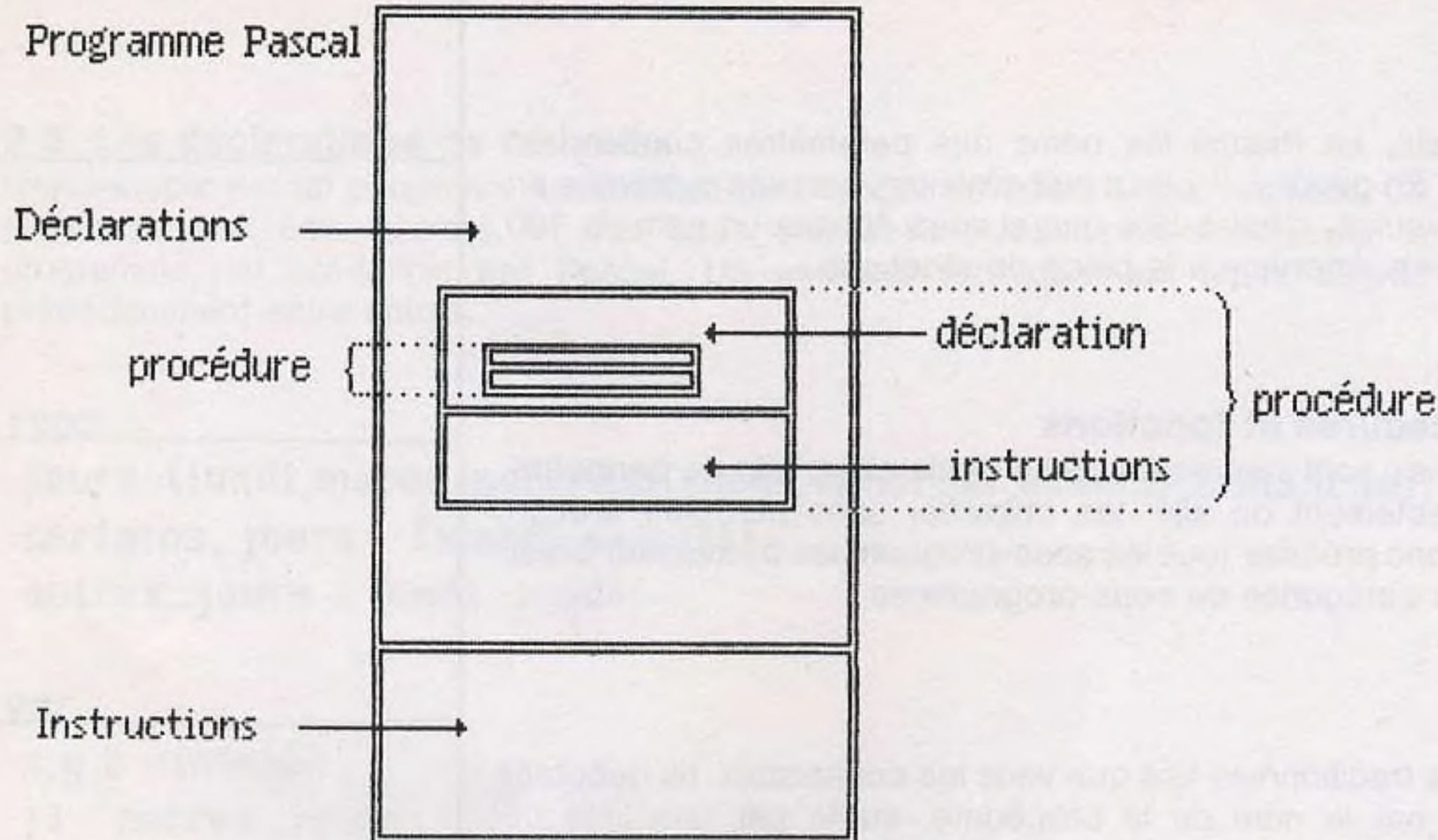
En Pascal, les sous-programmes sont insérés en partie déclaration, afin de permettre au programme de savoir directement où aller les chercher sans parcourir tout le programme pour cela. Il faut donc préciser tous les sous-programmes à l'avance. C'est ça la structuration. Il y a deux catégories de sous-programmes :

### 2.7.1. Les procédures

Il s'agit des sous-programmes traditionnels tels que vous les connaissez. Ils débutent par le mot **procédure** suivi par le nom de la procédure, suivie par une liste de paramètres utilisés par cette procédure. Pour le reste, il s'agit ni plus ni moins d'un programme dans le programme, avec sa zone de déclarations et sa zone d'instructions. Pour reprendre un schéma vu au début, voici la place d'une procédure dans un programme :



Il va sans dire que cette notion de procédure dont nous venons de dire qu'elle avait tout d'un programme dans un programme, peut s'imbriquer autant que vous le désirez, avec des procédures dans la procédure. Ceci donnerait le schéma :



Prenons un exemple d'utilisation de sous-programmes :

### **Program ex\_const(output);**

```

const
    pi=3.1415926;
    deux=2;

var
    resultat : integer;

procedure fait_tout(x,y:real;var z:integer);

    begin
        z := x*y
    end;

begin
    fait_tout(pi,deux,resultat);
    writeln(' deux fois PI = ',resultat)
end.

```

Ici, le programme envoie à fait\_tout les paramètres nécessaires au calcul, mais aussi la variable qui, au retour, contiendra la valeur de deux fois pi. Il y a donc échange dans les deux sens de données.

#### **2.7.2. Les fonctions**

Il s'agit de fonctions au sens courant du terme, c'est-à-dire que cette fonction prend une valeur d'un certain type, qui est le type de déclaration de la fonction.

En effet, une fonction est créée suivant le schéma suivant :

**function essai** (liste de paramètres) : type de la fonction ;

Par exemple, vous pourrez avoir :

```
function essai ( x,y:integer; a,b:real) : real;
```

Pour bien comprendre comment tout ceci fonctionne, reprenons l'exemple vu au début :

```
Program ex_const(output);
  const
    pi=3.1415926;
    deux=2;
  begin
    writeln(' deux fois PI =',deux*pi)
  end.
```

Nous allons le modifier artificiellement pour utiliser les connaissances dont nous disposons maintenant :

```
Program ex_const(output);

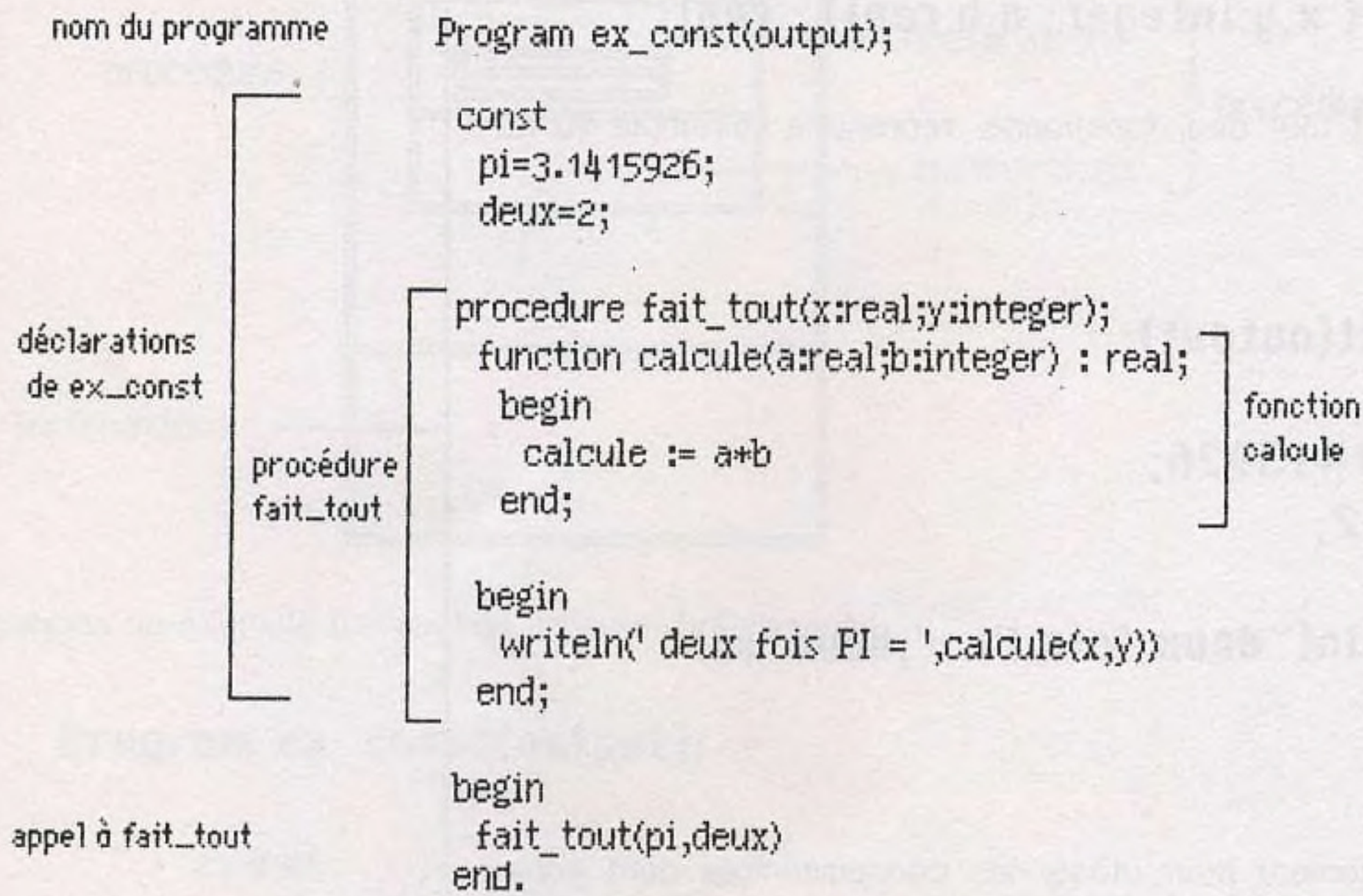
  const
    pi=3.1415926;
    deux=2;

  procedure fait_tout(x:real;y:integer);
    function calcule(a:real;b:integer) : real;
      begin
        calcule := a*b
      end;

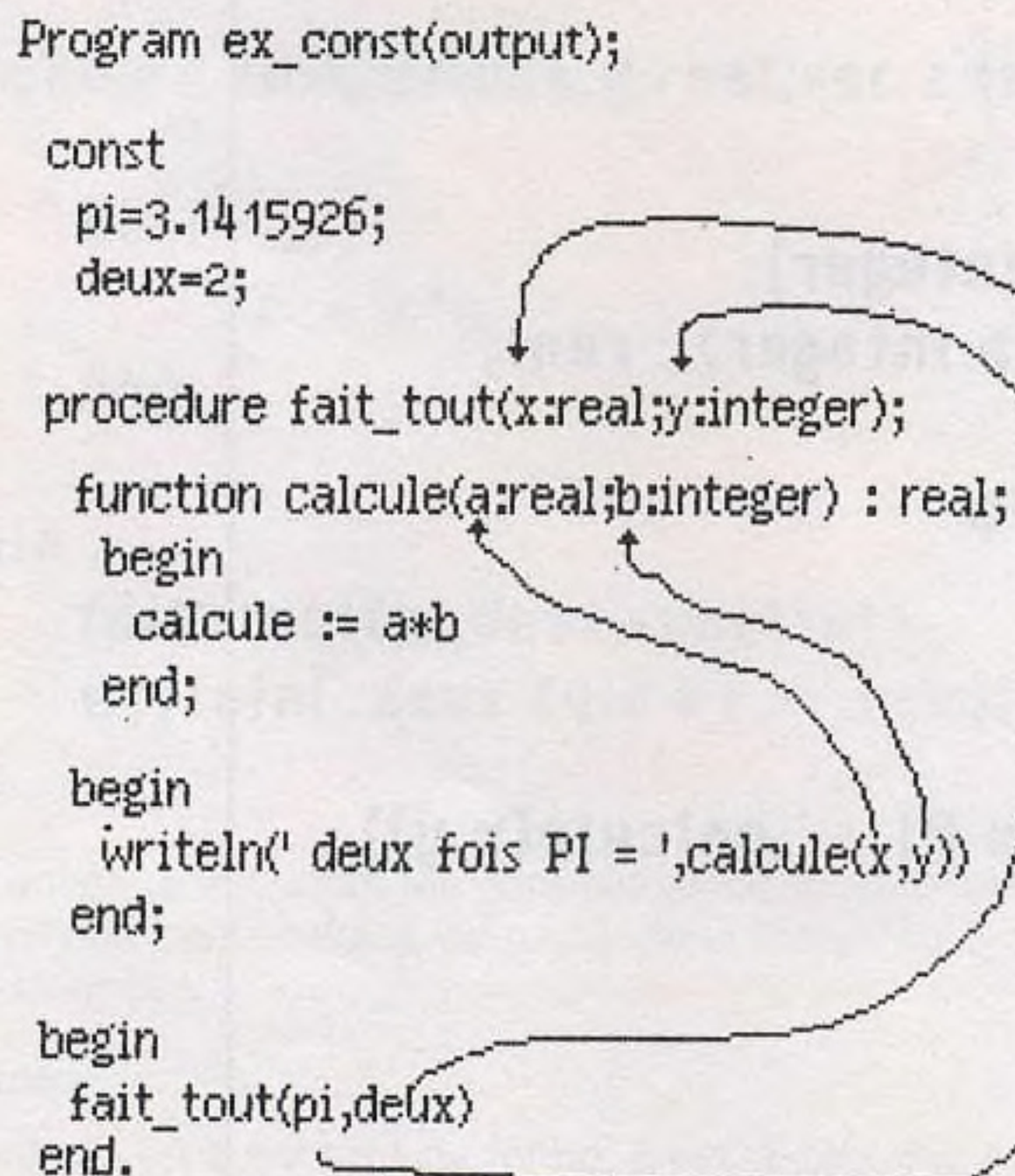
    begin
      writeln(' deux fois PI = ',calcule(x,y))
    end;

  begin
    fait_tout(pi,deux)
  end.
```

Analysons plus schématiquement ce qui se passe :



Maintenant, regardons plus en détail la circulation des paramètres :



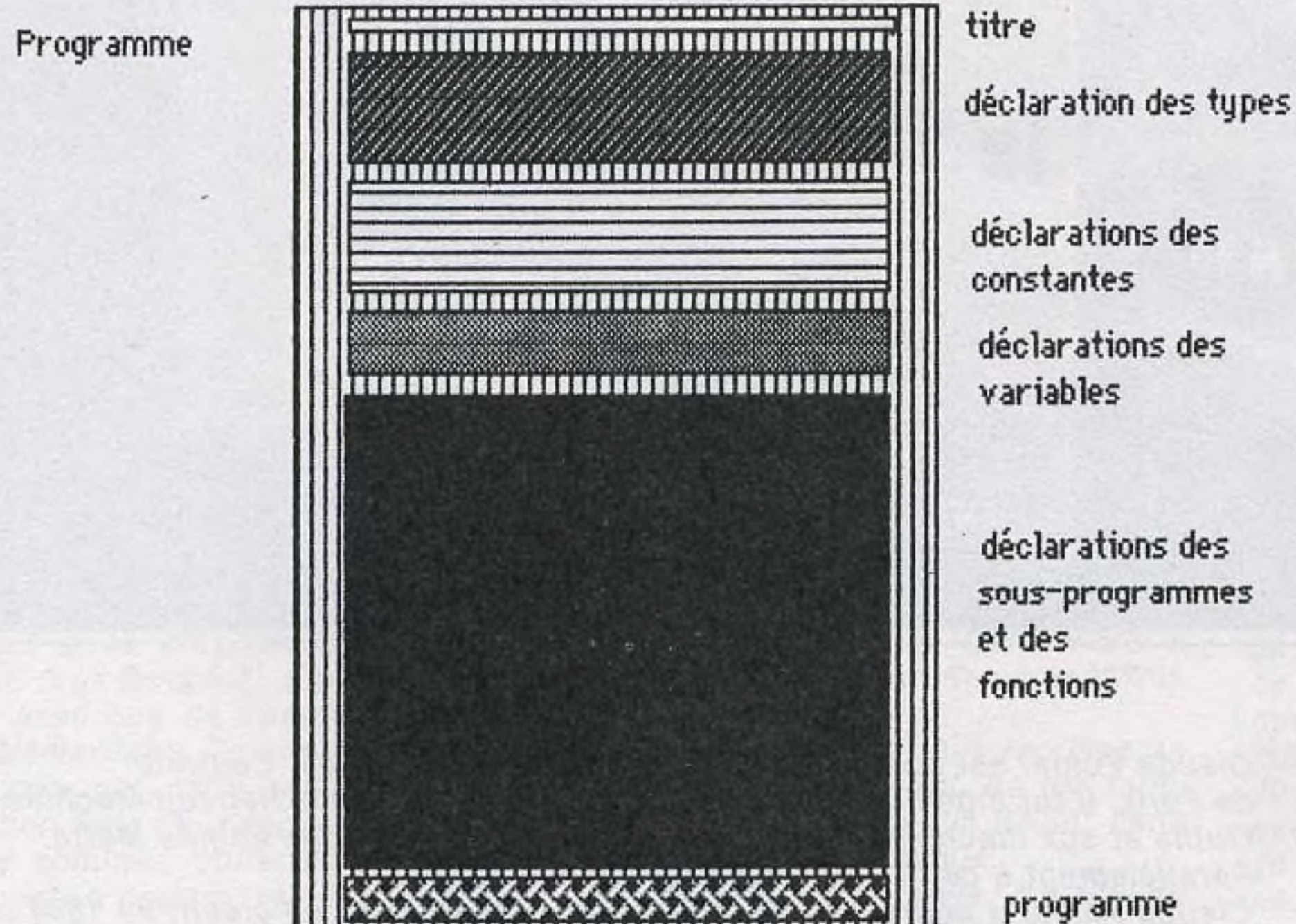


Le programme principal envoie à fait\_tout les paramètres pi et deux. Ce sous-programme les reçoit et les appelle localement, pour lui, x et y, puis les envoie à calcule qui, de la même façon les appelle, de manière purement interne, a et b. La fonction calcule se déroule et se termine par le renvoi de sa valeur, puisque c'est une fonction.

### III. CONCLUSION

Nous avons aujourd'hui fait le tour de ce qu'est un langage structuré. Nous avons décrit la rigueur de programmation qui va avec. Dès les prochains cours, nous pourrons donc étudier les commandes du langage Pascal, et vous vous rendrez vite compte que rien ne ressemble plus à un langage informatique qu'un autre langage informatique.

Pour nous résumer, nous avons pris connaissance de la forme générale d'un programme Pascal, qui est :



Et, croyez-nous, les proportions de ce dessin sont à peu près respectées, car le programme principal comporte en général fort peu d'instructions.

Nous avons vu de plus que cette structure est imbriquée, c'est-à-dire que chaque procédure a la même forme générale que le programme principal. Cela ressemble un peu à un problème de chimie moléculaire. Vous décrivez une molécule, qui est faite d'édifices atomiques, qui sont fait d'atomes, qui sont fait d'électrons et de protons, etc...

Le prochain cours sera consacré à la description de base d'un bloc d'instructions en Pascal, du BEGIN au END, soit en français bien de chez nous, du début à la fin. Nous parlerons également de l'affectation d'une valeur à une variable, des opérateurs arithmétiques et de l'utilisation du point-virgule (; pour ceux qui parlent le Pascal dans le texte).

Est-ce que nous vous communiquerons le virus du Pascal ?



*Claude Polgar est né en 1926 à Paris. Ingénieur de l'Ecole Centrale de Paris, il fut ingénieur d'études chez Kodak-Pathé, chez Renault-Machine-Outils et aux machines Bull puis chef de département aux engins Matra. Parallèlement à cette carrière classique d'ingénieur, Claude Polgar a poursuivi des recherches personnelles en créant en 1954 le matériel Prototypia (qui fut le premier «Meccano» de micro-robotique) et en 1982 le logiciel d'habillement Alamod (qui permet de réaliser des patrons personnalisés). Claude Polgar se consacre actuellement à l'enseignement des techniques modernes. Les Editions Fréquences ont publié son cours de programmation dans la revue Led-Micro.*

**2 volumes (près de 500 pages - format 21 x 27)  
représentant le récapitulatif de 2 ans des cours progressifs  
de Claude Polgar**

## DE NOMBREUX ADDITIFS

Que de changements depuis la sortie  
du numéro 1 de LED-MICRO !

Il n'est plus possible d'ignorer :

- le MS-DOS (le système d'exploitation de l'IBM PC)
- les Mémoires à Bulles
- le Compact-Disc
- le développement du Minitel et des réseaux de télématique amateur
- les notions de base de l'Intelligence artificielle (ce qu'est PROLOG etc...)
- l'emploi des calculettes aux examens.

J'ai profité de cette réédition pour ajouter des exercices, mieux présenter certains thèmes, donner aux professeurs le moyen de préparer des disquettes autochargeables.

Que voulez-vous ? C'est ma nature !

C. POLGAR

# le cours d'initiation à la micro-informatique le plus complet

## non, on ne s'initie pas à la micro-informatique et au basic en 5 leçons ou en 3 semaines !

Le mythe de l'informatique loisir facile s'est envolé, accéder à la programmation relève d'une pédagogie sérieuse et progressive, c'est le pari gagné que fit Led-Micro à une époque où fleurissait chaque jour un nouvel ouvrage-miracle.

Parmi les centaines de lettres reçues, nous nous permettons de citer 3 d'entre elles, elles permettent de situer comment, en général, a été perçu et apprécié ce cours.

*J'enseigne les mathématiques dans une Université de Sciences Humaines et j'ai été amenée, alors que je n'avais moi-même reçu aucune formation à la micro-informatique, à initier des étudiants de 1<sup>re</sup> année de Mathématiques et Sciences Sociales (MASS) à la programmation en S-BASIC (sur Goupil-3), dans le but de faire avec eux de l'analyse numérique élémentaire. Ce que j'ai fait, tant bien que mal, cette année, en collaboration avec deux autres collègues. Nous sommes conscientes d'avoir commis un certain nombre d'erreurs pédagogiques et nous souhaitons tenter d'y remédier l'an prochain. J'ai découvert votre revue tout récemment, alors que j'arrivais quasiment au bout de mon enseignement. J'ai été très sensible à votre démarche*

*pédagogique et je me sens personnellement tout à fait en accord avec votre manière de procéder. Je me suis procurée l'ensemble des n<sup>os</sup> de la revue et me permettrai de puiser dans votre cours certains exemples ou certaines façons de présenter les choses l'an prochain. Donc merci à vous...*  
C.L. St Cloud, le 22/5/85

*J'ai déjà essayé, à deux reprises au moins, antérieurement, de me familiariser vraiment avec le BASIC sans grand résultat, je l'avoue. La méthode que vous mettez en œuvre dans «Led-Micro» — me conduira-t-elle au but recherché, je n'en sais rien encore — a du moins le mérite d'être sympathique et agréable à suivre. Ma seule ambition étant d'utiliser les micros comme distrac-*

*tion intellectuelle (je suis retraité), j'espère ainsi y parvenir. Merci, donc, de votre aide et continuez à nous faire avancer progressivement et sûrement.*

Docteur Y.C. Sees, le 19/2/84

*Je viens de découvrir votre magazine ce matin dans un kiosque, cet après-midi je vous commande les 18 premiers numéros.*

*Je suis très emballé par vos cours, que je trouve très bien faits.*

*Je suis un «vrai» débutant, je possède un ZX81 que j'ai du mal à faire tourner, par manque d'information, grâce à vos cours je pense que j'y arriverais. Je possède pas mal de bouquins sur la question mais aucun n'explique aussi clairement que vous.*  
A.A. Marseille, le 17/4/85

Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

### Initiation à la micro-informatique C. Polgar

Bon de commande à retourner aux Editions Fréquences 1, boulevard Ney 75018 Paris.

Je désire recevoir le tome 1  140 F (130 F + 10 F de frais de port)  
le tome 2  140 F (130 F + 10 F de frais de port)  
les deux tomes  280 F (260 F + 20 F de frais de port)

Je joins mon règlement à la commande :

chèque bancaire

mandat

C.C.P.

Nom .....

Prénom .....

Adresse .....

Code postal .....

Localité .....

# COURS DE PROGRAMMATION APPROFONDIE

Dominique Chastagnier  
Jean-François Coblenz  
Patrick Gueneau

Le sujet de ce mois-ci, ce sont les programmes qui ne tournent pas. Vous conviendrez de l'utilité d'un tel cours ! Nous partagerons le problème en trois parties. D'abord la classification des types d'erreurs que l'on peut rencontrer. Ensuite, les méthodes usitées pour diagnostiquer et rectifier les erreurs qui ne manquent pas d'arriver. Enfin, un recensement des principaux messages d'erreurs envoyés par la plupart des ordinateurs vendus en France, avec leur traduction et leur explication.

Il n'y aura pas ce mois-ci de Dialogue avec les lecteurs. Patientez. La place prise par le début du Pascal ne nous a pas permis d'insérer les tours de Hanoi. Ce n'est que partie remise. Rendez-vous le mois prochain.

## COURS N° 10

### Les erreurs

#### PLAN DU COURS

1. Votre programme ne tourne pas
  - 1.1. Interprétation et compilation
  - 1.2. Les erreurs immédiates simples
  - 1.3. Les erreurs en cascade
  - 1.4. Les erreurs différées
2. Préparation aux erreurs
  - 2.1. Le routage
  - 2.2. L'affichage
  - 2.3. Précautions d'emploi
  - 2.4. Les tests
3. Les messages d'erreur
4. Exercices d'applications
  - 4.1. Les carrés magiques
  - 4.2. La communauté économique européenne

Il a plusieurs moyens de perdre une fortune : la manière la plus agréable, ce sont les femmes ; la plus rapide c'est le jeu ; mais la plus sûre demeure l'ingénieur. A cet adage, on pourrait substituer avantageusement l'informaticien à l'ingénieur. En voici quelques autres aphorismes dont votre propre expérience vous suggèrera de nombreux exemples :

- Aucun grand projet informatique n'a tourné correctement dans les délais et avec le budget prévu, pourquoi le votre serait-il le premier ?
- La mise au point de 90 % du programme prend 90 % du temps, la mise au point des 10 % restants prend aussi 90 % du temps.
- Lorsque un programme tourne complètement, cela signifie qu'il est périmé.
- Si un concepteur vous dit que son programme a quelques restrictions d'emploi, celui-ci ne marche que dans un seul cas.
- La même personne certifie avoir privilégié l'encombrement mémoire, attendez-vous à des performances temporelles désastreuses.

Ce ne sont que quelques remarques tendancieuses dispensées par les usagers de l'informatique mais elles soulignent le sujet toujours délicat de la mise au point des programmes. Combien d'entre vous se sont couchés beaucoup plus tard que prévu car leur programme avait encore un «bogue» (bug pour les anglicistes) aussi irréductible que les gaulois d'Astérix.

## 1. VOTRE PROGRAMME NE TOURNE PAS

### 1.1 Interprétation et Compilation

Nous n'allons nous préoccuper aujourd'hui que des programmes affichant des messages d'erreur et interrompant ainsi leur exécution. Nous n'aborderons que de manière anecdotique les programmes fonctionnant correctement mais ne faisant pas les opérations souhaitées. Vous soupçonnez bien que dans ce cas l'analyse du problème est à revoir ; cependant, il est possible de résoudre immédiatement certaines erreurs d'analyses bénignes, lors de l'exécution.

Si votre programme est simplement interprété, vous aurez le plaisir de découvrir conjointement à l'exécution de vos erreurs de syntaxes les plus grossières comme celles résultant plus pernicieusement mais de résultat tout aussi efficace, des valeurs de vos variables. A l'opposé, un programme compilé vous signale avant le démarrage de l'exécution, un certain nombre de vices de forme lié à la syntaxe du BASIC.

```
15 IF I = 1 GOTO 2000
```

```
10 A = 0
20 B = 50 / A
```

Si la première est détectée par tous les compilateurs (il manque THEN), la seconde ne l'est certainement pas (l'instruction 20 provoque une division par 0 : opération toujours périlleuse !). Toutefois, les compilateurs n'ont pas tous le même sens critique vis-à-vis de ce que vous écrivez : ainsi l'erreur suivante peut passer inaperçue.

```
30 A% = B / 100
```

Il y a incompatibilité de genre entre les deux membres de l'égalité. Il n'existe pas dans la plupart des BASIC compilés le message de WARNING qui vous signale un danger, que vous encourez à vos risques et périls en prenant certaines

libertés tolérées par le BASIC. Ces facilités, si elles sont une preuve de souplesse, sont néanmoins source de bien des problèmes. Les transferts entiers réels sont les plus fréquents.

### 1.2 Les erreurs immédiates simples

La plupart d'entre elles seront développées plus spécifiquement dans le paragraphe sur les messages fournis. Vous vous devez d'avoir à votre disposition une traduction française de tous les messages d'erreur de votre ordinateur, bien des programmes sont restés en plan, faute d'avoir pu comprendre ce que l'ordinateur demandait. C'est une erreur aussi exaspérante que commune ! Ne perdez pas de vue que le nombre de messages est plus limité que le nombre de types d'erreurs et que certains messages englobent plusieurs problèmes possibles. «Division par zéro» ne signifie pas nécessairement que la dernière instruction est une division mais plus généralement que l'exécution de la dernière instruction a engendré une division où le dénominateur était nul.

Par exemple, sur certains ordinateurs, demander le calcul de LOG (0) renvoie le message «Division par zéro».

### 1.3. Les erreurs en cascade

Les erreurs dites en cascade ne peuvent exister que dans le cadre des programmes compilés ; elles proviennent de la règle communément adoptée par les compilateurs ; le compilateur ignore la totalité de chaque ligne d'instruction où se trouve une erreur. Lorsque celle-ci est une définition de variable dimensionnée, cela peut engendrer un message pour chacune des lignes suivantes où se trouve référencé le tableau.

```

10 DIM A(10,20)      *** SYNTAX ERROR ***
20 A(5,11) = 1       *** SUBSCRIPT OUT OF RANGE ***
30 A(5,12) = 2       *** SUBSCRIPT OUT OF RANGE ***
20 A(5,13) = 3       *** SUBSCRIPT OUT OF RANGE ***
20 A(5,14) = 2       *** SUBSCRIPT OUT OF RANGE ***
20 A(6,11) = 1       *** SUBSCRIPT OUT OF RANGE ***
20 A(6,12) = 3       *** SUBSCRIPT OUT OF RANGE ***
20 A(6,13) = 2       *** SUBSCRIPT OUT OF RANGE ***

```

La simple correction de la première erreur verra la disparition de toutes les autres. De même, la disparition d'une ligne entraîne la disparition d'une autre ligne qui elle-même provoque de nouveaux dégâts. Certaines premières compilations sont décourageantes en faisant apparaître une centaine d'erreurs très rapidement ramenées à une dizaine, les autres n'étant que des cascades.

```

10 DEF FN CUBE (X) = ( X * X * X ) / A$      *** TYPE MISMATCH ***
20 A(1) = FN CUBE (5)      *** UNDEFINED USER FUNCTION ***
20 A(2) = FN CUBE (10)     *** UNDEFINED USER FUNCTION ***
20 A(3) = FN CUBE (13)     *** UNDEFINED USER FUNCTION ***
20 A(4) = FN CUBE (15)     *** UNDEFINED USER FUNCTION ***
20 A(5) = FN CUBE (17)     *** UNDEFINED USER FUNCTION ***
20 A(6) = FN CUBE (19)     *** UNDEFINED USER FUNCTION ***

```

### 1.4 Les erreurs différées

Si elles sont moins nombreuses que les erreurs en cascade, elles sont paradoxalement plus fréquentes. Elles se rencontrent essentiellement à l'exécution ; ce sont des instructions correctes syntaxiquement mais incorrectes dans la logique du programme. Elles ont donc pour conséquence d'entraîner plus loin dans le programme,

une erreur qui ne correspond pas à une remise en cause de l'instruction défailante. Si vous ne dimensionnez pas suffisamment votre tableau, l'exécution ne le diagnostiquera pas lors du passage sur l'instruction DIM mais seulement à la première référence dépassant les bornes permises puisque en 730  $I = 21$ .

```

10 DIM A(20)
.....
135 I = 7
.....
290 I = I * 2
.....
465 I = I + 5
.....
640 I = I + 2
.....
730 A(I) = 0      *** SUBSCRIPT OUT OF RANGE ***

```

Il est évident de reconnaître une erreur différée, la corriger est plus délicat.

## 2. PREPARATION AUX ERREURS

Comme aux échecs, toutes les fautes sont devant vous, attendant d'être commises. Aussi, apprend on à vivre avec elles et la meilleure façon est de se prévenir contre elles en les considérant comme inévitables et en prenant des mesures préalables.

### 2.1 Le routage

La première règle à respecter est la suivante :

A chaque portion de programme ayant une certaine cohésion, on insère un message d'entrée dans la section sous forme de PRINT «Entrée calcul de x (i)» par exemple et à la fin, on remet «sortie calcul de x (i)».

```

.....
100 PRINT " ENTREE CALCUL X(I) "
.....
200 PRINT " SORTIE CALCUL X(I) "
.....

```

Ce supplément d'information est certes fastidieux tant à saisir au clavier que du point de vue place mémoire, mais il aide sérieusement à la mise au point. Ses avantages sont les suivants :

- Il indique dans quel ordre sont exécutées les différentes portions du programme.
- Il met en évidence les portions où le programme ne va jamais, portions qui sont autant de fautes d'analyse du programme.
- Il signale enfin les sections les plus fréquentées, information importante pour une amélioration ultérieure du produit.
- Enfin, si votre programme boucle sur lui-même, la constatation est plus sûre qu'un programme tardant à retourner la main ; cette situation provoquant toujours un fond d'angoisse chez le programmeur : Interrompra ou n'interrompra pas une exécution qui ne boucle peut-être pas ?

Il n'existe malheureusement pas en BASIC de système donnant d'une part le nombre de passages dans chaque instruction, d'autre part le temps passé dans chaque portion, ne rêvons pas, mais c'est dommage.

## 2.2 L'affichage

Naturellement, certains programmes vous affichent lors d'erreurs la valeur de quelques variables caractéristiques ; la plupart des assembleurs vous présentent ainsi le contenu de tous leurs registres. Si cela n'existe pas sur votre BASIC, c'est à vous de le simuler. En effet, si vous ne le faites pas immédiatement, il y a fort à parier que vous y serez contraints à l'exécution pour déterminer les composantes de tel ou tel problème, alors ? Pourquoi attendre ?

De plus, cela vous permet, au déroulement du programme de pouvoir mettre le holà à votre exécution lorsque vous voyez vos variables prendre une direction que vous aviez été loin de prévoir originellement. La différence entre une installation préventive et une autre effectuée à chaud réside dans le fait que vous choisissiez tranquillement les endroits où disposer vos points d'arrêts, car il va sans dire que pour pouvoir aisément méditer sur la valeur de vos variables, il n'est pas toujours inutile de suspendre le déroulement de votre programme.

## 2.3 Précautions d'emploi

Ces outils ne doivent pas être plus présents que les gendarmes sur les routes de France. En effet, trop de contrôles d'identité nuirait fortement au temps nécessaire à votre voyage, de même, un programme ne doit pas envoyer exagérément de messages.

## 2.4 Les tests

Mais plus dangereux encore, se trouve un écueil auquel peu résiste : lorsque le programme semble tourner, il est difficile de ne pas éliminer toutes les informations de routage et autres affichages parfois disgracieux pour le programmeur. On se remet alors à travailler sans filet et un programme avec des variables sensiblement analogues ne va pas nécessairement au même endroit ni dans le même ordre. C'est alors la loi de la « tartine » (attribuée à Murphy), qui reprend le dessus : cela tombe toujours du mauvais côté, au mauvais moment et alors vous regrettez amèrement d'avoir tout enlevé car cela arrive toujours quand vous présentez le produit à votre fiancée ou pire encore belle-maman ! En effet, si le spectateur moyen est toujours fasciné par l'exécution inexorable d'un programme parfait, il n'est que peu enthousiaste par le spectacle de la recherche d'un bug, qui vous fait exprimer dans un langage fort sybillin à son goût. Aussi, pour conserver intacte votre image de marque auprès de votre entourage, gardez de quoi comprendre rapidement une nouvelle erreur.

Pour éviter cela, vous avez eu préalablement recours aux tests. Cette partie de la programmation est aussi utile que les autres. S'il est difficile d'envisager tous les cas possibles, il est impératif de penser à tous les cas probables.

## 3. LES MESSAGES D'ERREURS

### NEXT WITHOUT FOR (Next sans for)

Vous avez inséré un «Next» sans faire précéder cette instruction par sa conjointe «For». Ou bien votre «Next» n'est pas suivi de la même variable que le «For».

### SYNTAX ERROR (Erreur de syntaxe)

Votre instruction comporte une «faute de frappe» qui la rend incorrecte du point de vue de la syntaxe BASIC. Cette faute est souvent très simple : absence d'un point virgule, de deux points, mot réservé mal orthographié.

### RETURN WITHOUT GOSUB (Retour sans appel de sous programme)

Vous arrivez sur un ordre de retour de sous-programme sans être venu par le biais d'un sous-programme. Il ne sait donc où aller.

### OUT OF DATA (Plus de données)

Vos instructions «Read» de lecture des «Data» ont intégralement lu ceux-ci alors que vous demandez de nouvelles lectures. Soit il manque des données, soit il faut relire les mêmes («Restore»).



**ILLEGAL FUNCTION CALL (Appel illegal de fonction)**

Ce diagnostic englobe beaucoup d'erreurs ; toutes les erreurs consécutives à la présence d'une valeur erronée parmi les paramètres d'utilisation de la fonction ; racine de négatif, logarithme de 0 ou certaines fautes de frappe comme deux valeurs, fournies là où une seule est nécessaire. Mais aussi toutes les erreurs de même type survenues sur les fonctions manipulant les chaînes de caractère (Mid\$, Left\$, etc...).

**OVERFLOW (Dépassement)**

Vous avez obtenu un résultat dont la valeur est supérieure à celle calculable par la machine. Exemple : un entier supérieur à 32767 ou inférieur à - 32768.

**ILLEGAL QUANTITY (Quantité illégale)**

Toutes les erreurs numériques d'«illegal function call».

**OUT OF MEMORY (En dehors de la mémoire)**

Vous venez de saturer la place mémoire qui vous était allouée soit en insérant une nouvelle instruction, soit en créant de nouvelles variables.

**UNDEFINED LINE NUMBER - UNDEF'D STATEMENT - UNDEFINED LABEL (Numéro de ligne non défini)**

Votre instruction comporte la référence (goto, then, etc). 0 a une ligne inexistante ou détruite.

**SUBSCRIPT OUT OF RANGE - BAS SUBSCRIPT (Indice de limite)**

Vous avez fait référence à un élément de tableau d'indice erroné : non compris, dans les valeurs de votre instruction «Dim» ou supérieur à 10 si vous n'avez pas déclaré votre tableau.

**REDIMENSIONNED ARRAY - REDIM'D ARRAY - DUPLICATE DEFINITION (Tableau redimensionné - définition dupliquée)**

Vous avez défini deux fois la même chose : deux «dim» pour un même tableau.

**DIVISION BY ZERO (Division par zéro)**

Cette instruction amène l'ordinateur à une division dont le dénominateur est nul.

**ILLEGAL DIRECT (Direct illégal : la traduction fut pénible !)**

L'introduction est interdite à l'exécution en mode direct : elle doit être impérativement incluse dans un programme.

**TYPE MISMATCH (Type non correspondant)**

Vous avez mélangé des variables n'étant pas de même type : entières avec réels, caractères avec numériques.

**DISP TYPE MISMATCH (Type d'environnement non correspondant)**

Vous avez employé une introduction dans un mauvais contexte. Des ordres d'impression alors que vous êtes en graphique. Tous les ordinateurs ne le décèlent pas.

**OUT OF STRING SPACE (Dépassement de l'espace chaîne)**

Votre nouvelle chaîne de caractères provoque un dépassement de capacité de l'espace réservé aux chaînes.

**STRING TOO LONG (Chaîne trop longue)**

Votre variable chaîne de caractères dépasse les limites prévues par le constructeur. Il y a plus de deux IF-THEN compris dans cette instruction : c'est trop.

**STRING FORMULA TOO COMPLEX (Expression trop complexe)**

Instruction trop compliquée ou trop longue, il faut la diviser en deux.

**CAN'T CONTINUE (On ne peut continuer)**

Vous demandez de continuer alors qu'il ne peut pas, suite à un arrêt sur erreur, une modification du programme ou si le programme lui-même n'existe pas.

**UNDEFINED USER FUNCTION - UNDEF'D FUNCTION (Fonction non définie)**

Vous avez oublié de définir une fonction à laquelle vous faites référence ou bien vous l'avez mal orthographiée.

**REDO FROM STARD (???)**

Ce message de l'ORIC signifierait qu'un caractère non numérique aurait été saisi en lieu et place d'un caractère numérique.

**BAD UNTIL (Mauvais until)**

Le programme rencontre un «Until» sans avoir préalablement exécuté le «Repeat» nécessaire.

**DEVICE I/O ERROR (Erreur d'entrée/sortie)**

Une erreur est apparue lors d'un échange avec un périphérique. L'état, dans lequel se trouvent les fichiers éventuellement concernés, est souvent critique !

**DEVICE TIME OUT (Appareil hors delais)**

Un périphérique n'a pas conversé avec le programme dans le temps imparti.

**DEVICE FAULT (Erreur périphérique)**

Erreur provoquée par le périphérique.

**OUT OF PAPER (Plus de papier)**

Message concernant l'imprimante. Une absence de connexion peut produire un message analogue.

**INTERNAL ERROR (Erreur interne)**

Panne de l'unité centrale. Voyez votre revendeur.

**DISK FULL (Disque plein)**

Soit il faut détruire des fichiers pour faire de la place, soit en prendre un autre.

**DISK WRITE PROTECTED (Disk protégé à l'écriture)**

Tentative avortée d'écrire sur un disque protégé.

**DISK NOT READY (Disque non prêt)**

Une des conditions de lecture (présence d'un disque, non fermeture du volet) n'est pas remplie.

**DISK MEDIA ERROR (Erreur sur l'unité de disque)**

Panne matériel, détectée par l'ordinateur. Appelez votre revendeur.

**ADVANCED FEATURE (Amélioration)**

Votre programme requiert un BASIC plus sophistiqué que celui employé.

**VERIFY ERROR (Erreur de vérification)**

Vous avez un programme chargé en mémoire centrale différent de celui conservé sur bande ou disque.

**NO RESUME (Pas de reprise)**

Un branchement a été provoqué par une erreur, mais celui-ci n'a rencontré aucun ordre de reprise.

**RESUME WITHOUT ERROR (Reprise sans erreur)**

Il rencontre un ordre de reprise alors qu'il n'est pas arrivé ici, consécutivement à une erreur.

**MISSING OPERAND (Opérande manquant)**

Une expression contient un opérateur sans opérande.

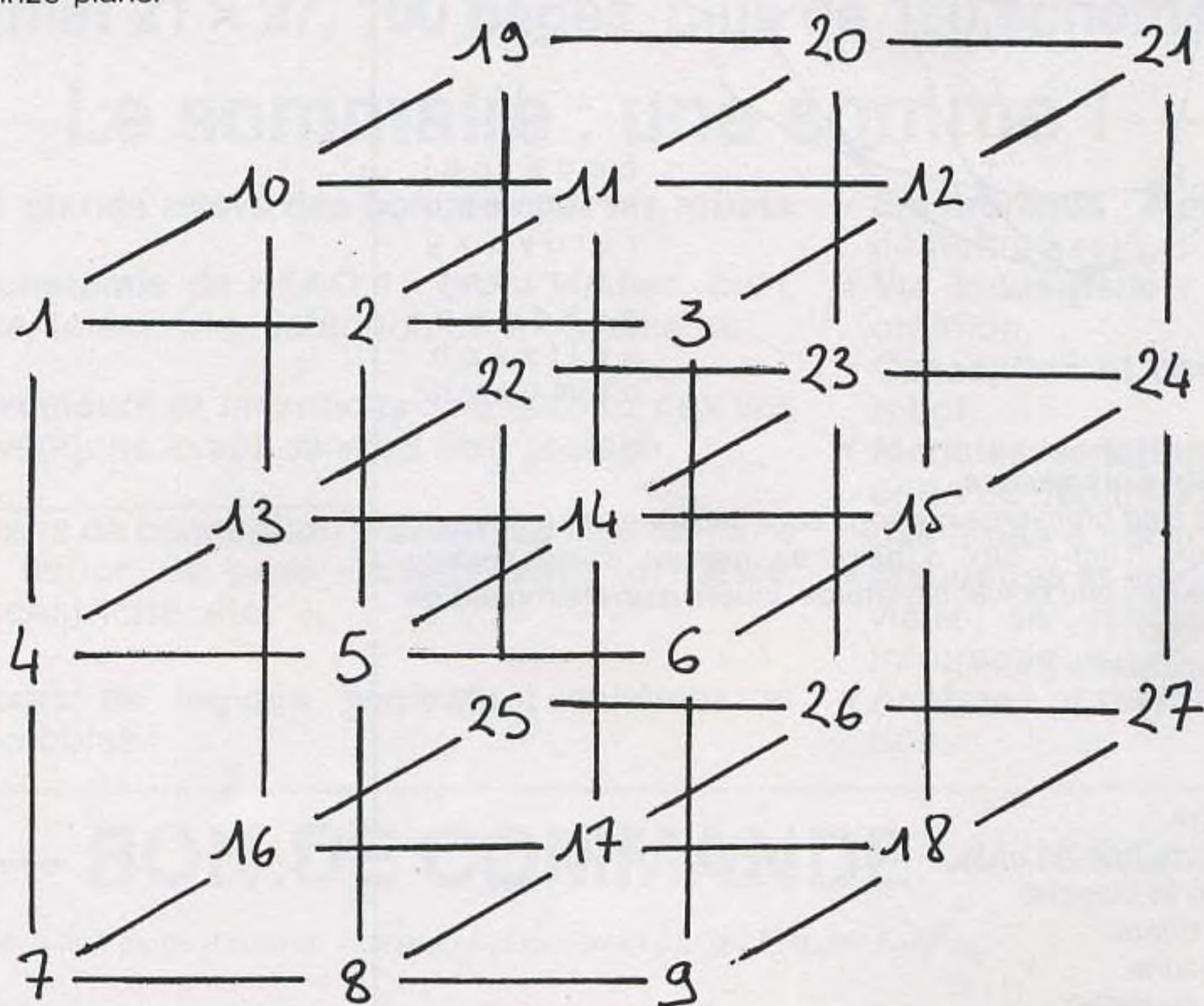
**LINE BUFFER OVERFLOW (Débordement du tampon de ligne)**

Votre ligne d'instruction contient trop de caractères. Divisez-la en deux.

**4. EXERCICES D'APPLICATIONS****1. Les carrés magiques (suite)**

Les carrés magiques semblent ouvrir bien des horizons et exciter l'imagination. Voici maintenant les carrés magiques hétérogènes : ce sont des carrés toujours formés des mêmes nombres mais là, toutes les colonnes, lignes et autres diagonales ont des sommes différentes !

Les cubes magiques sont des cubes  $3 \times 3 \times 3$  où tous les sous-ensembles de 9 nombres dans un même plan ont la même somme. Nous vous avons dénombré les quinze plans.



1.2.3.4.5.6.7.8.9

10.11.12.13.14.15.16.17.18

19.20.21.22.23.24.25.26.27

1.2.3.10.11.12.19.20.21

4.5.6.13.14.15.22.23.24

7.8.9.16.17.18.25.26.27

1.4.7.10.13.16.19.22.25

2.5.8.11.14.17.20.23.26

3.6.9.12.15.18.21.24.27

1.2.3.13.14.15.25.26.27

7.8.9.13.14.15.19.20.21

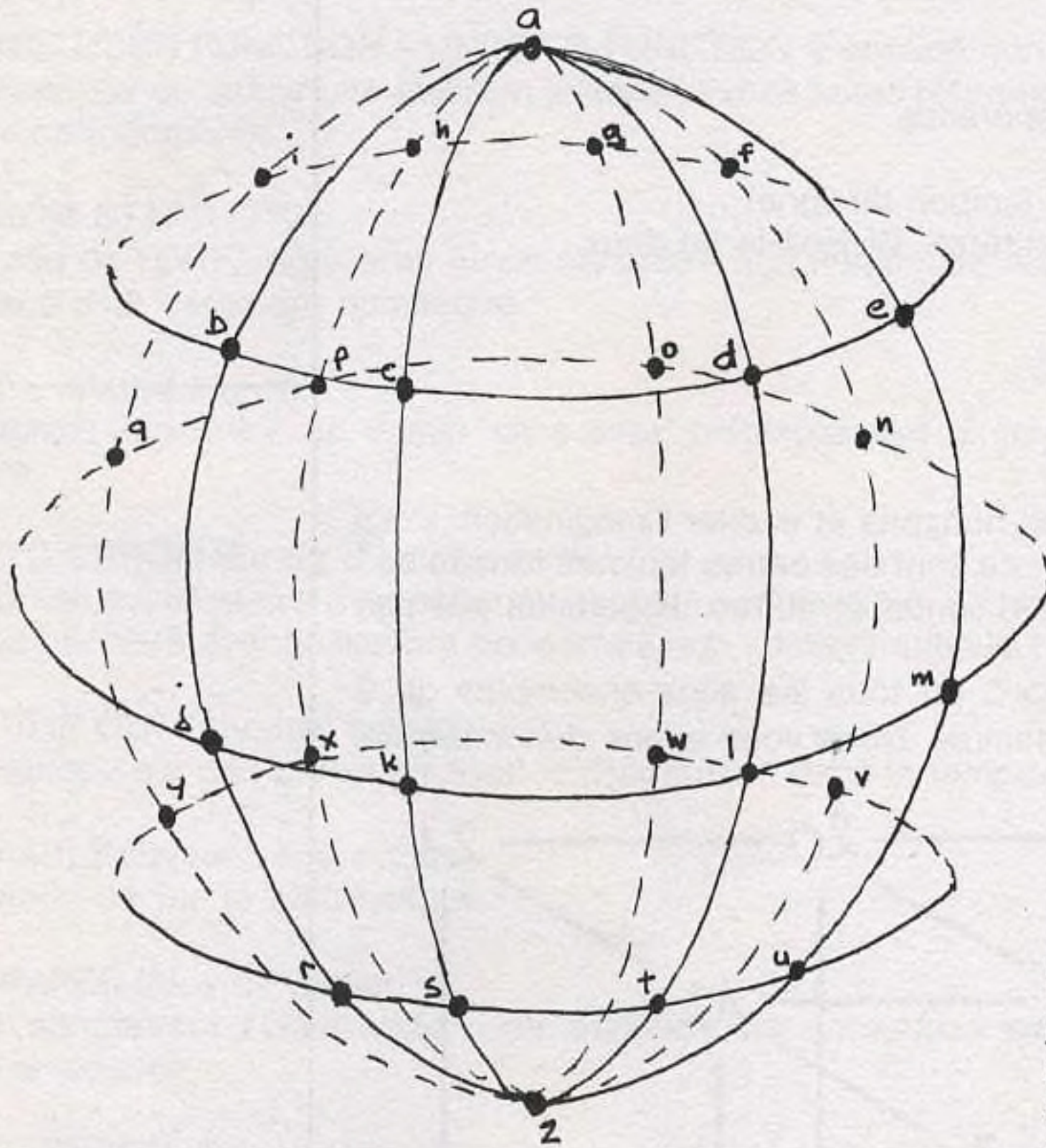
1.4.7.11.14.17.21.24.27

3.6.9.11.14.17.19.22.25

1.5.9.10.14.18.19.23.27

3.5.7.12.14.16.21.23.25

Enfin la sphère magique, où les 26 premiers nombres entiers sont rangés aux intersections de cercles comprenant chacun 8 nombres et dont les sommes sont une nouvelle fois égales.



bcdefghi  
 jklmnopq  
 rstuvwxy  
 abjrzvnf  
 ackszwog  
 adltzxp  
 aemuzyqi

## 2. La Communauté Economique Européenne

A Strasbourg, il y a une rue où les cinq premières maisons sont habitées par un député du Parlement Européen. Chacun d'entre eux a peint sa maison d'une couleur différente. Ils ont tous un animal favori, une boisson préférée, jouent d'un instrument de musique et aucun goût en commun.

- L'Allemand habite la première maison à gauche.
- L'Espagnol possède un chien.
- Le Grec est pianiste.
- L'Italien boit du thé.
- L'Anglais habite la maison rouge.
- Le propriétaire de la maison verte boit du café.
- La maison verte est à droite de la blanche.
- Le trompettiste joue avec son canari.
- Le guitariste habite la maison jaune.
- On boit du lait dans la maison du milieu.
- L'Allemand habite à côté de la maison bleue.
- Le violoniste boit du vin.
- Le poisson rouge est dans la maison voisine du batteur.
- Le cheval habite chez le voisin du guitariste.

Alors qui élève un lapin et boit de l'eau ?

Le problème peut se scinder en deux. Vous pouvez résoudre vous-même le problème à la main, mais aussi et surtout le faire résoudre par l'ordinateur.

**VOICI ENFIN LA PREMIÈRE PIERRE  
D'UN DOMAINE ENCORE INEXPLORÉ...**

L'ouverture au monde passionnant de la robotique, dans un style simple et direct, travail d'un collectif de spécialistes animé par Claude Polgar.



**PRIX TTC 115 F**

**hors série**  
LEDES D'AUJOURD'HUI  
**Led**  
**ROBOT**

# INITIATION A LA ROBOTIQUE

**Format 21 x 27, 100 pages, plus de 130 schémas et illustrations.**

## Le sommaire : une somme !

- **La grande relève des hommes par les robots**
- **L'anatomie de HERO 1** : bras, jambes, ouïe, vue, télémétrie, détection de mouvements.
- **Inventeurs et inventions** : ne confiez pas vos inventions avant de vous être protégé.
- **Cours de conception mécanique** : vocabulaire et notion de base - Ajustement, tolérance, excentricité, etc.
- **Cours de logique générale** : schémas et symboles.
- **Electronique industrielle** : du circuit au démultiplexeur.
- **Vie industrielle** : la CAO, assistante de la création.
- **Conception et construction** : de la tortue au robot.
- **Modules fonctionnels** : construction de la carte de départ pour commander les moteurs pas à pas à partir de votre micro.
- **Maquettes et modélisme** : le modélisme ferroviaire se renouvelle grâce à la micro-informatique.
- **Analyses et méthodes** : les rosaces d'évaluation.

## BON DE COMMANDE

Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

Je désire recevoir Led-Robot «INITIATION A LA ROBOTIQUE» (attention, cet ouvrage n'est pas vendu en kiosque) au prix de **125 F** (port compris).

Nom : ..... Prénom : .....

Adresse : .....

**ATTENTION** : Si je suis abonné soit à LED, soit à LED-MICRO, je bénéficierai d'une réduction de 20 % sur le prix de l'ouvrage et je ne paierai que **100 F** (port compris).

Je vous note, dans le cadre, mon numéro d'abonné :

Ci-joint un chèque bancaire  chèque postal  mandat .

Adressez votre commande et votre règlement aux **EDITIONS FRÉQUENCES 1**, boulevard Ney, 75018 Paris.



# BIBLIOTHEQUE TECHNIQUE

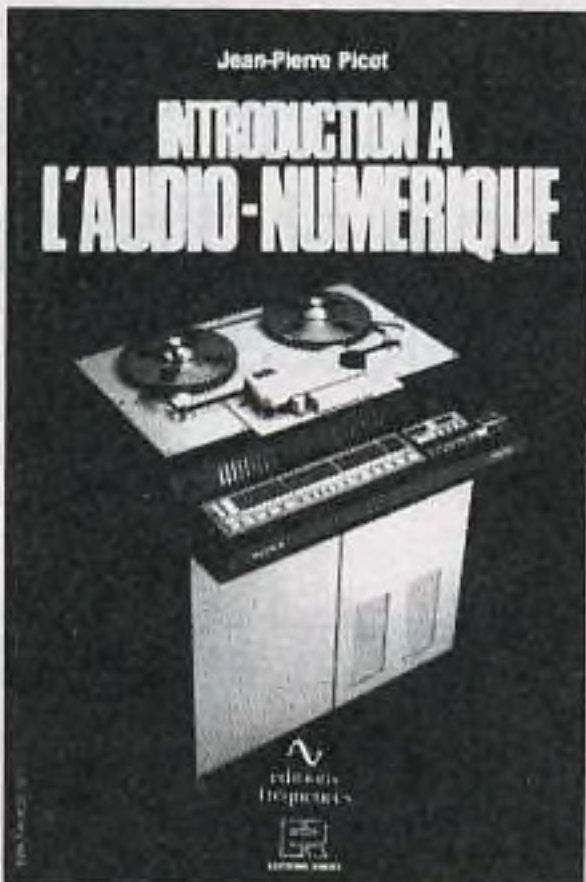
## Collection études (format 165 x 240)



**E 15.** 184 p. Prix : 140 F TTC  
Face au développement spectaculaire des synthétiseurs, grâce à l'électronique numérique, le besoin d'un ouvrage complet, accessible, et surtout bien informé des dernières ou futures techniques, se faisait ressentir. Le vœu est comblé, en 180 pages... à dévorer.



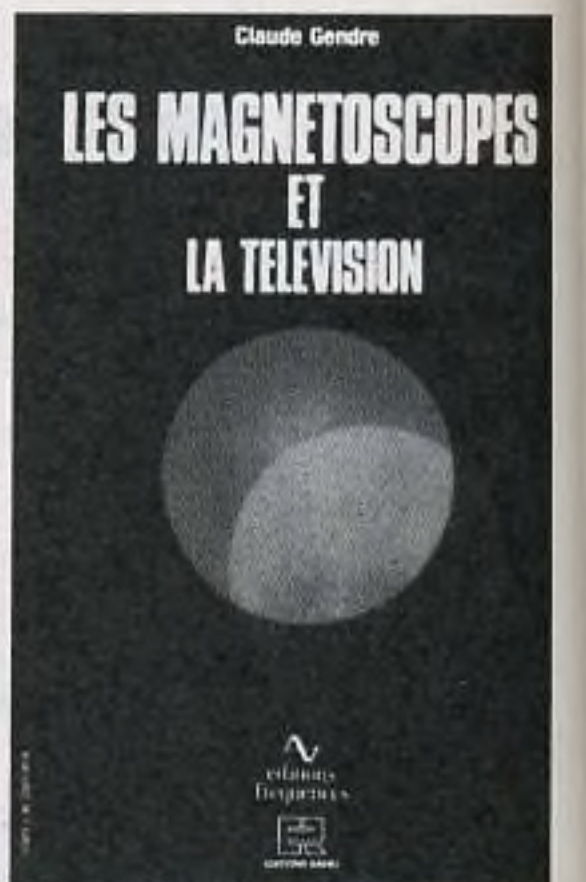
**E 01.** 320 p. Prix : 165 F TTC  
Un gros volume qui connaît un succès constant : bien plus qu'un traité, il s'agit d'une véritable encyclopédie, alliant théorie et pratique, histoire, en une mine inépuisable d'informations, reconnue dans le monde entier !



**E 05.** 160 p. Prix : 155 F TTC  
C'est le premier ouvrage paru en langue française traitant de l'audio-numérique : écrit par un professionnel, avec rigueur, simplicité, il explique brillamment les bases de cette technique : quantification, conversion, formats, codes d'erreurs.



**E 04.** 240 p. Prix : 154 F TTC  
Seconde édition améliorée d'un ouvrage fort attendu des passionnés d'électroacoustique. Ce livre permet aux amateurs et aux professionnels de se familiariser avec les rigoureuses techniques de modélisation des haut-parleurs et enceintes acoustiques et d'en mener à bien la réalisation.



**E 03.** 256 p. Prix : 155 F TTC  
Complément direct des «Magnétophones», les «Magnétoscopes et la Télévision» débute par un bel historique de la télévision et la description des premiers magnétoscopes. La théorie et la pratique de la capture et de l'enregistrement moderne des images vidéo en sont la teneur essentielle.



**E 22.** 136 p. Prix : 150 F TTC  
Faisant suite à la parution de «L'électronique des micro-ordinateurs», cet ouvrage s'adresse aux électroniciens qui désirent s'initier aux montages périphériques des micro-ordinateurs, interfaces en particulier, qui permettent la communication avec le monde extérieur.



**E 06.** 128 p. Prix : 150 F TTC  
Cet ouvrage est destiné aux électroniciens désireux d'aborder l'étude du «hard» des micro-ordinateurs. Cette étude s'articule autour du microprocesseur Z-80, très répandu, et en décrit les éléments périphériques : mémoires, clavier, écran, interfaces de toutes sortes.



**E 02.** 160 p. Prix : 92 F TTC  
Pour tout savoir sur le magnétophone, depuis l'avènement de cette mémoire des temps modernes, jusqu'aux enregistreurs numériques en passant par la cassette. «Les magnétophones» est un ouvrage pratique, complet, indispensable à l'amateur d'enregistrement magnétique.



**E 13.** 256 p. Prix : 165 F TTC  
Une sélection des meilleurs articles de la célèbre revue «l'Audiophile» choisis parmi les plus significatifs des quinze premiers numéros, introuvables aujourd'hui. Le tome 1 traite de l'électronique audio, à tubes et à transistors.



**E 12.** 256 p. Prix : 155 F TTC  
Dans un esprit identique, le tome 2 traite du domaine passionnant que constituent les transducteurs en audio : on y aborde la modélisation théorique des enceintes, la conception géométrique des tables de lecture, le réglage des cellules et des bras.

## Collection loisirs (format 135 x 210)



**L 07.** 160 p. Prix : 68 F TTC  
Le «dernier-coup de patte» apporté à un montage, celui qui fait la différence entre la réalisation approximative et le kit bien fini, ce savoir-faire s'acquiert au fil des ans... ou en parcourant «Conseils et tours de main en électronique».



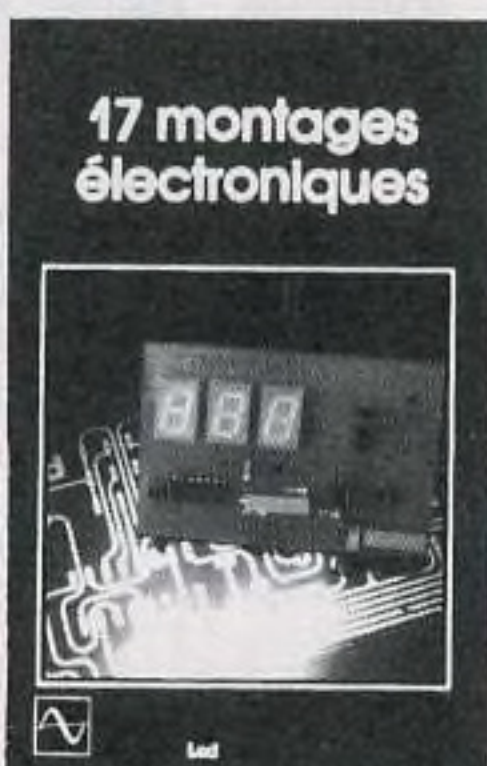
**L 10.** 200 p. Prix : 130 F TTC  
Tout beau, tout nouveau, le lecteur laser. Qu'en est-il réellement ? Pour en savoir plus, un livre traitant du sujet s'imposait. «Les lecteurs de compact-discs» permet de faire son choix parmi 37 modèles testés, analysés, examinés et écoutés.



**L 09.** 72 p. Prix : 65 F TTC  
Pour la première fois en électronique, un lexique anglais-français est présenté sous une forme pratique avec en plus des explications techniques, succinctes mais précises. Ce sont plus de 1 500 mots ou termes anglais qui n'auront plus de secret pour vous.



**L 11.** 160 p. Prix : 85 F TTC  
Finis les calculs fastidieux et erronés ! Grâce à cet ouvrage, les concepteurs d'enceintes acoustiques gagneront un temps appréciable durant la phase d'étude et de mise au point : 120 abaques et tableaux pour tous types de filtres et d'impédances de HP !

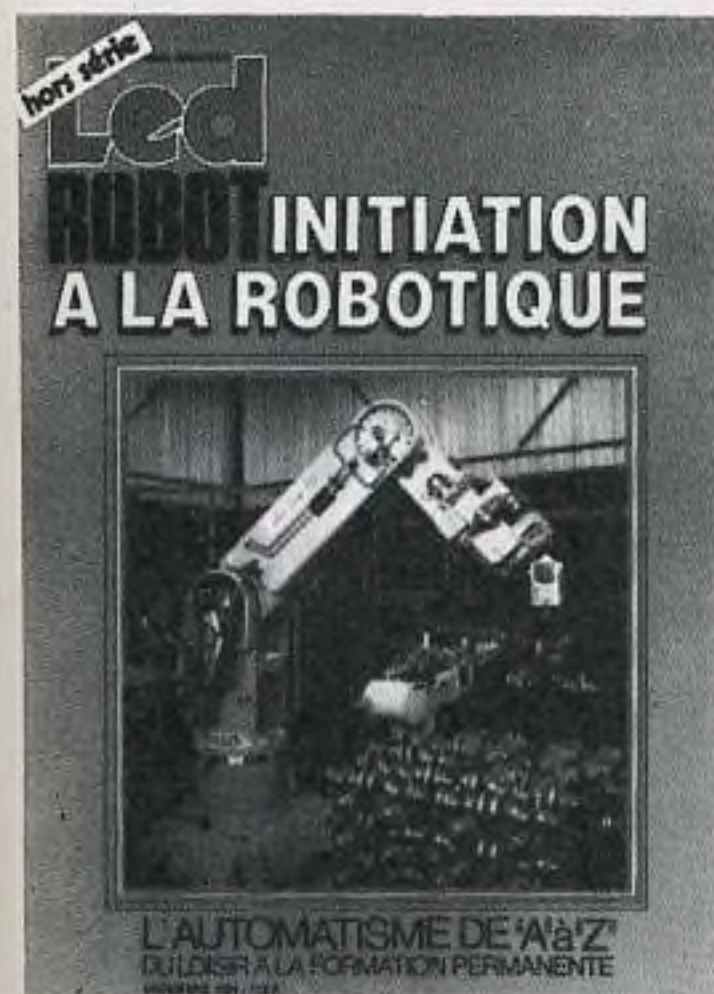


**L 14.** 128 p. Prix : 95 F TTC  
Voici enfin réunies dans un même ouvrage, dix-sept descriptions complètes et précises de montages électroniques simples. Il s'agit de réalisations à la portée de tous, dont bon nombre d'exemplaires fonctionnent régulièrement. Les schémas d'implantation et de circuits imprimés sont systématiquement publiés.

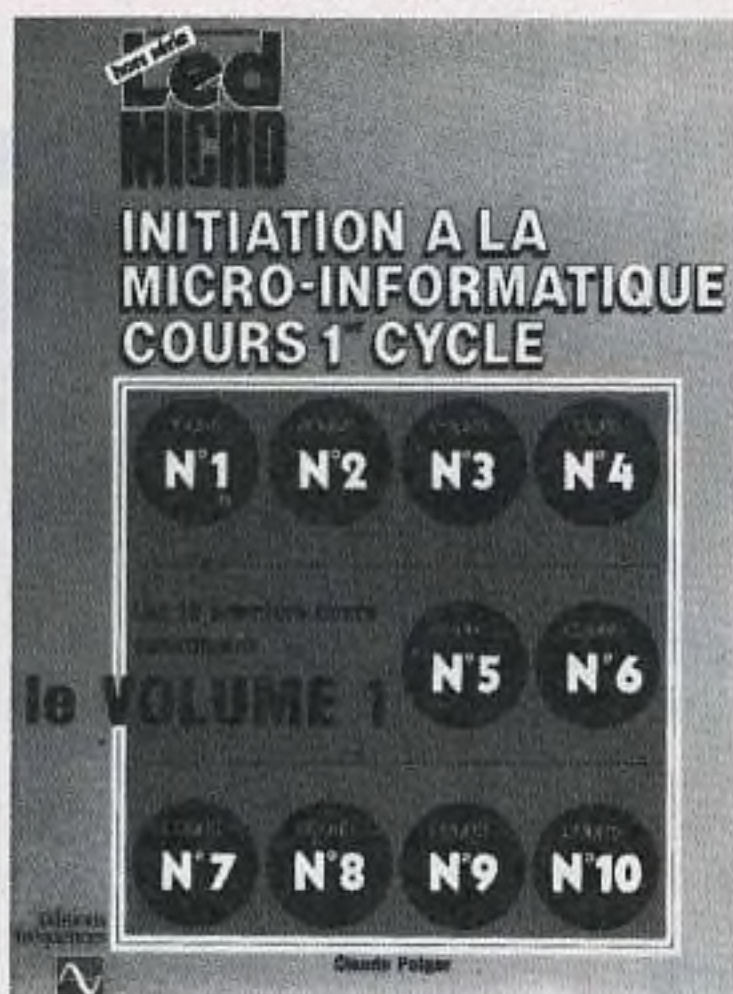


**L 20.** 208 p. Prix : 130 F TTC  
Accessible à tous, «Week-end photo» permet de découvrir de façon simple les différents aspects de la photographie actuelle. Vous y trouverez les bases indispensables pour vous perfectionner, un guide de choix des appareils 24x36 et des illustrations abondamment commentées.

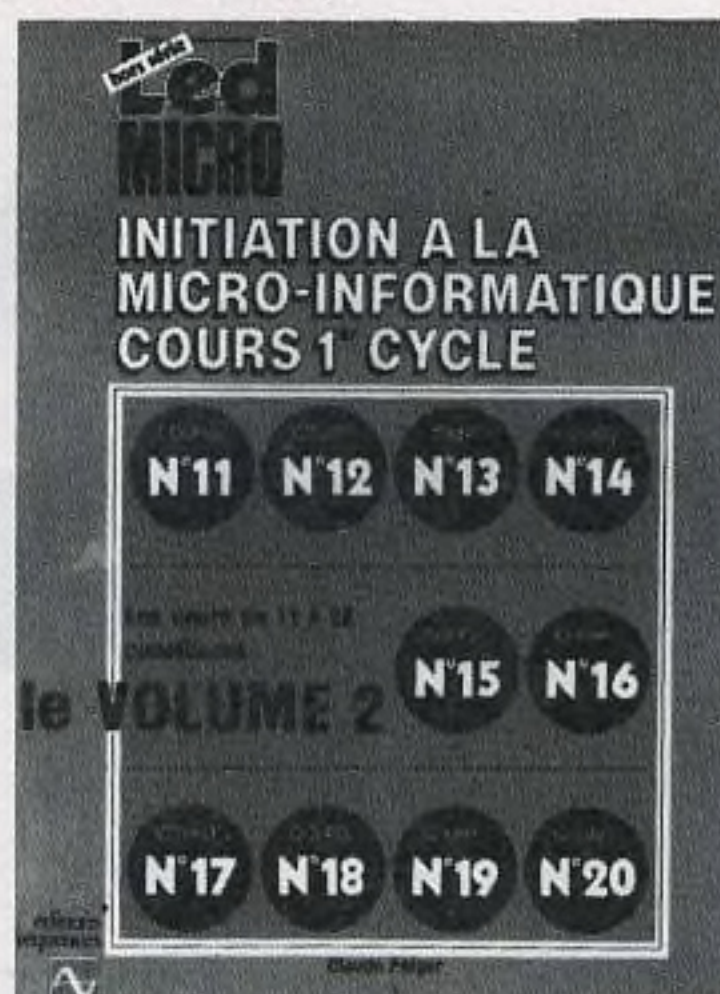
## Collection initiation (format 210 x 270)



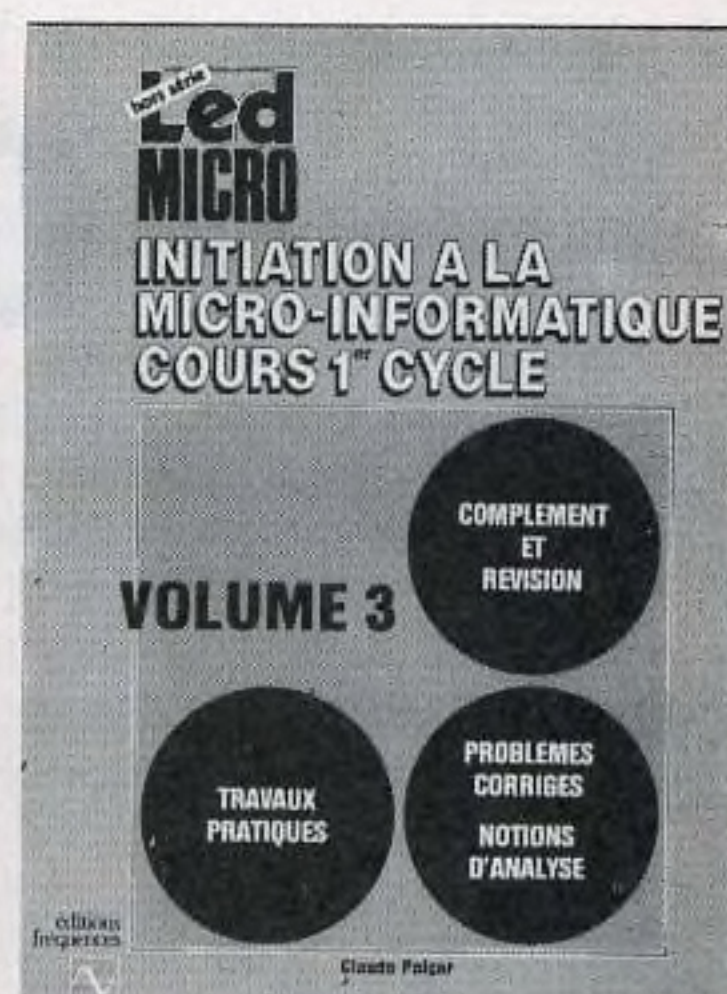
**P 08.** 96 pages. Prix : 115 F TTC  
 Cet ouvrage eut un succès retentissant dès sa sortie. Bien plus qu'un cours d'initiation, il s'agit aussi du premier recueil d'informations données par les concepteurs, les utilisateurs de robots et les fans de cybernétique, enfin réunis !



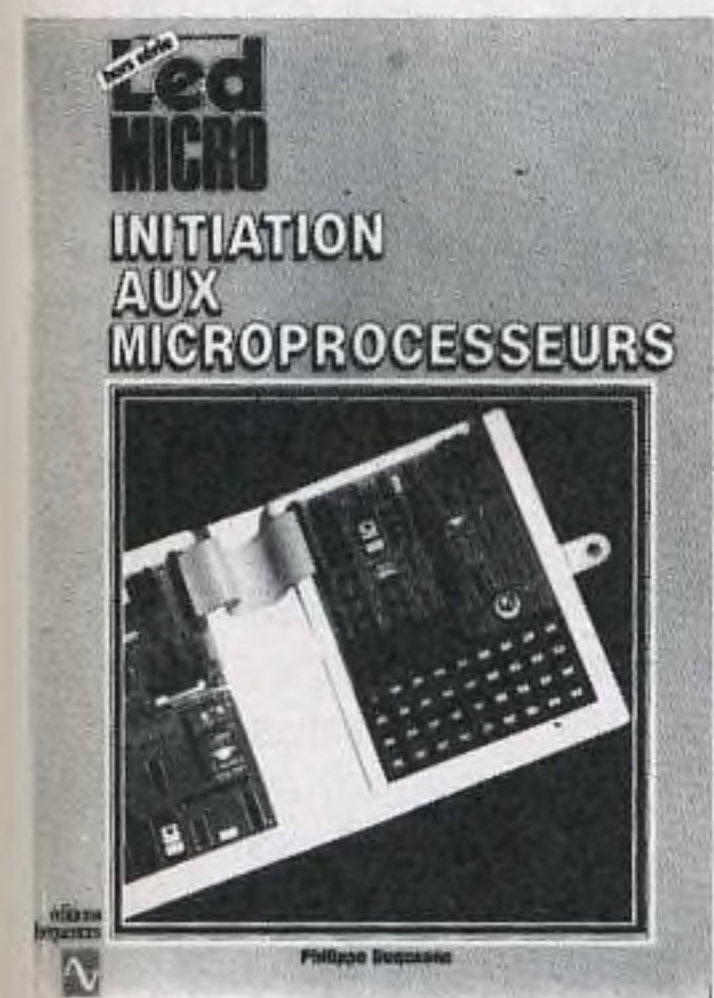
**P 16.** 272 pages. Prix : 130 F TTC  
 Passé les premiers remous de la révolution que fut l'avènement de la micro-informatique, il fallut bien tenter d'en réunir les enseignements. Une lacune apparut : celle d'un ouvrage d'initiation à la programmation, universel et complet. En voici le premier tome.



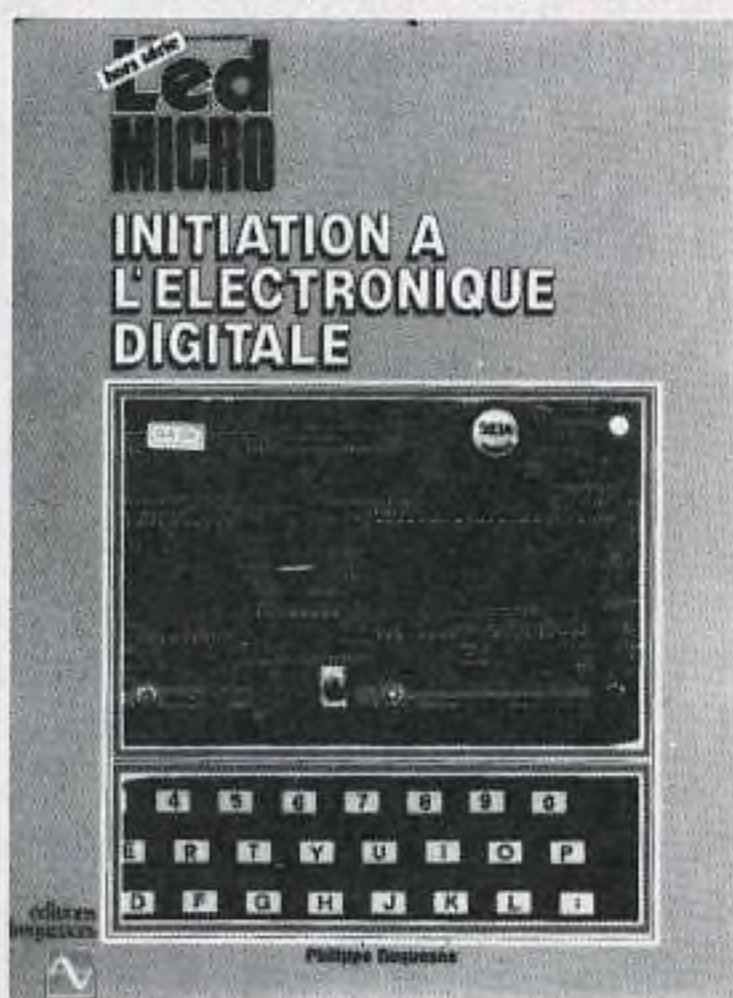
**P 17.** 208 pages. Prix : 130 F TTC  
 Le tome 2 est la suite du tome 1 : l'esprit puissamment didactique de l'auteur s'y retrouve, le contenu du livre permettra d'acquérir un niveau suffisant pour exercer l'analyse, la programmation, la gestion, l'automatisme, la simulation et d'autres choses encore !



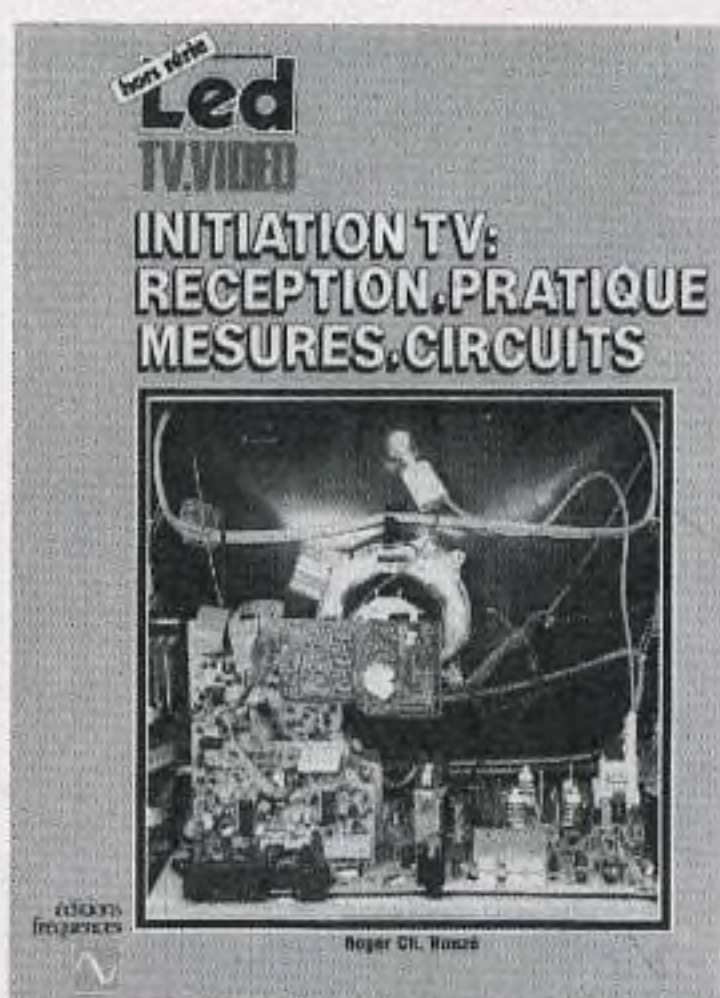
**A Paraître**  
 Le troisième volume du cours de Programmation, dû à Cl. Polgar, pédagogue apprécié de tous. Il continue dans la lignée d'un réel souci didactique, de haut niveau, maintenant, mais en conservant l'aspect progressif qui fit son succès initial.



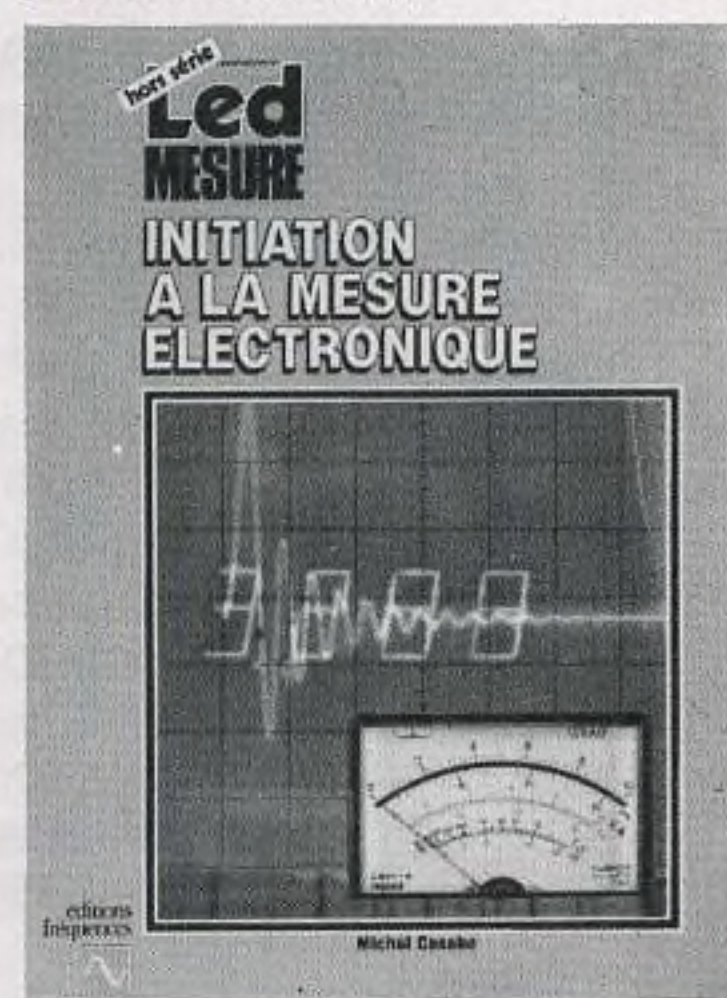
**P 18.** 136 pages. Prix : 95 F TTC  
 Du même auteur, Ph. Duquesne, on nous propose cette fois-ci, de pénétrer au cœur même de l'ordinateur, de comprendre le fonctionnement de l'élément vital qu'est le microprocesseur et enfin de maîtriser l'assembleur, langage du microprocesseur.



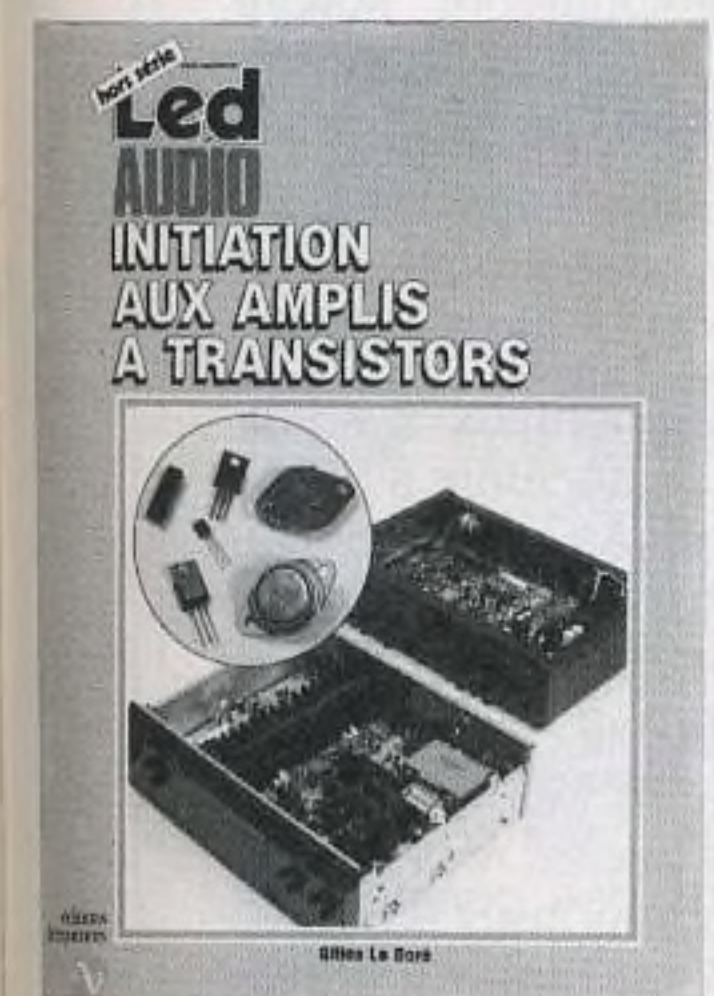
**P 19.** 104 pages. Prix : 95 F TTC  
 Ce cours d'Initiation à l'Electronique Digitale est dû à Ph. Duquesne, chargé de cours de microprocesseurs au CNAM. L'objet de cet ouvrage est de présenter les opérateurs logiques et leurs associations. La technologie est évoquée, brièvement, elle aussi.



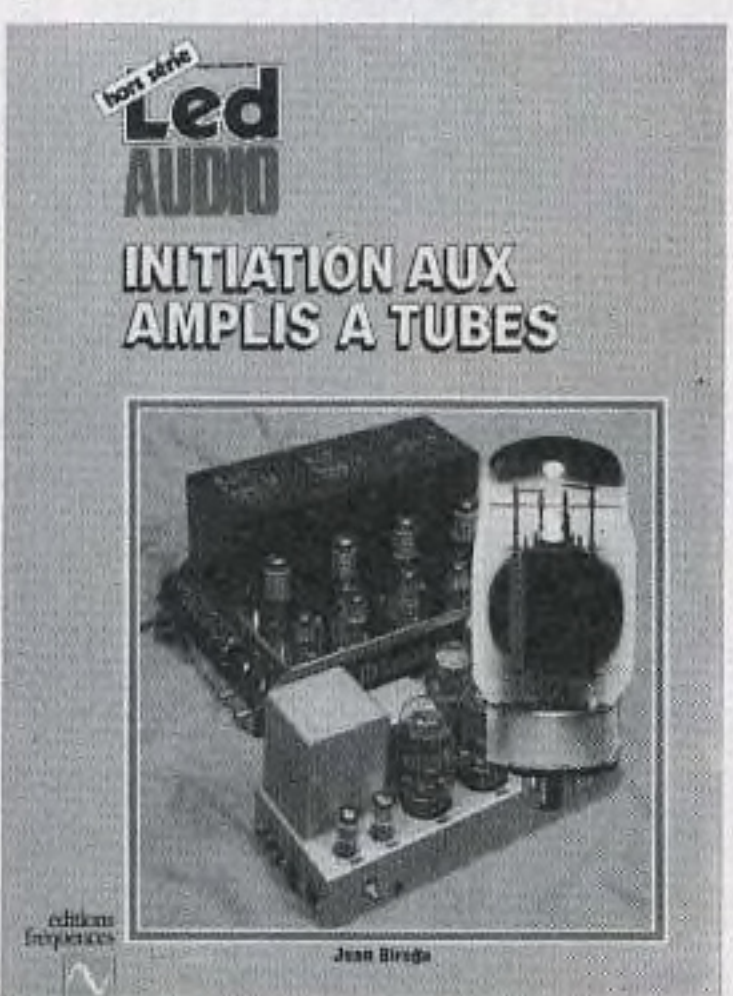
**P 21.** 136 pages. Prix : 135 F TTC  
 Issu d'un cours régulièrement remis à jour, ce livre permet à l'amateur comme au professionnel de se tenir au courant de l'état actuel de la technologie en télévision. De nombreux schémas explicatifs illustrent le contenu du livre.



**P 23.** 120 pages. Prix : 140 F TTC  
 Il n'existait pas, jusqu'à présent, un ouvrage couvrant de manière générale mais précise, l'ensemble des problèmes relatifs à l'instrumentation et à la méthodologie du laboratoire électronique. C'est chose faite aujourd'hui avec ce volume récemment paru.



**P 24.** 96 pages. Prix : 130 F TTC  
 Après un bref historique du transistor, cet ouvrage traite essentiellement de la conception des amplificateurs modernes à transistors. La théorie est décrite de manière simple et abordable, illustrée d'exemples de réalisations commerciales. Le but du livre est de donner à chacun la possibilité de réaliser soi-même son amplificateur...



**P 26.** 152 pages. Prix : 155 F TTC  
 Complémentaires des «Amplis à transistors», les «Amplis à tubes» sera certainement une petite encyclopédie sur ce sujet : historique, mais aussi polémique, puisque les tubes sont encore d'actualité et parce que les arguments en faveur de cette technique et ses défenseurs sont encore nombreux.

Diffusion auprès des libraires assurée exclusivement par les Editions Eyrolles.

Bon de commande à retourner aux Editions Fréquences  
 1, boulevard Ney 75018 Paris

Je désire recevoir le(s) ouvrage(s) ci-dessous référencé(s) que je coche d'une croix :

- |                               |                               |                               |                               |                               |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| E 01 <input type="checkbox"/> | E 02 <input type="checkbox"/> | E 03 <input type="checkbox"/> | E 04 <input type="checkbox"/> | E 05 <input type="checkbox"/> |
| E 06 <input type="checkbox"/> | L 07 <input type="checkbox"/> | P 08 <input type="checkbox"/> | L 09 <input type="checkbox"/> | L 10 <input type="checkbox"/> |
| L 11 <input type="checkbox"/> | E 12 <input type="checkbox"/> | E 13 <input type="checkbox"/> | L 14 <input type="checkbox"/> | E 15 <input type="checkbox"/> |
| P 16 <input type="checkbox"/> | P 17 <input type="checkbox"/> | P 18 <input type="checkbox"/> | P 19 <input type="checkbox"/> | L 20 <input type="checkbox"/> |
| P 21 <input type="checkbox"/> | E 22 <input type="checkbox"/> | P 23 <input type="checkbox"/> | P 24 <input type="checkbox"/> | P 26 <input type="checkbox"/> |

Frais de port : + 10 F par livre commandé, soit la somme totale ci-jointe, de Frs \_\_\_\_\_ par  
 CCP  Chèque bancaire  Mandat-lettre

Nom ..... Prénom .....  
 Adresse .....  
 Ville ..... Code postal .....

# C'EST ARRIVE DEMAIN



(en direct de notre envoyé permanent dans la Silicon Valley)

«Commodore c'est fini». Allons-nous entendre ce cri sous peu ? C'est ce qui risque de se produire si les banques refusent de reculer les échéances qu'affronte cette firme pourtant très connue. Le magazine Time allait jusqu'à titrer «Adios Amiga», ce qui fait preuve d'un humour morbide de mauvais goût. La faillite de cette société serait vraiment un coup dur, au moment où l'Amiga semblait reprendre le flambeau du Commodore 64, l'une des bonnes ventes d'ordinateurs dans le monde. De plus, cela serait de mauvais augure pour les possesseurs de cet ordinateur. Les magasins diffusant l'Amiga ici disent que les ventes sont toujours très élevées, mais il faut faire la part de la publicité destinée à éviter la fuite des clients vers des marques en meilleure santé. Les banques ont déjà accordé un sursis de un mois, mais peu

de gens pensent qu'elles le prolongeront. Malgré cela, les développeurs déclarent continuer de travailler pour l'Amiga. Alors l'avenir très proche nous dira si «Commodore c'est fini».

Pour concurrencer les extrêmes-orientaux sur leur terrain, les fabricants américains se décident à développer des usines de production en Asie du Sud-Est. Une compagnie, spécialisée dans les cartes d'extension pour IBM PC vient de montrer le (mauvais ?) exemple. Classic Technology Corp. a annoncé que sa production serait désormais assurée outre-mer. Ceci lui permettra de diminuer ses coûts de main-d'œuvre de 20 % et le prix des composants



qu'elle achète, de 10 % au moins. C'est vraisemblablement l'annonce d'un mouvement de la plupart des fabricants à court terme. L'Amérique dynamique commence sérieusement à se poser des questions sur ses possibilités de réponses, aux Japonais et autres Coréens. Si l'Amérique n'est plus sûre d'elle, que restera-t-il aux Américains ? Le Coca-Cola ? Piètre refuge.

De plus en plus de développeurs diffusent simultanément leurs logiciels sous deux grands systèmes d'exploitations. MS-DOS, pour IBM et compatibles, et Macintosh. Pour qu'ils soient amenés à le faire, il faut que le Macintosh commence à se vendre fort bien ici. En effet, cette diversification des logiciels fait de facto du Macintosh un standard en micro-informatique, ce que personne ne voulait reconnaître jusqu'à maintenant. Le plan de vente d'Apple serait-il en train de s'améliorer ? En tout cas, cette société investit pour le futur, car elle vient d'acheter un exemplaire de l'ordinateur le plus puissant existant, le Cray X-MP/48, pour l'aider à des simulations concernant les futurs Apple. Au prix au kilo, c'est peut-être une bonne affaire, mais je crains que ni vous ni moi ne puissions nous en offrir jamais un.

IBM et Apple continuent d'avoir des concepts commerciaux qui se trouvent aux antipodes les uns des autres. Pour IBM, un produit ne peut être retiré de la vente avant au moins six ans de bons et loyaux services. Chez Apple, en moins de deux ans, nous avons vu apparaître et disparaître le Lisa, le Mac 128, le Mac 512. Un collègue a proposé comme comparaison l'industrie automobile, et en tire une conclusion peu encourageante pour IBM. Il considère que Apple fait par là la preuve de son dynamisme et de ses facultés d'adaptation au marché, comme les constructeurs japonais d'automobiles l'ont fait ces dernières années, alors qu'IBM joue le monolithisme, comme les constructeurs américains le faisaient il y a encore peu. Moralité, le marché américain est envahi de voitures japonaises, et les géants américains voient leurs marges chuter. Pour y remédier, certains vendent des japonaises sous leur nom comme le fait Dodge. Si la comparaison devait s'avérer juste, ce serait la revanche de l'imagination sur la puissance commerciale. Il faut noter d'ailleurs qu'IBM accepte de moins en moins la part (en augmentation ininterrompue) de marché occupé par les compatibles. Pour cela, IBM modifie régulièrement certaines caractéristiques de ses produits (comme vient de le prouver l'apparition récente du RT), et baisse ses prix (le même RT est proposé avec 40 % de réduction si le client en achète 25). Si cela vous intéresse, voyez IBM directement. Il n'est que temps, car les compati-

bles valent actuellement de 15 à 40 % de moins qu'un IBM équivalent. Il est difficile de survivre avec de tels écarts de prix, et même le nom ne suffit plus comme avant, à faire vendre le PC original. A ce propos, le patron d'IBM vient de déclarer lors d'une conférence de presse que le prix étant le premier critère de choix d'un ordinateur de type micro, il pensait que la société qu'il dirige allait baisser ses prix pour devenir compétitif. IBM s'inquiétant de ses clients, c'est le monde à l'envers. Ainsi, la puissance et le nom de cette maison ne suffisent plus à imposer le produit. J'aurai tendance à m'en réjouir, car je suis de ceux qui pensent qu'IBM a eu bien peu de mérite à imposer un standard dépassé dès sa sortie (et cela fait 4 ans maintenant), et sans prestige, sur le seul nom d'IBM. Maintenant, les compatibles font mieux et plus vite. Pour 40 % de moins. Longue vie à IBM, mais avec des produits tournés vers l'avenir, pour changer.

Nous voyons apparaître des disques durs pour ordinateurs personnels, comme Apple II et Commodore 64. C'est une évolution fantastique, lorsque l'on pense que cette technique ne s'adressait qu'à des systèmes de plus de 40 000 francs, il y a moins de deux ans. Bien sûr, un tel achat représente encore une petite fortune, puisque le prix de base tourne autour de 15 000 francs, mais il permet à toutes les petites sociétés de s'équiper pour moins de 30 000 francs de systèmes performants, rapides et à grandes capacités de stockage. Pour gérer un cabinet libéral, ou les comptes d'une petite entreprise, ceci semble l'idéal, compte tenu de la très grande diversité de programmes disponibles pour ces ordinateurs. De tels disques existaient déjà, mais à des prix nettement plus élevés.

Pour la première fois, une étude découvre que si une entreprise ne forme pas correctement ses employés à une utilisation rationnelle de ses ordinateurs, le résultat peut être à l'inverse des souhaits, c'est-à-dire une baisse de productivité des employés. Il était temps que quelqu'un se rende à l'évidence, et il aura sans doute fallu attendre cette étude pour que les responsables de formations dans les entreprises ne voient plus l'informatisation comme un doux nuage dans un ciel uniformément bleu. En effet, l'étude est produite par le respecté MIT (Massachusetts Institute of Technology), et n'en aura que plus de poids. Mais il semble qu'ici, comme en France, le passage au calculateur ne se soit pas toujours produit sans grincement de dents. Il est fréquent de voir des ingénieurs passer trois mois à comprendre un environnement, ou un langage, par manque de formation, puis pour cause de formation inadaptée dans l'entreprise. Le plus incroyable est qu'il aura été nécessaire d'atten-

dre jusqu'à maintenant pour qu'une étude sur ce problème soit diffusée. A moins que ce ne soit qu'il ait été indispensable de produire une telle étude pour que certains ne prennent conscience du danger.

Microsoft vient de sortir le logiciel Windows, qui est un environnement et un système d'exploitation du type Macintosh. Pour moins de 75 \$, soit à peu près 500 francs, il est possible de transformer un IBM PC et compatibles en ordinateur convivial. Il est livré avec les logiciels Window Write, et Window Paint, et ceci représente une bonne partie des programmes disponibles sous l'environnement. L'analogie avec le Mac devient vraiment criante, lorsque l'on sait que le Mac est livré avec Mac Write et Mac Paint.

Le problème majeur de Window est la place mémoire dont il est plus que friand. Les publicités déclarent qu'il demande 256 K minimum, mais c'est en fait un AT sous son meilleur jour, et avec un disque dur, qui est requis pour ne pas passer des heures à regarder un écran et à écouter le disque tourner. Les essais que j'ai pu voir sur des XT sont des plus décevants à ce sujet, et il n'est pas possible de le conseiller pour cet ordinateur. La gestion de disques est très lente, ainsi que celle de l'écran, qui peut prendre jusqu'à 5 secondes à chaque fois. Autrement dit, c'est à la limite de l'inutilisable. Par contre, sur un AT, en fait sur un compatible AT, les performances sont vraiment honnêtes. Agréable d'usage, souple et assez rapide, l'AT devient un outil sympathique. Bel effort réalisé par le programme !!! Donc, un grand programme pour un AT, mais un veau sur XT et en-dessous.

Le développement des systèmes experts est fascinant à suivre. Il y a deux ans, peu de gens connaissaient le mot, et 1 % d'entre eux savait de quoi il retournait. Maintenant, ces programmes sont en pleine expansion, et il sort tous les mois. Il faut dire que ce qui était une gageure sur les micros de l'époque («époque» qui date de deux ans **au moins**) devient possible sur des petits monstres du type ordinateurs à plus de 1 méga en mémoire centrale, avec disque dur, processeurs rapides et spécialisés, etc. On arrive à des programmes d'intelligence artificielle commercialisables et rentabilisables, ce qui est un paramètre nouveau. Il n'est plus nécessaire d'attendre des heures le résultat de quelques inférences (le mécanisme de déduction et d'enchaînement de ces programmes) et l'environnement n'est plus spartiate comme avec les premiers qui ne présentaient que l'intelligence artificielle sans la présentation, ce qui les rendait à la limite de l'utilisable. Maintenant, vous disposez de beaux graphiques, de menus, d'aides en direct, de tout ce qu'un bon programme du commerce doit posséder pour justement mériter ce

qualificatif. Le nouveau IBM est un support performant pour ce type de programmes, et beaucoup de développeurs ont commencé d'ores et déjà à produire des versions de leurs programmes pour cet ordinateur. La diversité des programmes fait enfin plaisir à voir. Nous ne sommes plus restreints à la fécondation des tomates, aux théorèmes mathématiques, et autres programmes tests pour laboratoires. Il en existe maintenant qui corrigent les fautes d'orthographe au moment de la frappe d'un texte, d'autres qui aident à la préparation de bordereaux de vente, .... Seule l'imagination bride maintenant leur développement.

Nous avons déjà parlé de Commodore, passons à Osborne. Qui se souvient de cette marque, qui lors de la préhistoire de la micro-informatique, il y a bien six ans, avait inventé le micro portable ? L'idée en avait fait des milliardaires. Maintenant, les créanciers ont obtenu la liquidation de la société. Le portable a tellement évolué en six ans que l'Osborne 1 fait figure d'ancêtre, et les nouveaux produits Osborne ne proposaient pas de grandes innovations. Les portables sont maintenant des PC compatibles, eux aussi, disposant d'un lecteur de disquettes, d'un écran plat de 20 lignes, etc. Mais surtout, ils ne se sont pas développés autant que prévu. La puissance sans cesse accrue des micros de table ne se retrouve sur les portables qu'avec trop de retard pour intéresser le client potentiel, qui est en général un cadre, amoureux de la technique et du modernisme. De plus, les portables sont souvent proscrits en avion, en raison des interférences qu'ils créent parfois et qui peuvent être gênantes pour les contacts entre celui-ci et le sol. Il ne reste plus que l'utilisation chez soi, et le plus souvent un micro de bureau est préféré. Alors, l'Osborne et ses congénères sont voués à la disparition progressive.

Vous avez un Atari 520 ST. Et bien, pour le même prix, ou presque, vous auriez pu acheter un 1040 ST, qui est un 520, mais avec deux fois plus de mémoire. Le reste est identique, à des détails près. Mais, intérieurement, que de changements. Le 1040 est plus modulaire et extensible à 4 mégas. Le 520 est (ou sera) transformable en 1040. Atari annonce à l'occasion de la sortie du petit (grand) dernier que plus de 80 logiciels existent pour eux d'ores et déjà. Cela devrait dynamiser la vente de cet excellent ordinateur, mis à part des défauts de jeunesse. Le 1040 est livré avec GEM, dont nous vous avons déjà parlé. Enfin, on attend la sortie d'un circuit spécialisé se chargeant de graphiques et accélérant les opérations effectuées à l'écran, jusqu'à 20 fois.

# **COURS DE GENIE LOGICIEL**

## **De la théorie à la pratique**

**Charles-Henry Delaleu**

Bien que ce chapitre ne fut pas prévu au départ, il est intéressant d'aborder les problèmes liés à l'informatique industrielle. En effet, l'informatique industrielle est, sans aucun doute, une des branches les moins connues. Toutefois, elle fait appel à des notions qu'il est souhaitable de connaître.

Avec la téléinformatique, l'informatique industrielle nécessite des connaissances en électronique et en sécurité. Cette dernière ayant pour but de contrôler, de mesurer, d'asservir des systèmes, il ne sera pas question de réaliser un site en informatique industrielle, comme un simple site de calcul. Si les premières installations furent exécutées par l'industrie pétrolière (raffinerie), il convient de noter que les progrès réalisés en informatique permettent, actuellement, d'implanter un processus informatisé dans la plupart des sites où le besoin s'en fait sentir. Depuis quelques années, l'apparition des automates industriels a provoqué de nombreux bouleversements dans le domaine de la régulation. L'automate industriel (calculateur industriel simplifié, de taille très réduite) est devenu un investissement tout à fait abordable. Certains modèles coûtent même beaucoup moins cher qu'un micro-ordinateur de type PC.

## SYSTEMES EN INFORMATIQUE INDUSTRIELLE

La différence essentielle entre un système informatique conventionnel et un système d'informatique industrielle consiste, pour ce dernier, à être équipé d'un nombre non négligeable de cartes d'entrées-sorties. On appellera cartes d'entrées-sorties des cartes permettant à l'ordinateur de dialoguer avec les grandeurs physiques qui l'entourent. Si une interface classique (RS 232, Centronics, IEEE 488, etc.) autorise un dialogue avec un périphérique tel qu'une imprimante ou une unité de mémoire de masse, une carte d'entrées-sorties industrielle permet le dialogue entre un calculateur et un ensemble de capteurs ou d'effecteurs positionnés sur des machines de fabrication, des appareils de mesure, etc.

### UN CAPTEUR

Un capteur est un composant qui est en mesure de fournir une information sur une grandeur donnée : un microphone est un capteur qui délivre une tension électrique proportionnellement à une pression acoustique. Un accéléromètre fournira une information concernant une pression mécanique. Il existe, dans les domaines du contrôle et de l'industrie, un nombre considérable de capteurs permettant d'obtenir des informations précises sur presque la totalité des grandeurs physiques dont l'homme a besoin.

### UN EFFECTEUR

On appellera effecteur un objet piloté par ordinateur capable de modifier le comportement d'un processus industriel.

Les effecteurs les plus connus sont :

- les verrins
- les moteurs électriques
- les relais
- les électro-aimants
- les électro-vannes, etc.

### PROCESSUS

Un capteur fournit à l'ordinateur une information sur l'état d'un système de fabrication. Le calculateur analyse cette information et la quantifie. Si elle n'est pas conforme, il adresse à un effecteur l'ordre de modifier une séquence ou un réglage. Dès lors, le système de fabrication retourne immédiatement en configuration standard.

Ex : Dans un haut-fourneau, le minerai est mis en fusion par un four électrique permettant ainsi d'obtenir du métal. Nous avons, d'un côté, des résistances électriques permettant le chauffage de la cuve où est placé le minerai. D'autre part, nous avons des thermomètres nous indiquant la valeur de la température dans la cuve. Un capteur thermique va transmettre par l'intermédiaire d'un convertisseur via une carte d'entrées-sorties à un calculateur, une information sur la température. En fonction de l'étape du processus de fusion, le calculateur va transmettre une information électrique aux résistances chauffantes permettant ainsi un réglage parfait de ces derniers. Il y a optimisation du processus de coulée du métal grâce à un système d'informatique industrielle.

## L'INFORMATIQUE INDUSTRIELLE AUJOURD'HUI

Les domaines de l'informatique industrielle sont de plus en plus nombreux. Ils sont présents dans les automatismes, les appareils de mesures, la sécurité.

### LES AUTOMATISMES

Dans les automatismes entrent de nombreuses techniques :

- les asservissements,
- les machines-outils,
- les contrôles processus,
- etc.

### LES ASSERVISSEMENTS

Il existe un nombre considérable de types d'asservissement. Qu'il s'agisse de contrôler le nombre de tours-minute effectué par un moteur électrique, la température d'un haut-fourneau ou la position d'une commande mécanique, l'informatique simplifie la tâche.

Il n'est pas question ici de développer les asservissements, mais en simplifiant le problème, il est possible de l'expliquer ainsi :

Un capteur détecte : le nombre de tours par minute, la température, la position angulaire de l'objet à contrôler. L'information est envoyée à un convertisseur analogique-digital qui numérise le signal analogique pour qu'il soit compréhensible par un ordinateur. Ce dernier quantifie l'information et la compare avec ses propres valeurs pré-programmées. Si les données sont correctes, le système ne réagit pas. Si la valeur émise par le capteur est non conforme, le calculateur émet un ordre de correction. Ainsi, une commande est envoyée au moteur, aux résistances électriques du four, à un vérin, pour ramener l'objet à contrôler dans une position normale. Le système est ainsi bouclé, il y a asservissement.

### LES MACHINES-OUTILS

Les machines-outils ont subi, grâce à l'informatique, une véritable mutation. Il s'agit de la numérisation de leurs commandes. C'est ainsi que les tours et autres fraiseuses de haut de gamme, se sont transformés en tours à commandes numériques, en fraiseuses à commandes numériques...

Ici, il n'y a plus d'opérateur actionnant des leviers ou des manivelles, mais un technicien qui programme par avance un automate ou un micro-ordinateur qui, par la suite, enverra des ordres à toutes une série d'actionneurs (d'effecteurs) qui permettront un façonnage des pièces à réaliser en grande quantité.

### LES CONTROLES PROCESSUS

Si les machines se sont numérisées, les ateliers sont de plus en plus contrôlés, voire gérés par des ordinateurs. Une machine de base est pilotée par un automate programmable, un groupe d'automates est relié à un micro ou un mini-calculateur. L'ensemble des micro ou mini est connecté à un ordinateur central. Cette technique autorise un parfait contrôle d'un vaste processus industriel de fabrication.

La répartition des tâches et des contrôles par niveau hiérarchique permet une meilleure souplesse et évite un blocage total de l'atelier ou de l'usine à piloter (voir les ateliers flexibles).

### LES APPAREILS DE MESURE

L'informatique a aussi fait son entrée dans le monde de l'instrumentation. Il existe actuellement de grandes familles d'appareils de mesure :

- les appareils de mesure analogique conventionnels,
- les appareils de mesure numérique interfaçables à un calculateur.

Cette dernière catégorie d'appareils a vu arriver son heure de gloire grâce à la société Hewlett-Packard qui développa pour son usage personnel l'interface GPIB. Cette

interface allait très vite être normalisée par différentes instances telles que l'IEEE, l'IEC, etc. Grâce à cela, plusieurs appareils peuvent être reliés ensemble et s'échanger toutes sortes d'informations, mais aussi être contrôlés à distance. Un laboratoire se retrouvera donc équipé d'une série d'appareils de mesure interfacés contrôlés par un ordinateur. Ceci permet la réalisation de tests en automatique, mais aussi l'ouverture de nouveaux horizons. C'est ainsi que dans certains domaines, de nouveaux types de mesure peuvent être réalisés en mélangeant mesure, calcul et représentation graphique en deux ou trois dimensions. Actuellement, nous assistons aussi à une situation assez intéressante ou, dans les faits, plutôt que de réaliser un appareil de mesure classique avec commande numérique, certains fabricants préfèrent réaliser des micro-ordinateurs ayant la fonction d'un appareil de mesure. En électronique de haut niveau, la plupart des appareils de mesure sont, en réalité, des calculateurs faisant office d'appareils de mesure. Cela permet des fonctions beaucoup plus intéressantes et finalement un coût de revient plus faible.

### LA SECURITE

Depuis quelques années, l'informatique industrielle fait une entrée en force dans les domaines de la sécurité. Des ensembles tels que le Forum des Halles ou le Musée des Techniques de la Villette à Paris sont contrôlés par informatique. Les ascenseurs, la ventilation, la climatisation, le chauffage, l'éclairage, les accès, tout est contrôlé, géré par informatique. On retrouve ici le même système hiérarchique que pour le contrôle processus, avec un ordinateur central, et des mini ou micro-esclaves. Grâce à des capteurs très nombreux, grâce à une gestion graphique du contrôle, dès qu'un escalier roulant s'arrête, dès qu'une ventilation se bloque, le poste de sécurité central est averti. Il ne reste plus, à l'aide d'un zoom graphique qu'à détecter la position exacte de l'objet en panne (utilisation de page vidéo comme plan) et de signaler le point exact de la panne aux services techniques pour une intervention immédiate.

### LA NAVIGATION

Qu'il s'agisse d'avions, de bateaux, de trains (TGV, VAL, etc.), les moyens de transport modernes sont aujourd'hui largement informatisés. Dans le VAL (métro de Lille), les rames sont pilotées automatiquement. Dans le TGV, les trains sont contrôlés par un dispositif s'ajoutant à la conduite du machiniste, ce qui évite tout débordement vis-à-vis de la sécurité. Quant aux avions, il y a longtemps qu'ils possèdent toutes sortes d'asservissements, de contrôles et de dispositifs d'aide à la navigation et au pilotage.

### DIVERS

Dans le monde médical, il existe des systèmes d'analyse automatique, des systèmes de soins et de surveillance de malades gérés et contrôlés par ordinateur. Dans les banques, les systèmes de retrait d'argent à distance en-dehors des ouvertures des guichets sont des systèmes d'informatique industrielle.

### LES PREMIERS SYSTEMES D'INFORMATIQUE INDUSTRIELLE

Dans le premier système d'informatique industrielle, il n'existait pas de cartes d'entrées-sorties spécialisées. En fait, une grande partie de l'espace-mémoire RAM était transformée en connexions d'entrées-sorties. Pour envoyer un ordre, il suffisait de pointer une certaine adresse mémoire et de lui envoyer l'ordre voulu traduit par un octet.

### LES CARTES D'ENTREES-SORTIES

Les calculateurs d'informatique industrielle possèdent tous aujourd'hui un bit de commande permettant d'avoir un espace mémoire RAM classique et un espace d'entrées-sorties. Cet espace d'E/S est implanté sous forme de carte d'entrées-sorties spécialisées. Il s'agit en fait de bus d'entrées-sorties parallèles possédant

toutes sortes de lignes :

- un bus d'entrées parallèles,
- un bus de sorties parallèles,
- un bus de commandes.

Chacune de ces cartes est appelée par un numéro qui lui est propre. Le bus de sortie peut être équipé de sorties de puissance afin de pouvoir attaquer directement certains dispositifs de commande.

Dans les systèmes évolués, il est possible de commander ainsi toutes cartes d'entrées-sorties avec un maximum de souplesse.

Des ordres précis permettent un contrôle de la carte, des différents registres, des interruptions, etc.

## LES DISPOSITIFS SPECIALISES

**LA DMA.** Dans certaines applications, le calculateur doit réaliser des entrées à très hauts débits. Le processeur n'est alors pas assez rapide. Dès lors, le chargement de ces informations se fait suivant le dispositif de la DMA (Direct Memory Acces).

Les informations, lors du stockage, ne transitent plus par le processeur mais vont être directement stockées en mémoire vive et être reprises ensuite pour les calculs.

## LES PROCESSEURS D'ENTREES-SORTIES

Outre la DMA, il existe aussi les processeurs d'entrées-sorties. Dès lors, le processeur central s'occupe des calculs et un co-processeur uniquement des entrées-sorties.

## MEMOIRES-TAMPONS

Dans les calculateurs lents n'ayant ni DMA, ni processeur d'E/S, il est possible d'insérer entre le calculateur et le périphérique, une mémoire tampon, d'où le calculateur reprendra les informations dont il a besoin à son rythme.

## UNE CARTE D'ENTREES-SORTIES PARALLELE : DESCRIPTIF DU DISPOSITIF

1. Toutes les entrées, sorties ou commandes peuvent être programmées en logique positive ou négative.
2. Des petits switches permettent de programmer l'adresse de la carte pour lui donner un code afin de faciliter son interrogation.
3. Il est possible de sélectionner diverses priorités en mode d'interruption.
4. Il est possible de modifier légèrement le chronogramme des échanges.
5. Sur certaines cartes, les vitesses d'acquisition peuvent être ralenties.
6. Un bit de contrôle permet de savoir si le dispositif avec lequel la carte d'E/S doit dialoguer est sous tension.
7. Il existe des bits de RESET et de PRESET autorisant une remise à zéro des dispositifs.
8. Un bit indique si la carte est en entrée ou en sortie.
9. Plusieurs bits contrôlent les échanges de données :
  - bus libre,
  - début d'interrogation,
  - fin d'interrogation.
  - etc.
10. Certaines cartes disposent en plus du bus d'entrées et du bus de sorties de bits supplémentaires en entrées et en sorties permettant d'augmenter le contrôle des dispositifs externes.

## LES INTERRUPTIONS

Chaque carte d'entrées-sorties possède donc son dispositif d'interruption. Il s'agit d'une commande qui, dès qu'elle est activée, signale au calculateur qu'une carte d'E/S doit recevoir une information. Dans ce cas, il ne s'agit pas d'un simple échange,

mais du signalement d'une anomalie. En activant ainsi sa demande d'interruption, la carte signale au processeur central d'arrêter momentanément ses calculs pour analyser les causes de la demande d'interruption. Exemple : Un voltmètre central indique la tension issue d'un alternateur. Toutes les 30 secondes, l'ordinateur lit la grandeur de cette tension par l'intermédiaire d'un convertisseur analogique-digital connecté à une carte d'entrées-sorties. Parallèlement, un dispositif de détection de maximum est branché. Si, entre deux lectures, l'alternateur délivre une tension trop grande, le détecteur bascule une porte logique qui actionne en interruption l'interface puis le calculateur. Ainsi, le contrôle détecte immédiatement une anomalie.

### LOGIQUE POSITIVE - LOGIQUE NEGATIVE

Dans la majorité des cas, un ordinateur, ou tout autre appareil numérique, est câblé en logique positive. En ce qui concerne les interfaces et les nappes de liaisons pour les cartes entrées-sorties, la logique négative est tout à fait souhaitable. En effet, dans le cas d'une logique positive, si un câble ou si la nappe reliant le processus au calculateur est coupé, il y a de fortes chances que la panne ne soit pas découverte immédiatement.

Dans le cas d'une logique négative, la carte d'E/S via le calculateur détecterait sans délais un mauvais fonctionnement.

Nous savons qu'une carte d'E/S possède un bit qui contrôle la mise sous tension de l'appareil à contrôler. Dans le cas d'une coupure de ce fil, la tension aux bornes de la carte d'E/S sur ce point serait à +5 V. De ce fait, tout échange serait interrompu et l'alarme serait donnée.

Dans le cas du bus des DATA en entrée ou en sortie, une coupure ou un débranchement du câble aurait comme conséquence, de mettre en lecture (calculateur) tous les bits à zéro.

### BYTE ET WORD

Nous savons que le mot octet signifie 8 bits ; en parallèle il peut se dire BYTE. Dans le cas des machines 16 bits, le mot byte est remplacé par WORD.

### CONCLUSION

L'informatique industrielle, pour être efficace, demande une grande rigueur ainsi que des connaissances certaines :

- en informatique,
- en électronique,
- en control process.

Le fait qu'un calculateur soit utilisé en contrôle sécurité, en control process, en asservissement, augmente considérablement les problèmes liés à la sécurité. Un effort tout particulier doit être réalisé en ce qui concerne les régimes d'interruption. Le programme devra être très largement modulaire, l'assistance et les modifications doivent être facilement réalisables.

Il est préférable de simplifier les tâches. Mieux vaut utiliser plusieurs petits calculateurs qu'une grosse machine. Plus qu'ailleurs, les logiciels seront architecturés à partir de modules, de bibliothèques, de routines simples, précis, interchangeables.

Les notions de vitesse d'exécution seront optimisées. Toutes les détections d'erreurs, calcul, machine, etc., devront être gérées et ne pas bloquer le déroulement du programme principal.



<b>LISTE DES PRINCIPAUX CAPTEURS</b>
--------------------------------------

## Capteurs optiques :

Cellule photo-conductrice  
 Photodiode  
 Phototransistor  
 Capteur photoémissif  
 Détecteur thermique

## Capteurs de température :

Thermométrie par résistance      Transistor  
 Thermocouple                              Quartz  
 Diode    Thermométrie par bruit de fond

## Capteurs de position et de déplacement :

Potentiomètre      Roue codeuse  
 Capteur inductif      Capteur à propagation d'ondes élastiques  
 Capteur capacitif      Capteur de proximité

## Capteurs de déformation :

Jauges résistives                          Piezorésistances

## Capteurs tachymétriques :

Tachymètre électromagnétique  
 Tachymètre de vitesse angulaire à impulsions  
 Gyromètres

## Capteurs de force, pesage, couple :

Capteur piézoélectrique  
 Capteur à jauges d'extensiométrie  
 Capteur de force par mesure de déplacement  
 Capteur de couple  
 Capteur tactile

## Capteurs d'accélération, vibration, choc : Accéléromètre

## Capteurs de vitesse, débit, niveau de fluide :

Anémomètre  
 Débitmètre  
 Détecteur de niveau

## Capteurs de pression de fluides :

Capteur de pression                      Jauges diverses

## Capteurs de mesure du vide :

Jauge à déformation  
 Jauge à fil chaud  
 Jauge à ionisation

## Capteur acoustique :

Microphone électrodynamique  
 Microphone à condensateur  
 Microphone à ruban

## Capteurs de rayonnements nucléaires :

Détecteur à ionisation dans les gaz  
 Détecteur à scintillation  
 Détecteur semi-conducteur

## Capteurs d'humidité :

Hygromètre à condensation  
 Hygromètre à sorption  
 Hygromètre à variation d'impédance  
 Hygromètre électrolytique  
 Psychromètre

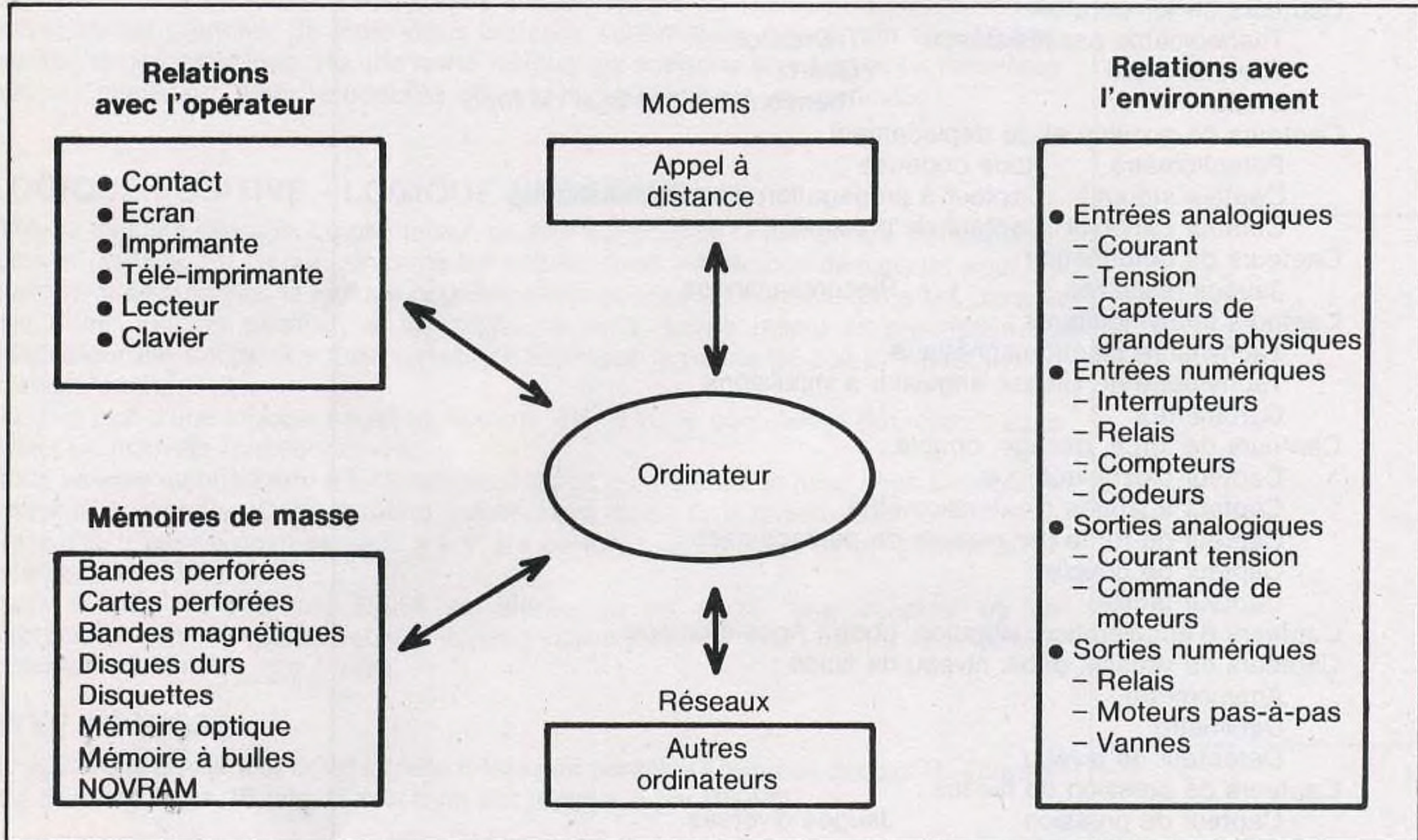
## Capteurs électrochimiques :

Capteur potentiométrique      Capteur conductimétrique  
 Capteur ampérométrique

## Capteurs de composition gazeuse :

Capteur à électrolyte solide      Catharomètre  
 Capteur électrique                  Analyseur optique  
 Capteur paramagnétique

## INFORMATIQUE INDUSTRIELLE



Un ordinateur en secteur industriel est confronté à toutes sortes de problèmes :

- des problèmes de calcul
- des problèmes de sécurité
- des problèmes d'environnement
- des problèmes de contrôle processus.

**L'environnement du calculateur est composé de :**

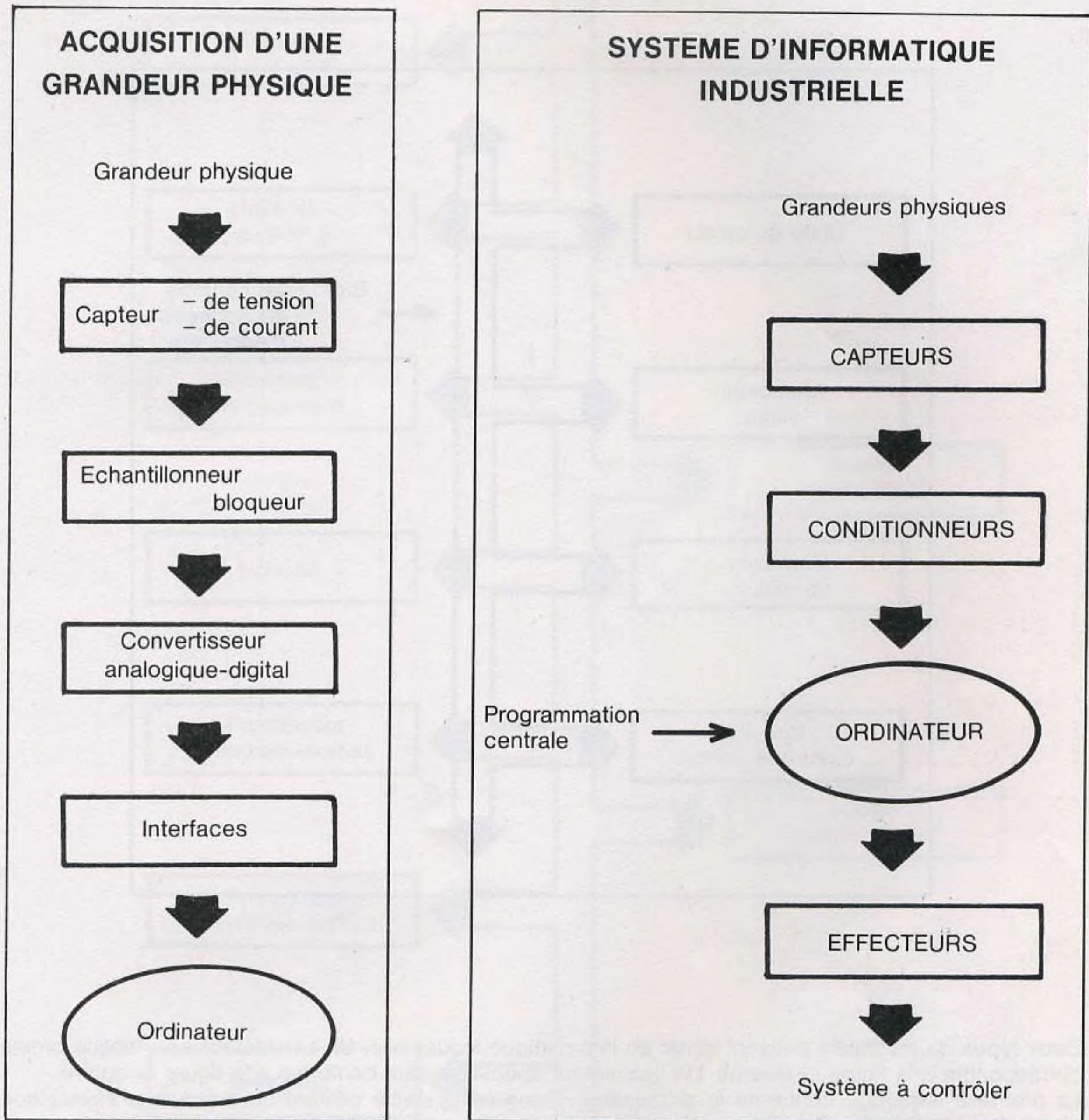
- mémoire de masse (stockage des programmes)
- cartes d'entrées-sorties (pilotage des processus)
- modems (appel à distance, télé-commande)
- réseaux (système à plusieurs calculateurs pilotés par un central)

**Relations avec l'opérateur.** Dans les endroits sans risque, l'opérateur dialogue avec le calculateur de manière conventionnelle. En atelier de fabrication, il est fortement conseillé de ne pas utiliser les terminaux habituels. Dans ce cas, il faudra faire appel à des pavés numériques de commande simple, étanche et robuste, des systèmes de visualisation simplifiés, faciles à lire, et de loin.

**Sécurité.** Dans un système informatique en milieu industriel, les problèmes de sécurité sont doubles. En plus des problèmes classiques, deux précautions sont à prendre :

- Sécurité concernant les processus industriels pilotés
- Sécurité pour le calculateur et ses périphériques concernant les parasites industriels.

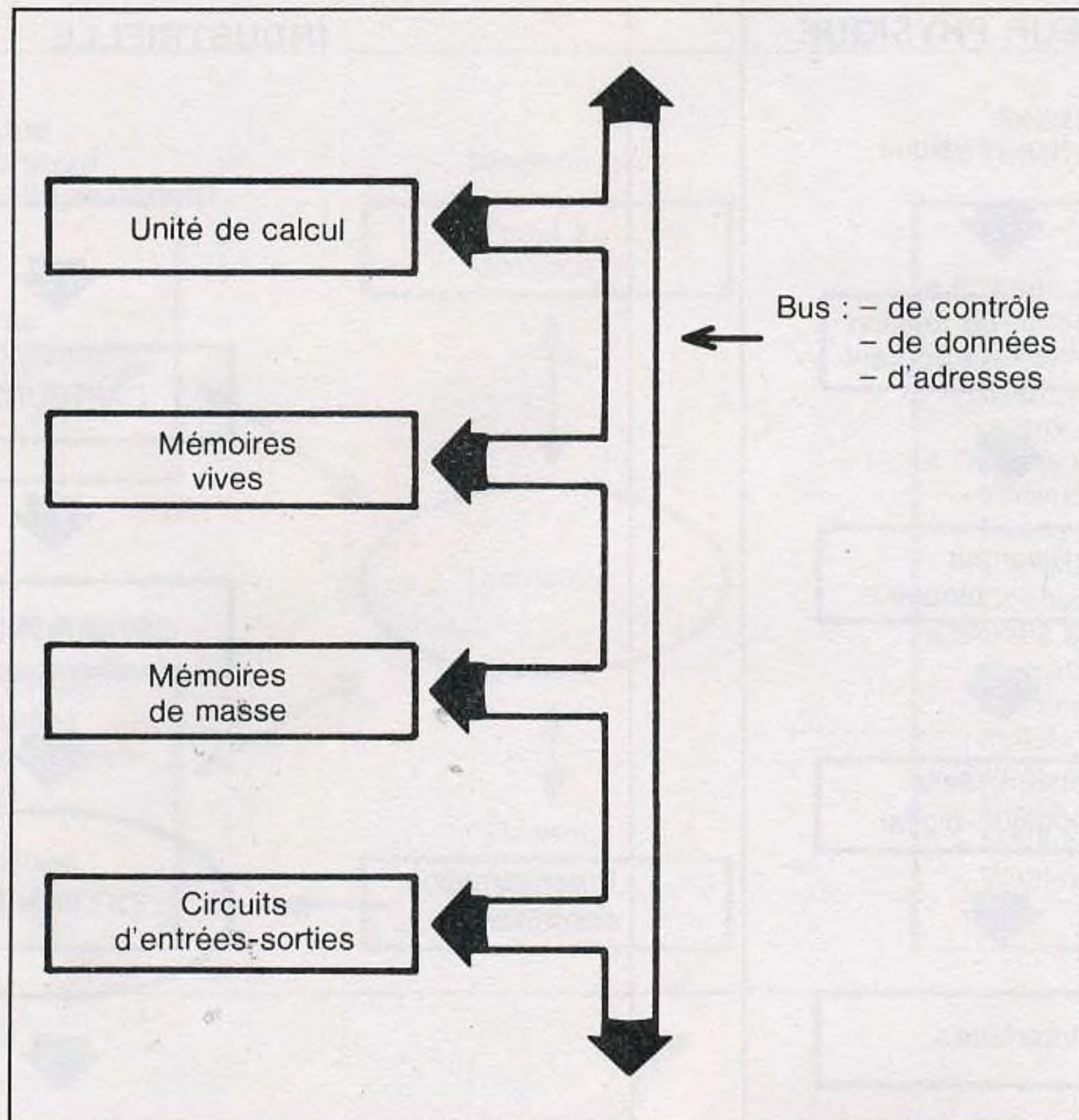
## PRINCIPE DE BASE EN INFORMATIQUE INDUSTRIELLE



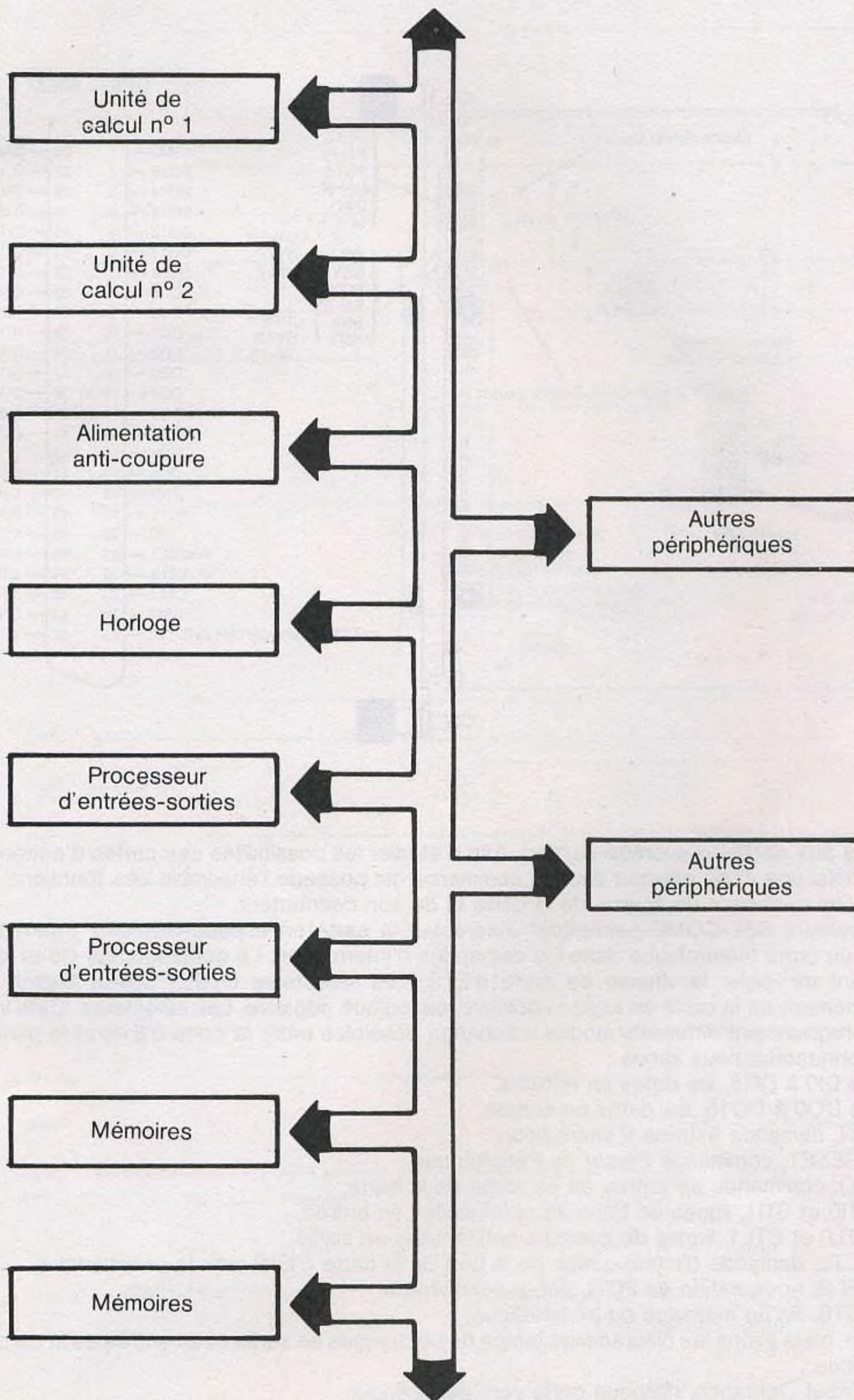
Le principe de base d'un système d'informatique industrielle consiste à analyser et à contrôler un processus et le cas échéant en modifier le déroulement. Il convient donc, à l'aide de capteurs, de mesurer les différentes phases du déroulement du processus à piloter. Un programme analyse les données et demande à des effecteurs d'optimiser les réglages.

Il existe deux types de variables accessibles par les cartes d'entrées-sorties d'un ordinateur : des variables logiques ou numériques directement exploitables ; des variables analogiques qu'il faut numériser afin qu'elles soient accessibles par le calculateur. Cette opération est réalisée par un convertisseur analogique-digital. Entre le convertisseur et la machine est placée une interface synchronisant les deux ensembles et servant de tampon. Entre le capteur et le convertisseur, un échantillonneur-bloqueur permet de figer le signal analogique pendant le temps de conversion du convertisseur.

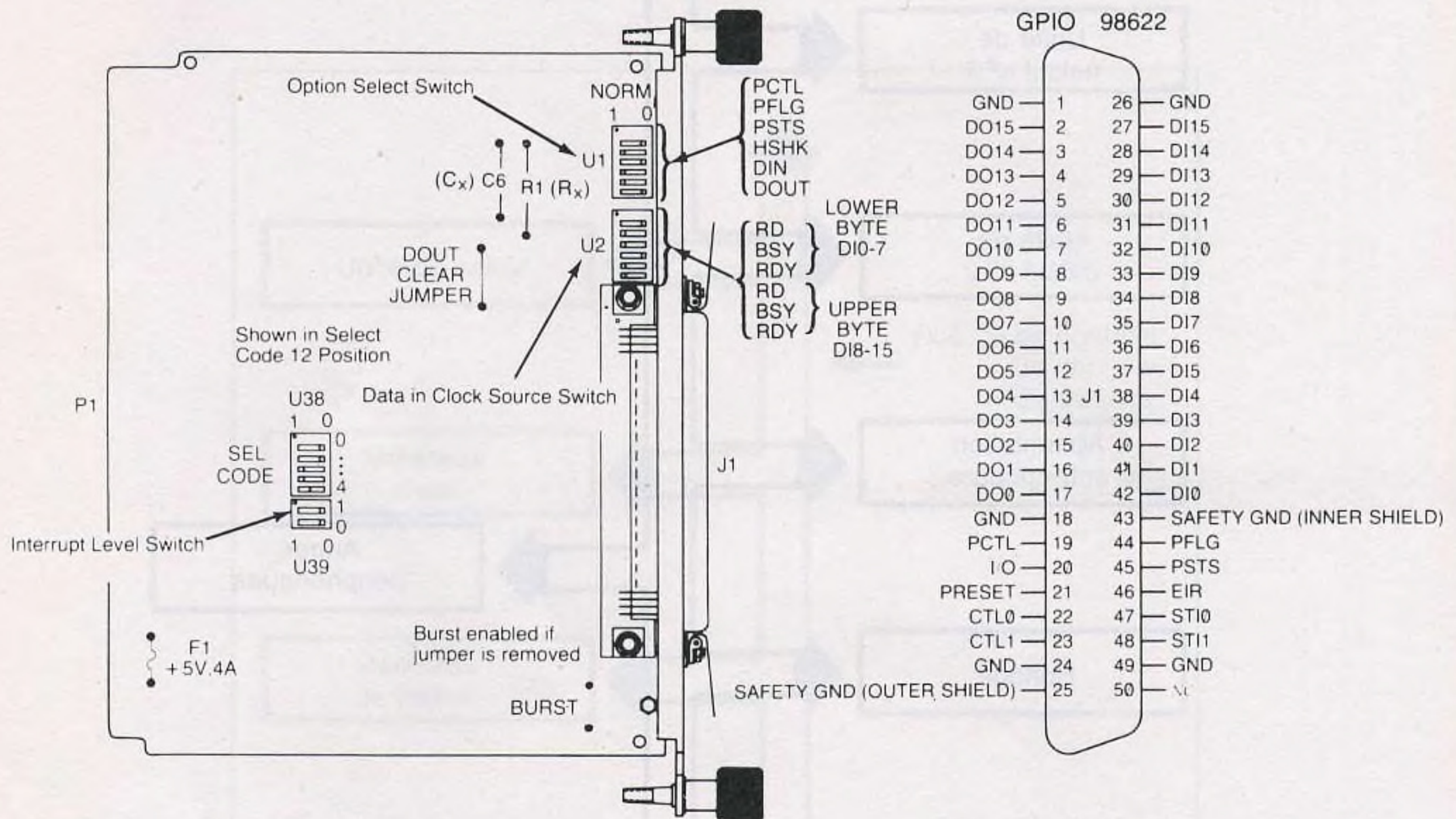
## CALCULATEUR EN INFORMATIQUE INDUSTRIELLE



Deux types de machines peuvent servir en informatique industrielle. Un calculateur scientifique ordinaire correspondra à la figure ci-dessus. Un calculateur spécialisé sera conforme à la figure ci-contre. La première remarque concerne le processeur. Dans un système pilotant un processus stratégique, il convient de doubler l'unité centrale. En effet, dans certains cas (pont roulant, presse, etc.), une panne ne doit, en aucun cas, stopper le déroulement des tâches. De ce fait, son circuit est doublé. Un bon système doit posséder une horloge indépendante de l'unité centrale. Cette horloge doit servir de base pour le chien de garde. On appelle «chien de garde» en informatique industrielle, un processus qui teste le bon déroulement du programme. Pour des raisons de sécurité, il est absolument nécessaire que le programme de base tourne dans un laps de temps maximum et ne se bloque pas sur une étape. Pour ce faire, à chaque passage d'un temps pré-déterminé, on demande à la machine de lire l'heure. Si l'opération ne se fait pas, cela signifie qu'il y a une panne. La non-lecture est alors assimilée à une interruption grave et il se produit une alerte immédiate. Dans un système évolué, certaines mémoires de masse peuvent être implantées en mémoire à bulles. Dans ce cas, il n'y a pas possibilité de risque de détérioration de disque par de la poussière. Enfin, il sera nécessaire de doter l'ensemble d'une alimentation-secteur anti-coupure avec batteries internes.



## UNE CARTE D'ENTREES-SORTIES



**Initiation aux cartes d'entrées-sorties.** Afin d'étudier les possibilités des cartes d'entrées-sorties, nous avons choisi une carte existant dans le commerce qui possède l'ensemble des fonctions souhaitables. Nous avons ci-dessus un aperçu de la carte et de son connecteur.

Les sélecteurs SEL-COME permettent d'adresser la carte en espace-mémoire. Interrupt Level Switch autorise un ordre hiérarchique dans les demandes d'interruption. Le condensateur C6 et la résistance R1 permettent de régler la vitesse de carte d'E/S. Les sélecteurs Option Select Switch déterminent le fonctionnement de la carte en logique positive, ou logique négative. Les sélecteurs Data in Clock Source Switch programment différents modes d'échange possibles entre la carte d'E/S et le périphérique.

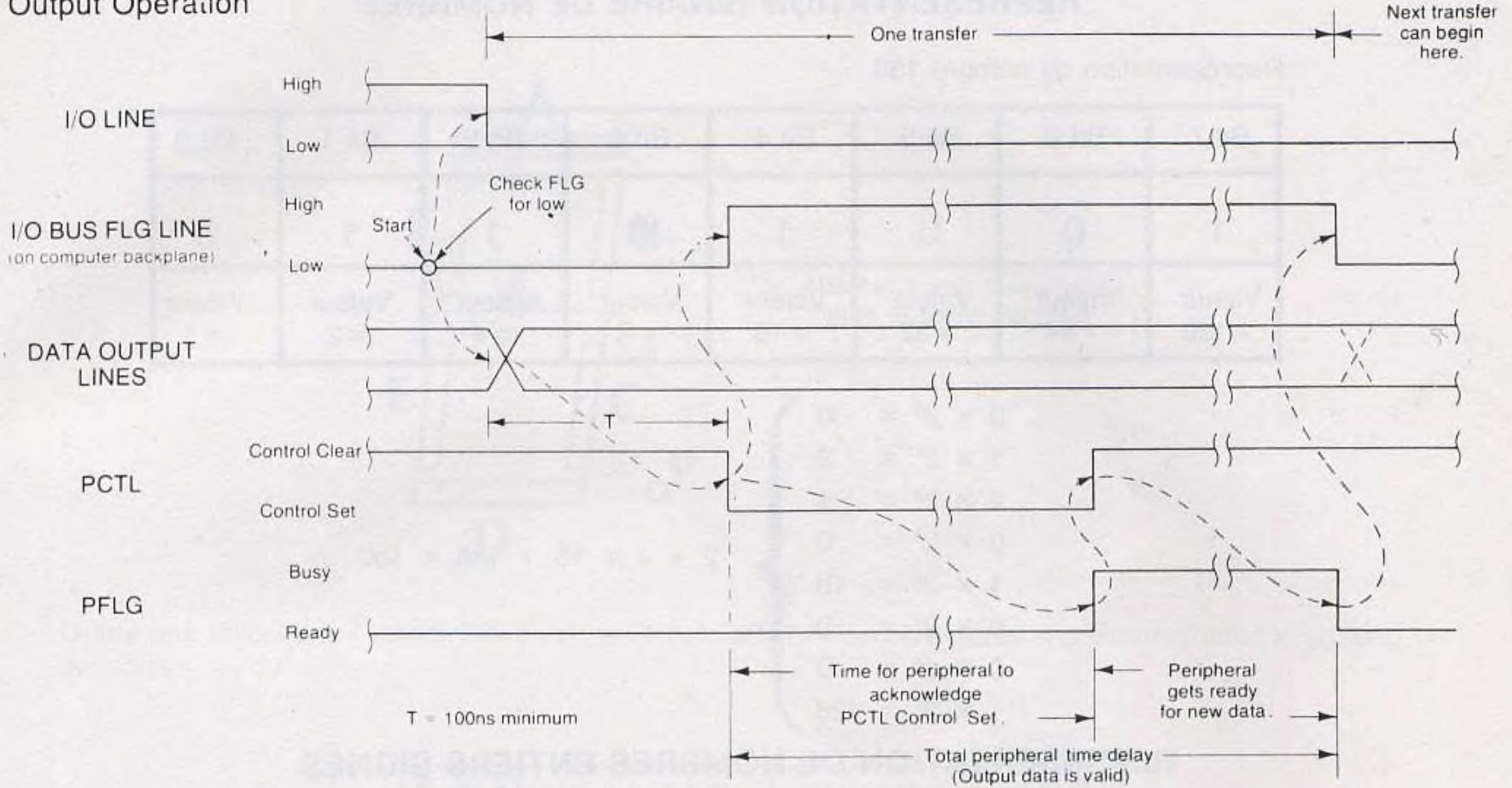
Sur le connecteur, nous avons :

- de DI0 à DI15, les datas en entrées,
- de DO0 à DO15, les datas en sorties,
- EIR, demande externe d'interruption,
- PRESET, commande Preset du Périphérique,
- I/O, commande en entrée ou en sortie de la carte,
- STI0 et STI1, lignes de contrôle optionnelles en entrée,
- CTL0 et CTL1, lignes de contrôle optionnelles en sortie,
- PCTL, demande d'interrogation de la part de la carte d'E/S vers le périphérique,
- PFLG, acceptation du PCTL par le périphérique,
- PSTS, fin de message du périphérique.

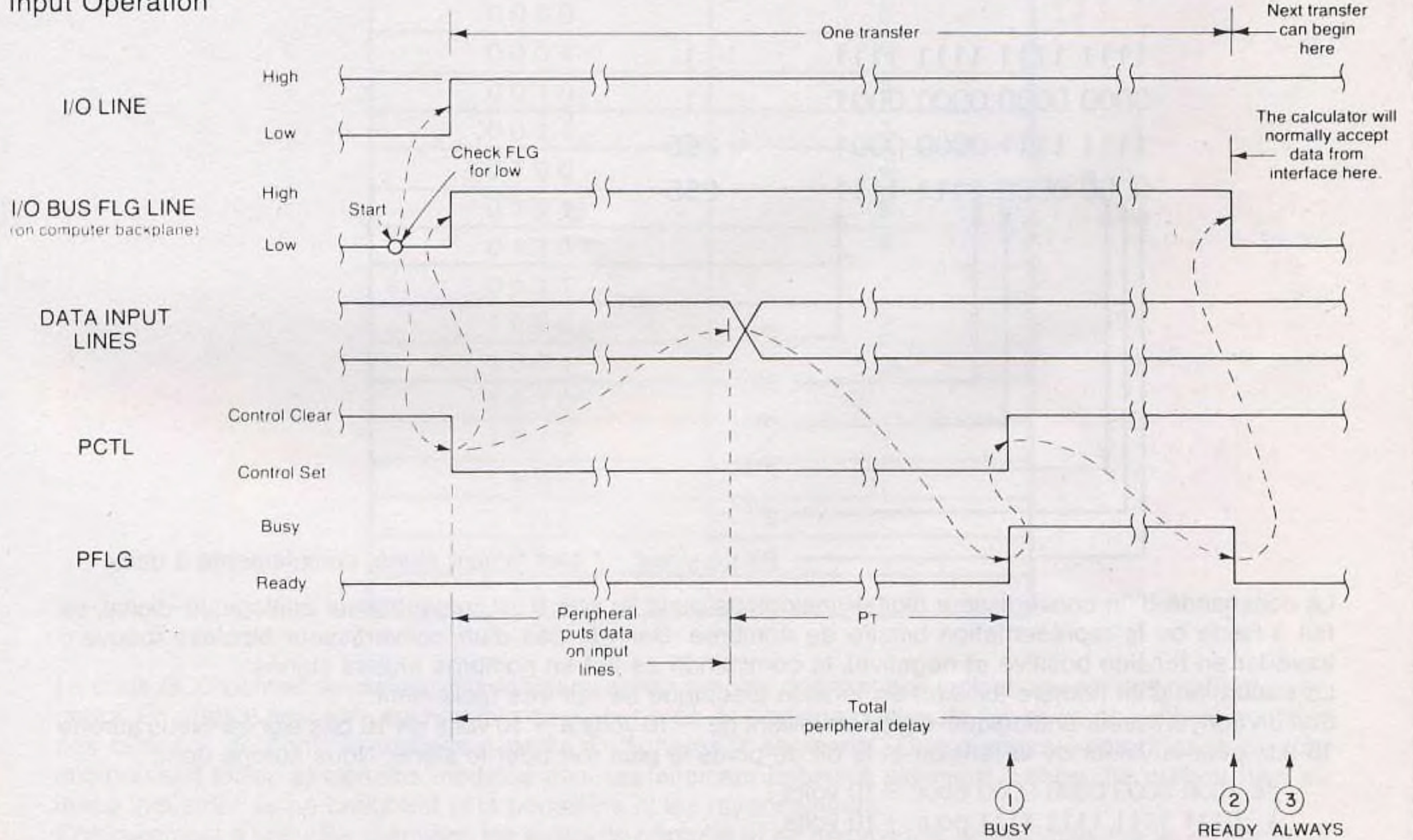
Ci-contre, nous avons les diagrammes temps des échanges en sortie et en entrée de la carte d'E/S avec le périphérique :

- Output Operation, dialogue carte vers périphérique,
- Input Operation, dialogue périphérique vers carte.

### Output Operation



### Input Operation



## CONVERTISSEUR ANALOGIQUE-DIGITAL CONVERTISSEUR DIGITAL-ANALOGIQUE : CONVERSION

### REPRESENTATION BINAIRE DE NOMBRES

Représentation du nombre 150.

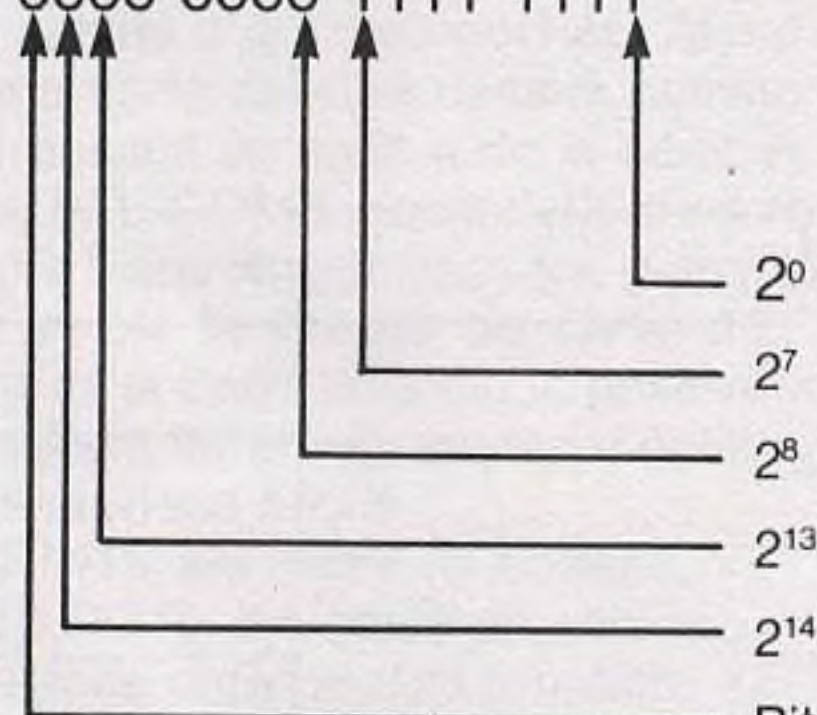
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	1	0	1	1	0
Valeur = 128	Valeur = 64	Valeur = 32	Valeur = 16	Valeur = 8	Valeur = 4	Valeur = 2	Valeur = 1

$$\begin{array}{l}
 0 \times 2^0 = 0 \\
 1 \times 2^1 = 2 \\
 1 \times 2^2 = 4 \\
 0 \times 2^3 = 0 \\
 1 \times 2^4 = 16 \\
 0 \times 2^5 = 0 \\
 0 \times 2^6 = 0 \\
 1 \times 2^7 = 128
 \end{array}
 \left. \vphantom{\begin{array}{l} 0 \times 2^0 \\ 1 \times 2^1 \\ 1 \times 2^2 \\ 0 \times 2^3 \\ 1 \times 2^4 \\ 0 \times 2^5 \\ 0 \times 2^6 \\ 1 \times 2^7 \end{array}} \right\} 2 + 4 + 16 + 128 = 150$$

### REPRESENTATION DE NOMBRES ENTIERS SIGNES

Représentation binaire \* Nombre

1111 1111 1111 1111	- 1
0000 0000 0000 0001	1
1111 1111 0000 0001	- 255
0000 0000 1111 1111	255



Bit de signe \* Mot binaire signé, complémenté à deux.

La commande d'un convertisseur digital-analogique ou la lecture d'un convertisseur analogique-digital, se fait à l'aide de la représentation binaire de nombres. Dans le cas d'un convertisseur bipolaire (pouvant travailler en tension positive et négative), la commande se fait en nombres entiers signés.

La traduction d'un nombre (binaire) en tension électrique se fait très facilement.

Soit un convertisseur analogique-digital travaillant de - 10 volts à + 10 volts en 16 bits signés. Nous aurons 15 bits pour la valeur de la tension et le bit de poids le plus fort pour le signe. Nous aurons donc :

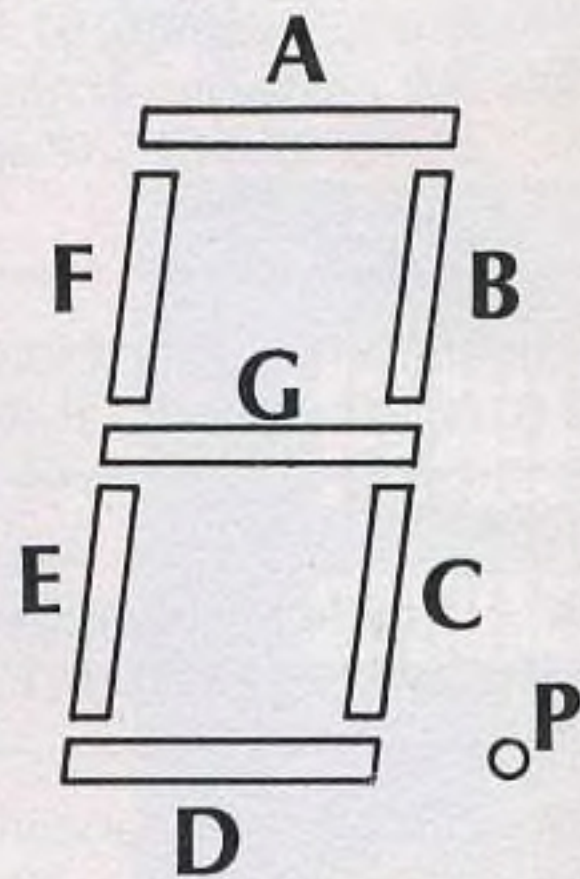
de 1000 0000 0000 0000 pour - 10 volts  
à 0111 1111 1111 1111 pour + 10 volts

Dans le cas d'un convertisseur 8 bits non signé fonctionnant de 0 à 5 volts, nous aurons :

soit 0 volt = 0000 0000 en binaire  
soit 5,88 volts pour 1001 0110 (soit 150).



## CODE BCD, AFFICHEURS 7 SEGMENTS



Affichage sur 7 segments plus le point décimal.

Grâce aux afficheurs 7 segments, il est possible d'afficher les caractères représentés dans le tableau ci-dessous.

Code binaire sur 4 bits	Caractère représenté
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	line feed
1 0 1 1	+
1 1 0 0	,
1 1 0 1	-
1 1 1 0	E
1 1 1 1	.

Le code BCD permet de réaliser des tableaux assez simples donnant des indications sur des processus en cours. En effet, il peut être souhaitable de placer près de certains postes de travail ou en salle de contrôle des tableaux de contrôle réalisés à partir d'afficheurs 7 segments. Ces derniers existent dans de très nombreuses tailles et certains modèles sont parfaitement lisibles à plusieurs mètres. Se prêtant bien au milieu industriel, ils ne craignent ni la poussière ni les rayonnements.

Contrairement à une idée première, les salles de contrôle et de commande, en industrie lourde, ne sont pas équipées de terminaux classiques en informatique industrielle, mais de larges tableaux de commande et de visu, visibles de loin et actionnables même si l'opérateur est équipé de gros gants.



# TOUT SUR LES PÉRIPHÉRIQUES

**NOUVEAU**

dans la  
**COLLECTION**  
**«ETUDES»**  
aux  
éditions  
fréquences



- 85 schémas
  - 20 tableaux
  - 136 pages
- Prix : 150 F

Les périphériques font partie intégrante d'un système informatique. En parallèle de l'unité centrale, qui gère et synchronise l'ensemble, ils sont responsables de différentes fonctions comme :

- la mémoire de masse : unités de disques souples et de disques durs, lecteur de cassettes ;
- le dialogue avec l'utilisateur : clavier, écran vidéo, imprimante ;
- les télécommunications : modem.

Tous ces périphériques sont décrits dans cet ouvrage avec, pour chacun d'eux, une partie technologie (principe de fonctionnement, caractéristiques techniques) et une partie interface (coupleurs d'entrées-sorties, connecteurs de liaison).

Dans chaque grande catégorie (mémoire, imprimante), une analyse comparative des différents produits existants est effectuée.

*Philippe Faugeras, docteur ingénieur en électronique, est responsable matériel dans une entreprise d'informatique traitant des réseaux de P.C. Au préalable, il a acquis son expérience en travaillant sur des sujets comme les automatismes et les télécommunications dans deux grandes sociétés françaises (Bull, CGE). Philippe Faugeras est l'auteur d'un premier ouvrage «L'électronique des micro-ordinateurs» paru aux Editions Fréquences.*

En vente chez votre libraire et aux Editions Fréquences.

## **BON DE COMMANDE**

Je désire recevoir l'ouvrage «Périphériques interfaces et technologie» au prix de 160 F (150 F + 10 F de port).

Nom .....

Adresse .....

A adresser aux EDITIONS FRÉQUENCES 1 boulevard Ney, 75018 Paris

Règlement ci-joint :

Par chèque bancaire

par chèque postal

par mandat

# Le Victor PC ne coûte que 24.900 F n'en déplaie à [REDACTED].

Le Victor PC 15 ne coûte que 24.900 F\*.

Certains d'entre vous penseront peut-être – et nous en connaissons qui aimeraient bien que ce soit vrai – qu'à 24.900 F\*, il ne peut s'agir que d'un PC "bradé". Une telle réaction est d'ailleurs compréhensible quand on songe aux prix pratiqués sur le marché, en matière de PC. Prenons par exemple [REDACTED]. Son PC coûte 50% plus cher que le Victor PC 15.

Et pourtant, les performances du Victor PC 15 sont équivalentes, voire supérieures, à celles de l'[REDACTED] PC. La preuve, la voici :

Alors que la plupart des micro-ordinateurs propose une capacité de stockage de 10 Mo, le Victor PC 15, lui, offre une capacité de 15 Mo! De plus, l'utilisateur du Victor PC 15 bénéficie, grâce à un moniteur de 14 pouces, de 30% de surface écran supplémentaires (la quasi-totalité du matériel concurrent étant équipée d'un moniteur 12 pouces).

Et ce n'est pas tout! Le Victor VU – l'interface utilisateur – permet un gain de temps appréciable en guidant dans son travail l'utilisateur, par de simples messages organisés comme des menus. Finie, désormais, la consultation fastidieuse et peu pratique du manuel du système d'exploitation!

Et l'on pourrait parler des 5 emplacements d'extensions disponibles pour accroître les possibilités du PC...

Non décidément, [REDACTED] devra se faire une raison et s'accommoder de la présence sur le marché du Victor PC 15! Un PC compatible avec les standards du marché, aussi performant que celui que fabrique [REDACTED] et à un prix bien plus séduisant que celui affiché par [REDACTED].

Car au risque de le répéter et de déplaie à [REDACTED], ces 50% sont difficilement justifiables. D'ailleurs les vendeurs d'[REDACTED] doivent déjà en savoir quelque chose...

Lesquels vendeurs d'[REDACTED] ne vont sans doute guère apprécier que nous vous donnions nos coordonnées - et que vous puissiez nous contacter à Victor Technologies - Tour Horizon, 52, quai de Dion-Bouton, 92800 Puteaux (tél. : 778.14.50) ; ou encore à Lyon : (7) 234.12.45 ; Montpellier : (67) 64.71.72 ; Nantes : (40) 89.24.28. Mais l'on ne peut contenter tout le monde et [REDACTED]!



\* Configuration complète avec clavier et écran monochrome. Prix H.T. au 1/9/85. (Possibilité de location financière : 700 F par mois sur 48 mois - CEGEDATA.).

## VICTOR

Comme [REDACTED] moins cher qu'[REDACTED]