Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

U. S. T. H. B.

École Nationale Polytechnique

DÉPATEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLOME D'INGÉNIEUR D'ÉTAT

TRAITEMENT DE FICHIERS SEQUENTIELS ET ETUDE D'UNE INTERFACE CASSETTE POUR L'APPLE II PLUS

PROPOSÉ PAR :

HADDADI

ÉTUDIE PAR :

A. BOUHADJERA

Y. BOUDJEDIR

PROMOTION - JANVIER 1983

-o- DEDICACES -o

- A mes frères et soeurs, plus particuliérement NOUI et LYES
- A tous mes amis

ABDELMALEK

- A mon pére
- A ma mére
- A mes fréres et soeurs
- A toute ma famille
- A tous mes amis

YOUCEF

-O- REMERCIEMENTS -O-

Nous remerçions Monsieur M.HADDADI, enseignant à L'E.N.P.A, pour nous avoir proposè ce sujet.

Nous tenons à remercier aussi MM. H.FARAH et B.BOUDRAA, enseignants à L'E.N.P.A, pour nous avoir aidè dans notre travail ainsi que Mademoiselle ZIZI.

Que tous ceux qui ont contribué à la realisation de cette étude trouvent ici l'expression de notre sincère gratitude.

-o TABLE DES MATIERES -o

- TROISIEME PARTIE -

ETUDE DE L'INTERFACE CASSETTE

- INTRODUCTION -

Dans les systèmes photovoltaïques, la connaissance des données météorologiques est fondamentale. C'est pourquoi, depuis I980, certaines de ces données sont continuellement enregistrées sur cassette magnétique au laboratoire des cristaux et couches minces du C.E.N. (Centre des énergies nouvelles), mais leur traitement necessite l'utilisation d'un ordinateur.

Le laboratoire comptant acquérir un micro-ordinateur du type APPLE II; nous nous sommes donc proposés d'étudier la possibilité d'effectuer le traitement de ces données par ce micro-ordinateur, ce qui aura pour effet de poser certaines contraintes au niveau de la compréhension des données enregistrées. Dans ce cas, l'unique solution est de réaliser une interface entre la cassette et le micro-ordinateur.

Notre travail se divise en trois parties essentielles :

- La première est consacrée à la présentation du travail et du micro-ordinateur utilisé.
- La deuxième expose le programme des traitements des données.
- La troisième contient les résultats trouvés et le schéma synoptique proposé.


REMIERE
ARTIE

I - PRESENTATION DU TRAVAIL

I.1 : Position du problème

Les mesures qui sont effectuées au laboratoire des cristaux et couches minces du C.E.N. concernent la température ambiante, la température des cellules, l'ensoleillement global l'ensoleillement direct et la puissance délivrée par le générateur.

Ces mesures sont enregsitrées sur cassette standard à l'aide d'un enregistreur ALEP type ED.7620 qui permet aussi d'obtenir ces données enregistrées grace à une imprimante, ces dernières sont codées suivant un code propre à l'enregistreur.

Elles sont effectuées automatiquement par intervalles de I2 minutes, néanmoins, on peut faire des mesures manuelles.

Notre travail consiste à :

- Faire un programme informatique qui nous permet de traiter ces données.
- Déterminer les codes d'enregistrement correspandant à l'APPLE II et à l'enregistreur.
- Proposer un schéma synoptique pour l'interface cassette.

Le traitement informatique se fera comme suit :

* Dans une première étape , on recopiera les données en mémoire RAM avec des DATA à l'aide du clavier. Ces données représentent un fichier séquentiel en mémoire.

* Dans une deuxième étape, on devra élaborer un programme de traitement de ces données.

* Enfin, dans une troisième étape, on enregistrera ces données sur cassette à l'aide de l'APPLE II et on fera un programme qui pourra s'adresser directement à ces données et les traiter.

Cet enregistrement nous servira par la suite à faire une comparaison entre les deux codes utilisés par l'APPLE II et par l'enregistreur.

Ce traitement informatique se fait par un même programme qui sonsiste à :

- eliminer les mesures manuelles.

- Décoder ces données et faire la moyenne sur chaque heure.
- Faire un tableau des mesures sur chaque heure.
- Faire un tableau des moyennes sur 24 heures.
- Faire un graphe pour chaque mesure sur 24 heures.
- Faire la moyenne sur 24 heures pour chaque mesure.

La partie électronique se fera en deux étapes :

*Iére Etape : détermination des formes d'enregistrement et des fréquences de l'APPLE II et de l'enregistreur ALEP.

*2émé Etape : Faire la comparaison entre les deux codes et proposer un schéma synoptique de l'interface cassette.

I.2 : Notions générales sur les fichiers

L'une des propriétés des cassettes mágnétiques et de certains disques magnétiques est leur "amovibilité" : aprés avoir écrit sur ces mémoires, l'utilisateur peut les démonter de l'unité lecture-ecriture et les ranger afin de disposer ultérieurement des informations qui y ont été stockées.

Les caractéristiques physiques de ces mémoires entrainent pour l'utilisateur des contraintes qui apparaissent dans une certaine mesure au niveau des instructions d'entrée-sortie.

Ces caractéristiques générales sont les suivantes :

- Echanges d'informations par bloc et non par caractères.
- Accés "séquentiel" seulement pour les cassettes magnétiques
- Possibilités d'accés "séléctif" pour les disques magnétiques.

Il s'en suit que les fichiers peuvent avoir des structures différentes suivant le type de mémoire utilisé. Il existe plusieurs façons de ranger un fichier. Les compilateurs BASIC les plus complets reconnaissent deux méthodes d'organisation :

* L'organisation séquentielle.
* L'organisation directe.

I.2.1 : L'organisation séquentielle :

A la création du fichier, les articles sont rangés les uns à la suite des autres sur le support. Les traitements ultérieurs du fichier doivent être prévus de façon à accepter les articles dans l'ordre où ils se présentent. Ils ont l'avantage de pouvoir être "listés" comme un programme.

I.2.2: L'organisation directe :

Les fichiers à accés direct sont d'un emploi plus souple que ceux à accés séquentiel car avec de tels fichiers, il est possible d'avoir accés directement à n'importe quel enregistrement, sans avoir à lire les précédents.

I.3 : Méthode de travail :

Pour faire cette étude, on a trouvé indispensable de connaître au moins un langage de programmation pour pouvoir enregistrer les mêmes données sur une autre cassette. A cette fin, il s'est avéré que le langage BASIC APPLESOFT, qui est à la fois simple et puissant, s'adapte parfaitement.

Nous commencerons par la recherche d'un programme de traitement des données enregistrées avant de nous tourner vers leur exploitation directe, ceci pourra se faire en créant un fichier séquentiel en mémoire RAM.

I.3.1 : Elaboration du programme :

Puisque le traitement des données ne comporte pas de nombreux calculs, nous avons préféré utilisé un seul programme.

I.3.2 : Enregistrement sur cassette :

Aprés avoir concu le programme précédent, on l'enregistre sur cassette pour le sauvegarder, puis on passe à l'étape qui consiste à séparer les données du programme. Cela sera possible, si on réussit à écrire et à lire des données enregistrées sur cassette à l'aide d'un programme approprié. Le traitement sera le même car nous serons en présence d'un même fichier séquentiel.

I.3.3 : Etude de l'interface cassette :

Cette étude, qui représente la partie électronique de notre travail, ne peut évidemment être menée que sur la base des resultats précédents.

II - PRESENTATION DE L'APPLE II PLUS :

L'APPLE II est concu autour du microprocesseur 6502 qui lui donne, entre autres possibilités, le moyen d'adresser directement jusqu'à 65536 (64K) mots de mémoire de 8 bits dont une partie (4K octets) sera résérvé aux fonctions d'entrée et de sortie de données.

Dans les 60 K octets restants, le système APPLE II peut disposer de mémoire ROM (Read only mémory ou mémoires mortes) et de mémoire RAM (Random accéss mémory ou mémoires vives). Pour communiquer avec l'environnement par l'intermédiaire d'un clavier, d'un écran vidéo, d'un lecteur-enregistreur de cassettes par exemple, le système contient des interfaces spécifiques intégrées à la carte-mère, mais il permet aussi la connection de 8 cartes d'interfaces supplémentaires qui augmentent les possibilités de communication et de mémorisation.

Le logiciel de base du système APPLE II est implanté dans les mémoires mortes (ROM)(I2K) permettant,dès la mise sous tension, un dialogue soit avec le moniteur soit avec un interpreteur BASIC.

II.1 : LE Matériel :

Nous décrirons les caractéristiques principales du microprocesseur, les deux types de mémoires utilisées et les particularités du système entrée/Sortie.

II.1.1 : Aspect de la maquette :

La figure ci-aprés montre la disposition des principaux éléments de la maquette.

II.I.2 : Le microprocesseur 6502 :

Ce circuit intégré de 28 broches de type 6502 est fabriqué par M.O.S Technology Inc. Synertek ou Rockwell.

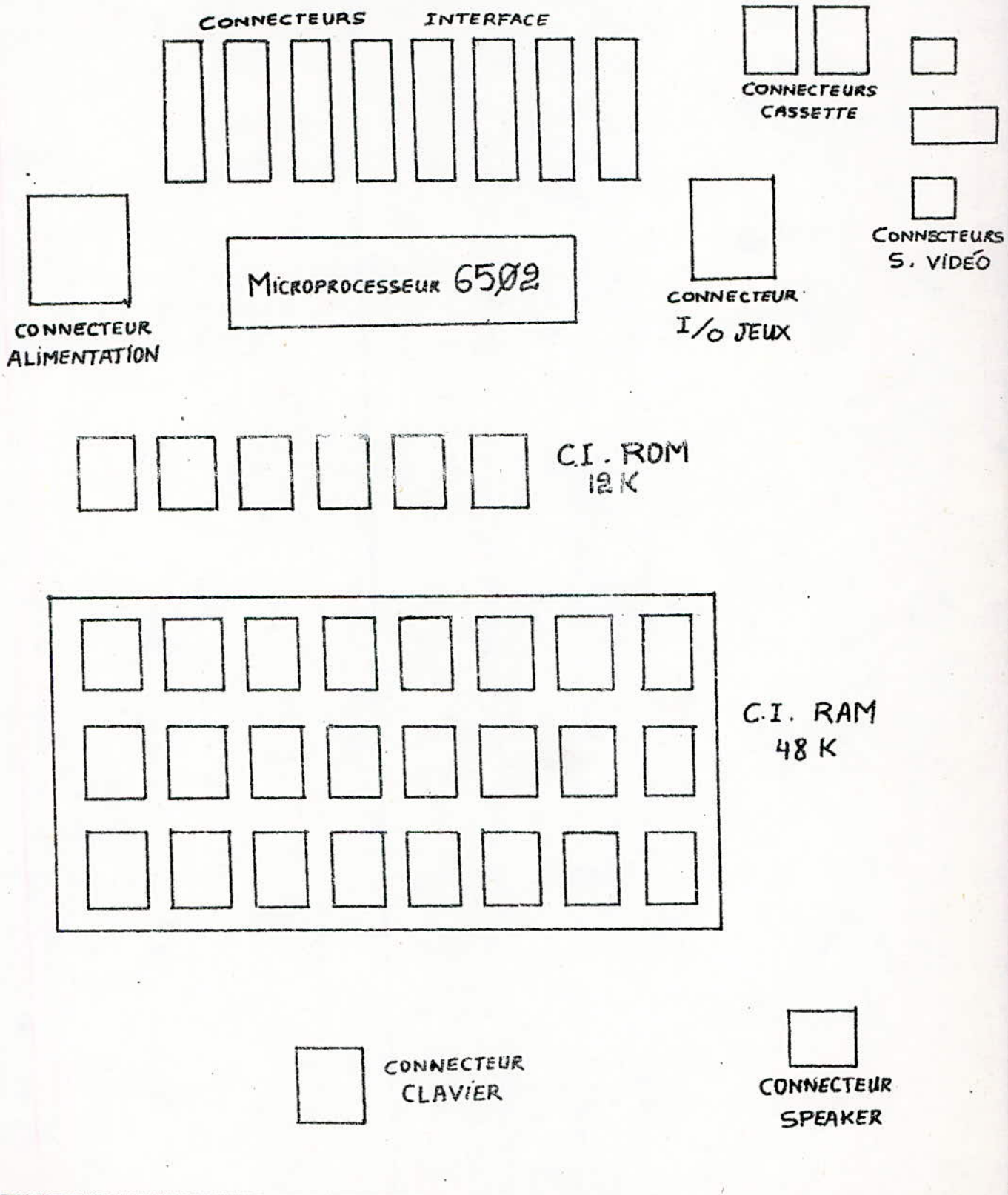
L'unité de traitement gère des régistres de 8 bits en parallèle, et est capable de recevoir ou de canaliser des informations sur 8 bits provenant de ou allant à 65536 position de mémoires distinctes (les adresses sont données sur I6 bits).

Un régistre accumulateur, deux régistres d'indexe, un régistre pointeur de pile (256 octets de profondeur maximum), un régistre d'état du processeur et le régistre compteur ordinal (I6 bits) associés à 56 instructions et I3 modes d'adressage, composent les moyens de programmation du microprocesseur.

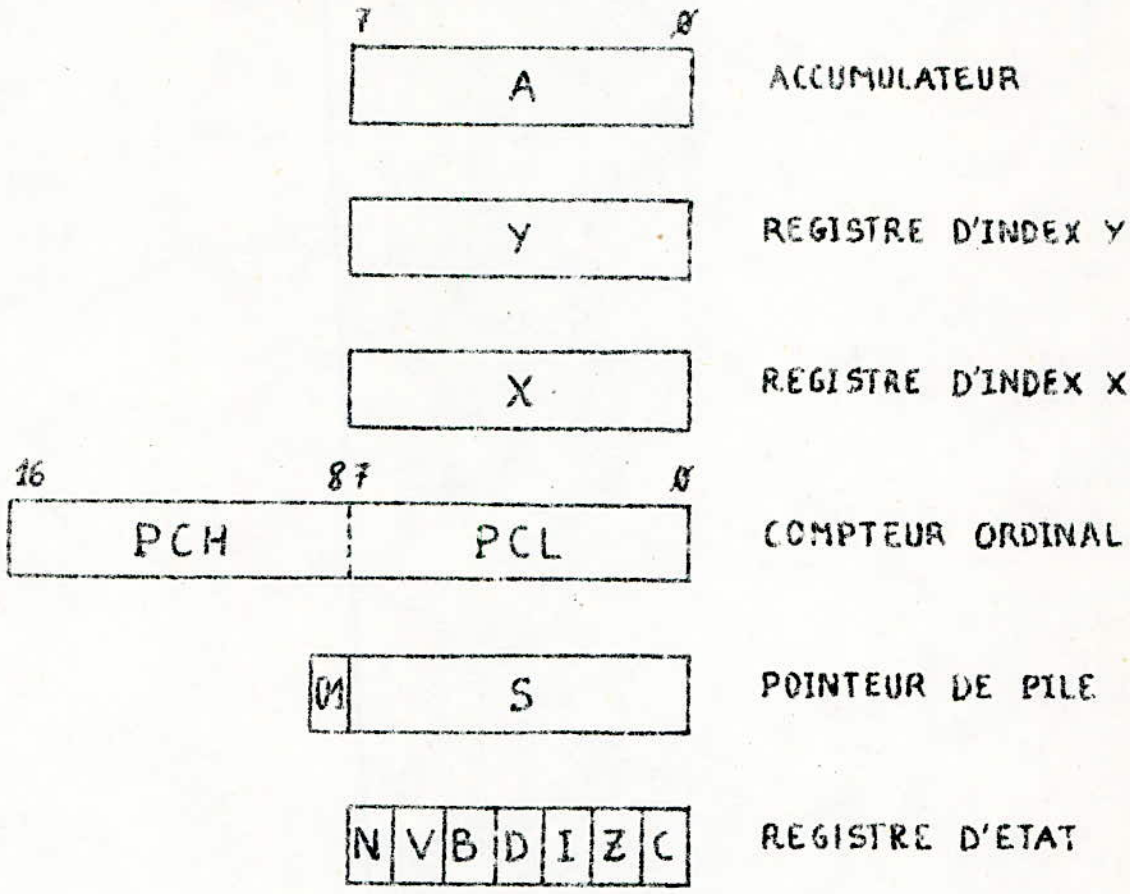
La cadence des opérations est réglée à I.023.000 cycles par seconde, soit I,023 MHz.

Les régistres programmables du 6502 sont représentés en page 8.

ASPECT DE LA MAQUETTE



REGISTRES PROGRAMMABLES DU 6502:



* INDICATEURS :

N: signe
V: Dépassement
B. Arrêt
D: Decimal
I: Interruption
Z: Zéro
C: Retenue

* LES BUS :

- Bus d'Adresse 16 bits (Parallèles)
- Bus de données 8 bits (Parallèles bidirectionnelles)

II.I.3 : Les mémoires :

Le microprocesseur 6502 de l'APPLE peut choisir directement un total de 65536 adresses. La mémoire contient un livre de 256 pages chacune d'elle contenant 256 mémoires.

Les 256 pages se divisent en trois catégories : les RAM, les ROM et les mémoires réservées aux E/S. cette répartition est illustrée par la figure ci-dessous :

REPRESENTATION DE LA MEMOIRE		
NUMERO DE LA PAGE		
DEC.	HEXA.	
Ø . . . I9Ø I9I	Ø Ø Ø . . . Ø B E Ø B F	RAM (48 K)
I92 . . I99	Ø C Ø . . Ø C 7	I/O (2K)
2ØØ . . 2Ø7	Ø C 8 . . Ø C F	I/O ROM (2K)
2Ø8 . . 255	Ø D Ø . . Ø F F	ROM (I2 K)

a) Les mémoires vives (RAM) :

Elles sont de technologie N MOS et de type dynamique (faible consommation, grande intégration).

Les circuits intégrés composant la mémoire RAM sont disposés à raison de trois lignes de 8 circuits de mémoire chacune.

Le temps d'accés à une position quelconque de mémoire RAM est 250 nano-secondes, et le temps de cycle de 375 nano-seconde.

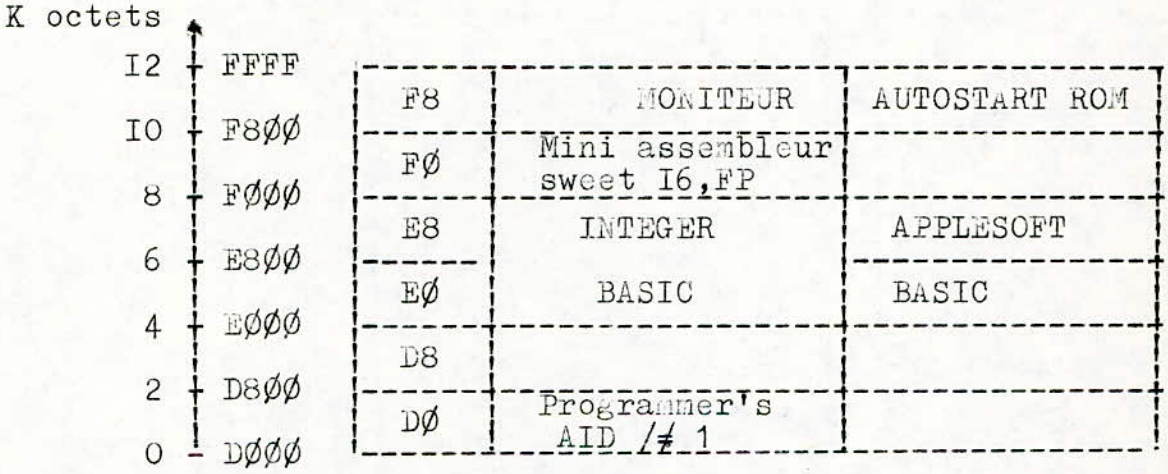
La plus grande partie de la RAM est disponible pour stocker des programmes et des données, toutefois l'APPLE réserve certaines adresses pour le moniteur, certains langages et d'autres fonctions du système. Le schéma de l'utilisation de la RAM est donné en page suivante.

b) Les mémoires mortes (ROM) :

Les six emplacements situés entre le microprocesseur et les mémoires vives sont réservés à des blocs de deux Koctets de mémoire morte.

Un certain nombre de programmes sont disponibles en plusieurs boitiers ROM, comme les moniteurs, l'integer BASIC et d'autres utilitaires du système.

Ces mémoires ont les adresses les plus hautes du système et occupent I2 K octets de l'espace total. Voici comment les programmes en mémoire ROM se situent :



Si le programme interpreteur est installé en ROM (Cas de l'APPLE II plus), il bloque IOK de ROM et empèche donc de disposer d'autres programmes utilitaires en mémoires mortes.

UTILISATION DE LA RAM				
numéro de pages déc.	numéro de pages hexa.	utilisation		capacité
0	$00	programmes du micro-ordinateur		32 bytes
1	$01	pile du microprocesseur		32 bytes
2	$02	tampon d'entrée utilisé par GETLN		32 bytes
3	$03	adresses utilisées par le moniteur		32 bytes
4 . . . 7	$04 . . . $07	Texte graphisme basse résolution page 1		1K
8 . . . 11	$08 . . . $11	Texte graphisme basse résolution	RAM DISPONIBLE	1K
12 . . . 31	$12 . . . $31			5K
32 . . . 63	$32 . . . $63	Graphisme haute résolution page 1		8K
64 . . . 95	$64 . . . $95	Graphisme haute résolution page 2		8K
96 . . . 191	$96 . . . $191			24K

1K

c) Entrées/sorties (I/O) :

Dans l'espace d'adresses $C000 à $CFFF, sont disposés 4096 (4K) octets exclusivement réservés aux fonctions d'entrée et de sortie d'information.

Que les fonctions soient implantées dans le système APPLE sur la carte-mère où sur les cartes d'interface avec des des périphériques, elles communiquent avec le microprocesseur par l'intermédiaire de ces adresses réservées.

UTILISATION DES 4K RESERVES AUX E/S

Nombre d'octets	ADRESSES	UTILISATION
128	$C000 à $C07F	Entrées-sorties controlées sur la carte-mère
I6 X 8	$C080 à $C0FF	I6 Adresses pour chacun des 8 connecteurs d'E/S
256 X 7	$CS00 à $CSFF S : de 1 a 7	256 Adresses par carte d'interface pour un programme d'E/S en ROM
2048	$C800 à $CFFF	2048 Adresses pour d'autres programmes ROM sur les cartes d'interface (une seule ROM de 2K est active)

II.2. : Le Logiciel :

De par son architecture, le système APPLE ouvre de multiples portes pour y implanter des logiciels en ROM et en RAM

L'utilisateur dispose de la quasi-totalité des 48 K de la RAM de la carte-mère pour son propre logiciel, et il peut opérer sur l'APPLE a tous les niveaux (Moniteur et langages évolués).

II.2.1 : Le Programme Moniteur :

Ce logiciel réalisé en ROM et fourni avec le matériel sur la carte-mère, couvre les 2K octets les plus hauts placés en mémoire centrale de l'adresse $F800 à l'adresse $FFFF.

on accède à ce programme en s'adressant à $FF69 par l'instruction CALL - I5I (Cas de l'autostart ROM).

Il signale sa présence en ligne par le symbole * précédant le curseur de l'écran vidéo.

On sort de ce programme par l'instruction 'C T R L C' puis RETURN pour donner la main au BASIC APLLESOFT.

Le moniteur exécute à la demande les routines d'entrée sortie concernant le clavier, le vidéo, les cassettes, les manettes ; Il peut-être appelé pour examiner des mémoires, transférer des zones de mémoire d'une adresse à une autre, exécuter les programmes en langage machine en mode pas à pas. ou observer les contenus de registres etc...

II.2.2. : Le langage BASIC APPLESOFT :

Ce logiciel, réalisé en ROM, installé sur la carte-mère ou sur une autre carte Firmware, couvre IO K octets depuis $D000 jusqu'à $F7FF.

C'est un interpreteur BASIC étendu, permettant l'utilisation de variables réelles, entières ou alphanumériques, comprenant de nombreuses fonctions sur les chaînes, sur les nombres réels, sur les tableaux à plusieurs dimensions etc...

L'implantation en mémoire RAM d'un programme interprété par le BASIC APPLESOFT est divisée en trois zones :

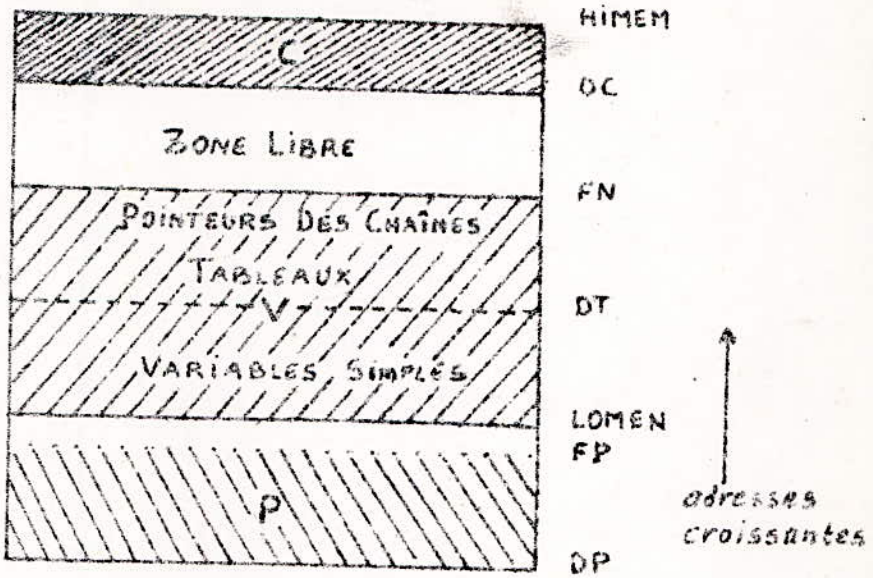
* La zone P des instructions du programme, située dans le bâs de la mémoire.

* La zone V des variables entières et réelles et des des pointeurs des chaînes alphanémuriques au dessus de la précédente.

* La zone C des chaînes de caractères, située dans le haut de la mémoire.

Dans la zone des variables ou des données némuriques, on distingue une zone de variables simples et, au dessus, la zone des tableaux de variables.

carte de la mémoire RAM
pendant l'exécution d'un programme
écrit en Basic Applesoft :



Les adresses de début et de fin de ces zones sont accessibles dans les adresses suivantes :

ADRESSES HEXADECIMALES	DECIMALES	CONTENU	Identification
$67, $68	103, 104	Adresse de début de la zone du programme en général $800	DP
$AF, $B0	175, 176	Adresse de fin de la zone du programme	FP
$69, $6A	105, 106	Adresse de début de la zone des données numériques. En général égale à FP.	LOMEN
$6B, $6C	107, 108	Adresse de début de la zone des tableaux numériques.	DT
$6D, $6F	109, 110	Adresse de fin de la zone des variables numériques.	FN
$6F, $70	111, 112	Adresse de début de la zone des chaînes de caractères.	OC
$73, $74	115, 116	Adresse de fin de la zone des chaînes de caractères	HIMEM

-o DEUXIEME PARTIE -o

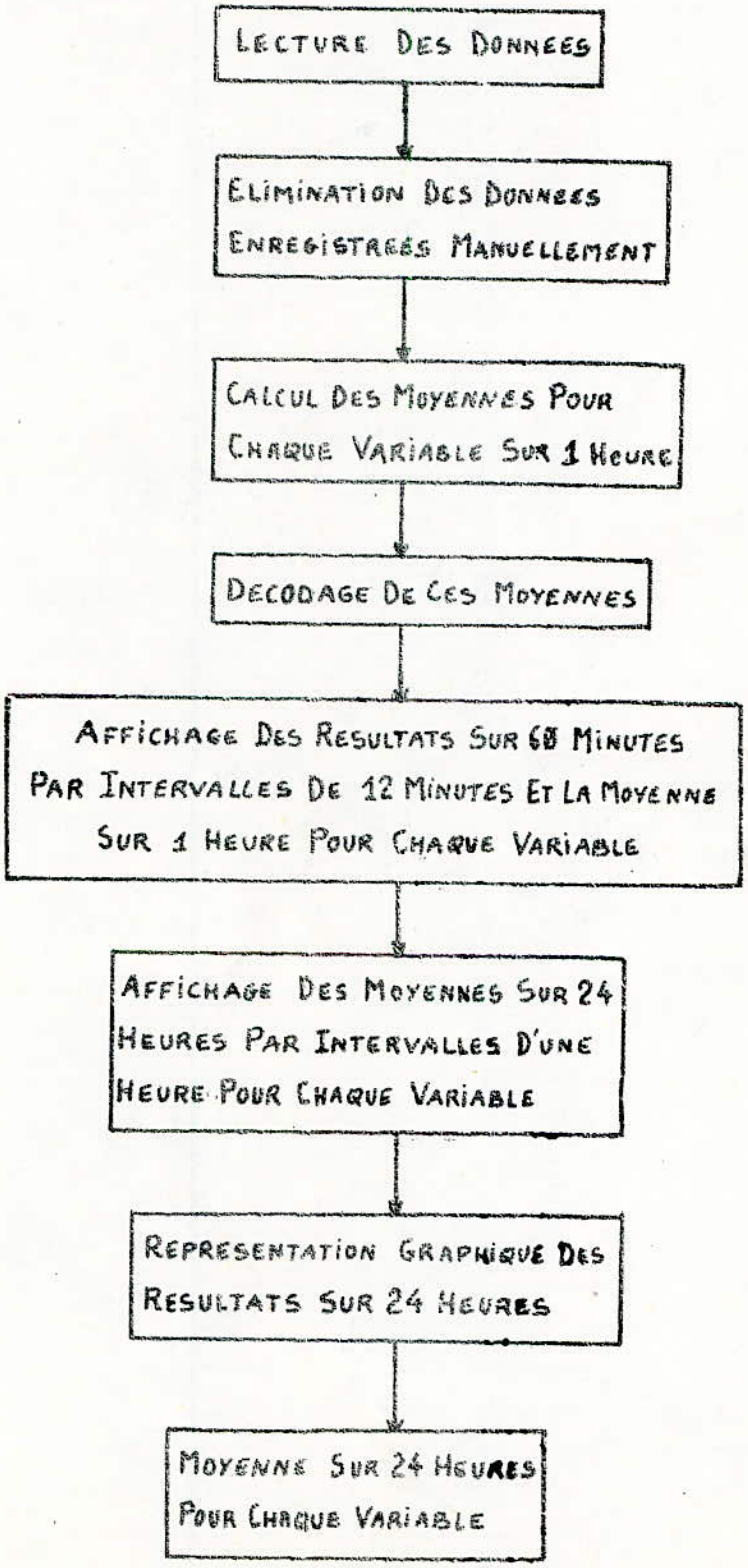
I - ELABORATION DE L'ALGORITHME DE TRAVAIL :

I.1 : Algorithme général :

L'algorithme général se décompose en huit étapes essentielles, telles qu'à chacune d'elles, onaura l'affichage d'un résultat bien précis sur l'écran avec une temporisation plus ou moins grande.

Nous avons préféré éliminer les mesures manuelles pour rendre le traitement moins compliqué ; de même le décodage s'effectue aprés le calcul des moyennes sur chaque heure. Enfin, les représentations graphiques pour chaque mesure et les moyennes sur 24 heures, seront affichées en dernier lieu car elles représentent l'essentiel du traitement.

L'algorithme général est donné dans la page suivante.


LECTURE DES DONNEES
ELIMINATION DES DONNEES ENREGISTREES MANUELLEMENT
CALCUL DES MOYENNES POUR CHAQUE VARIABLE SUR 1 HEURE
DECODAGE DE CES MOYENNES
AFFICHAGE DES RESULTATS SUR 60 MINUTES PAR INTERVALLES DE 12 MINUTES ET LA MOYENNE SUR 1 HEURE POUR CHAQUE VARIABLE
AFFICHAGE DES MOYENNES SUR 24 HEURES PAR INTERVALLES D'UNE HEURE POUR CHAQUE VARIABLE
REPRESENTATION GRAPHIQUE DES RESULTATS SUR 24 HEURES
MOYENNE SUR 24 HEURES POUR CHAQUE VARIABLE

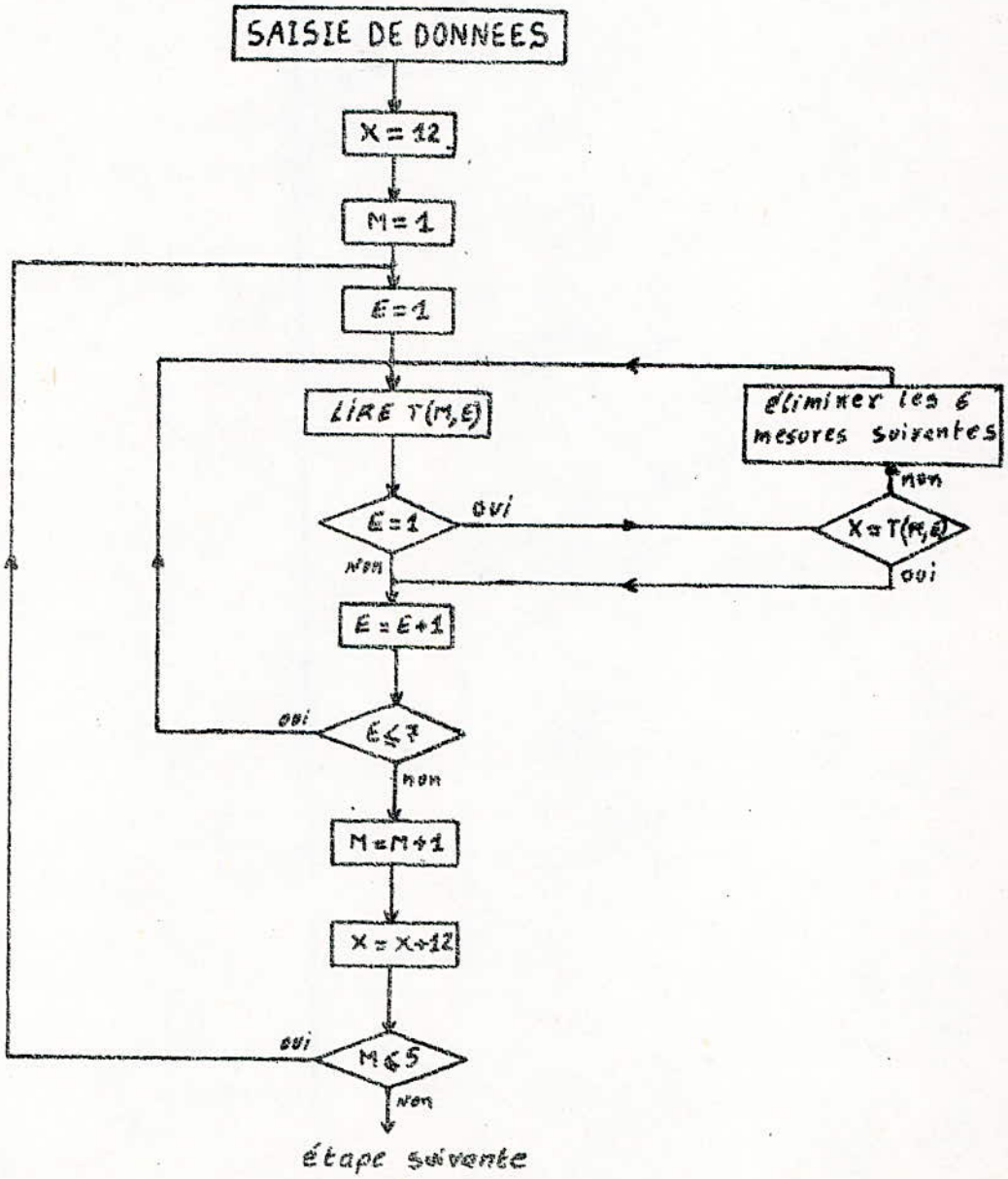
I_2: Organigrames:

Dans ce paragraphe on va décrire l'organigramme de traitement pour chaque étape.

I_2_1: Lecture de données et élimination de mesures manuelles:

les données, en mémoire RAM sous forme de DATA, seront classées dans des tableaux afin d'être traitées.

L'organigramme est le suivant:



I_2_2 : Calcul des moyennes pour chaque variable sur 1 heure avec décodage :

le décodage des mesures s'effectue suivant les formules qui se trouvent dans le document concernant l'enrégistreur.

Ces formules sont :

1_ Pour $I=1$, $M(I) = K$.

2_ Pour $I=2$, $M(I) = M(I)/20 - 20$.

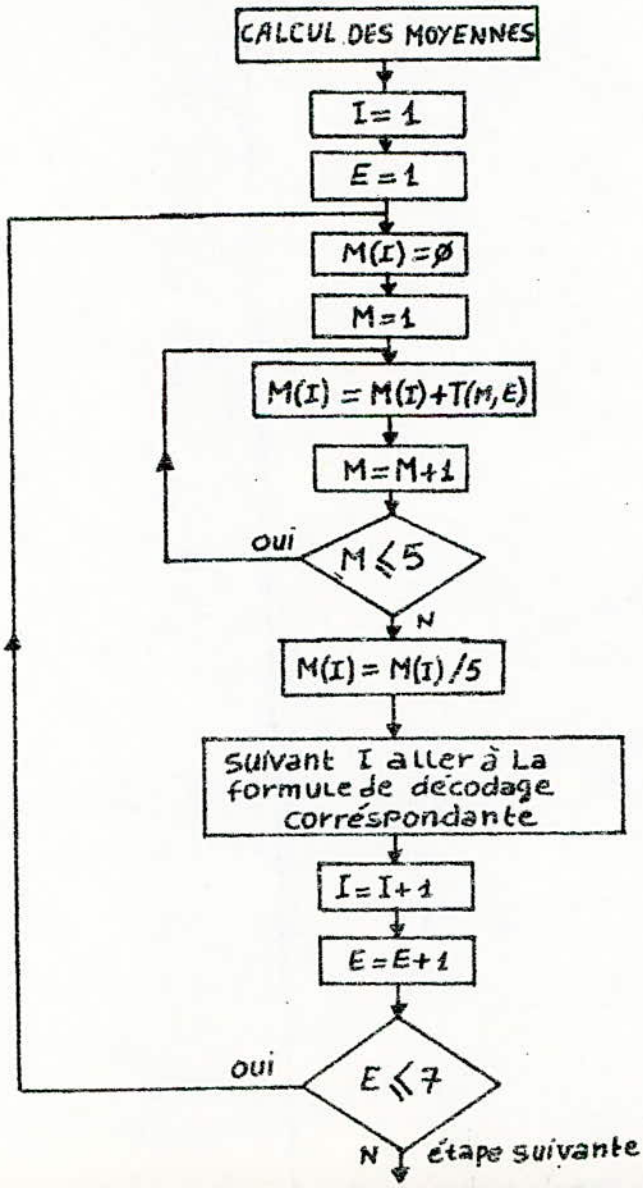
3_ Pour $I=3$, $M(I) = M(I)/20$.

4_ Pour $I=4$, $M(I) = M(I)/100$.

5_ Pour $I=5(6)$, $M(I) = M(I)/160$.

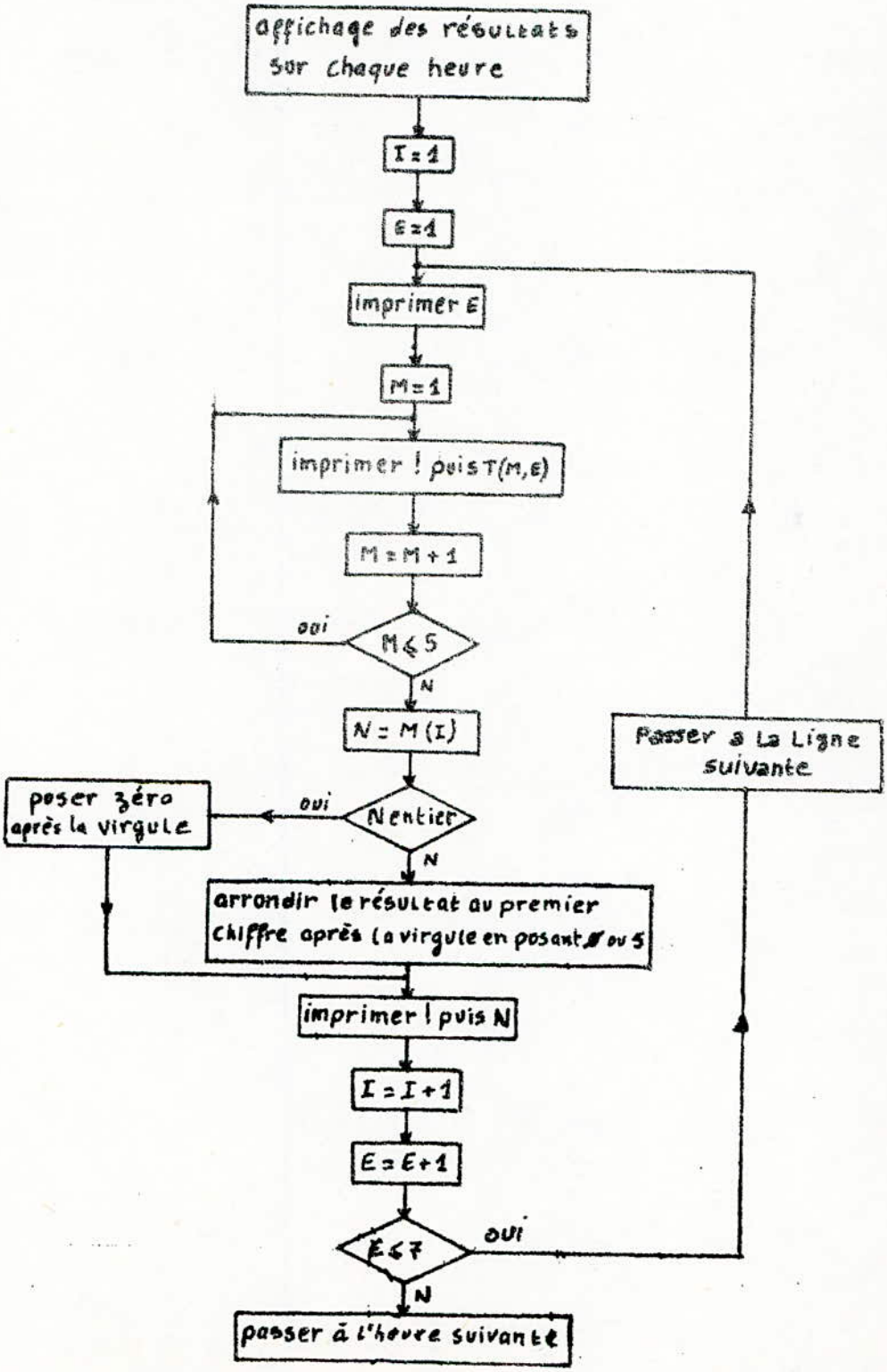
6_ Pour $I=7$, $M(I) = M(I)/16.38$.

Organigramme :



I.2.3: Affichage des résultats sur 60 minutes par intervalles de 12 minutes et de la moyenne sur une heure pour chaque variable.

organigramme:



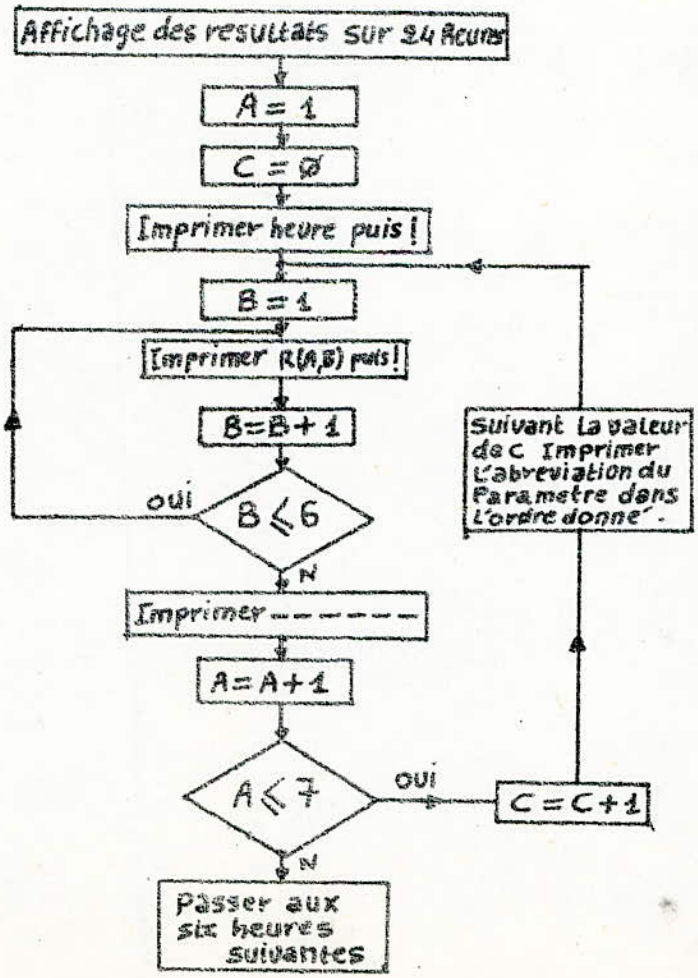
I_2_4 : Affichage des moyennes sur 24 heures par intervalles de heure Pour chaque variable .

Pour faciliter l'écriture sur l'écran, on désigne les differents paramètr Par leurs abréviations :

- Le temps : HEURE.
- la température ambiante : TAMB.
- La température des cellules : TCELL.
- l'ensoleillement global : WGH.
- la puissance : WL.
- l'ensoleillement direct : WD.
- le seuil : SEUIL.

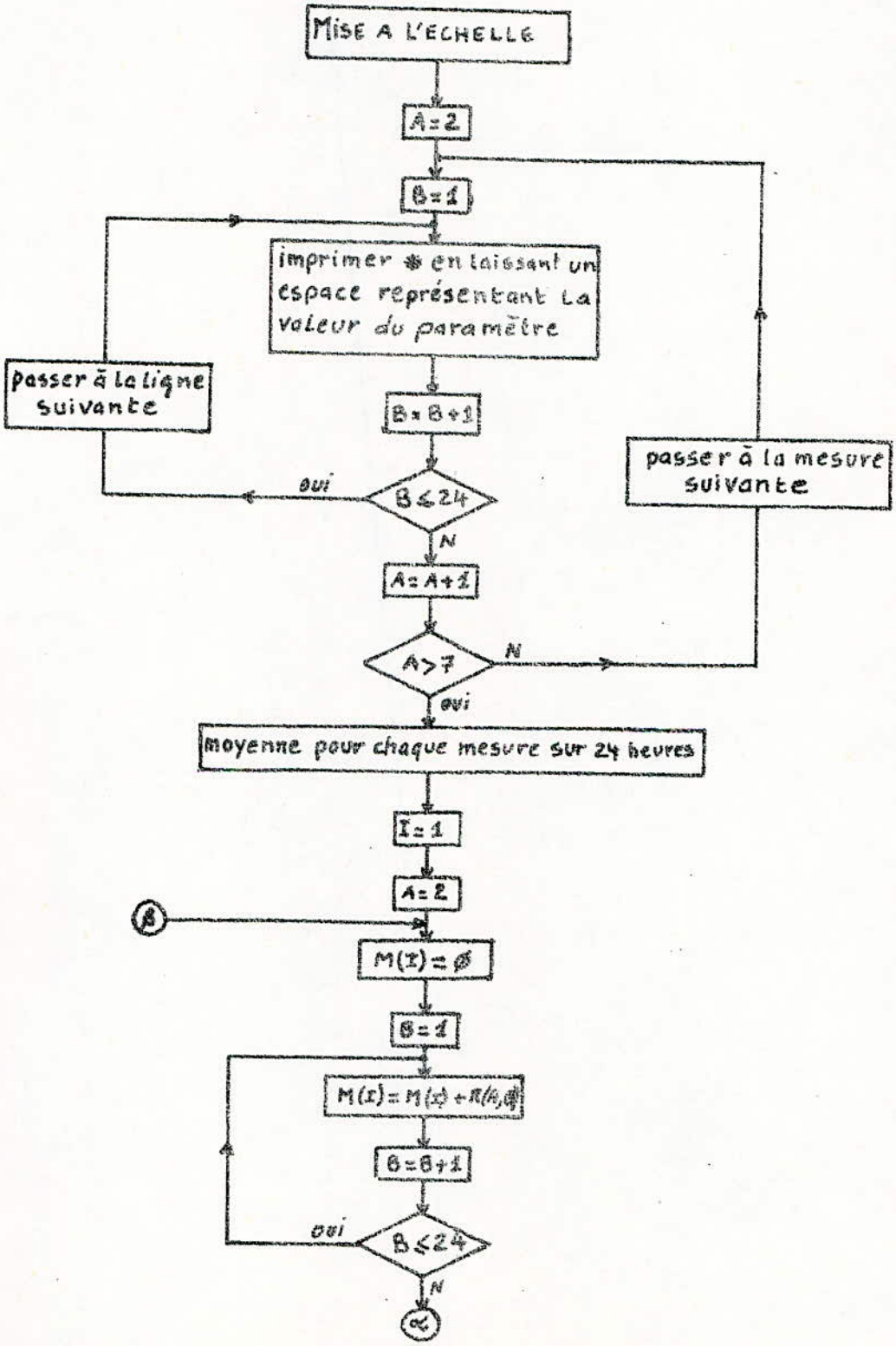
A cause de la largeur limitée de l'écran (40 caractéres par Lignes) on a préféré sortir les résultats par intervalles de 6 heures, ce qui fait 4 tableaux au total.

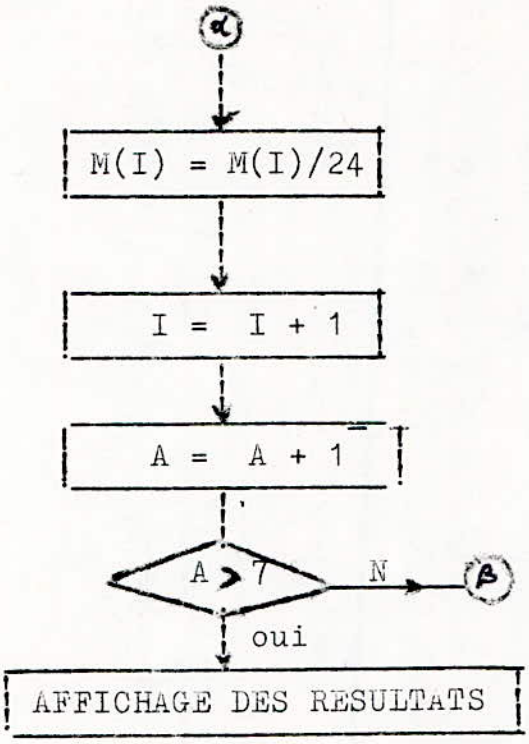
organigramme :



I_2.5: Représentation graphique des résultats sur 24 heures et affichage de la moyenne pour chaque mesure sur 24 heures:

Organigramme:





II - PROGRAMME DE TRAITEMENT :

II.1 : Exemple :

Pour faciliter la compréhension de notre travail de programmation, on va donner un exemple illustrant la méthode choisie pour le traitement de données.

On utilise à cette fin des mesures correspondants à une heure pour les six paramètres.

PROGRAMME :

```
IØ REM TRAITEMENT DE DONNEES
2Ø REM SUR 1 HEURE
25 DIM T(6,7),M(7)
28 FOR M = 1 TO 6
3Ø FOR E = 1 TO 7
35 READ T(M,E)
4Ø NEXT E, M
5Ø REM POUR LES DIFFERENTES MESURES
6Ø REM TRAITEMENT DE DONNEES
65 FOR E = 1 TO 7
7Ø FOR M = 1 TO 6
```

```
73 M(E) = M(E) + T (M,E)
74 NEXT M
75 M(E) = M(E)/6
78 NEXT E
8Ø LET M(1) = 1
82 REM SORTIE DES RESULTATS
85 FOR E = 1 TO 7
9Ø PRINT ; SPC (2 - LEN(STR$(E))) ; E ;
IOO FOR M = 1 TO 6
IIØ X$ = STR$ (T(M,E))
II4 PRINT "!" ; SPC (3 - LEN(X$)) ; X $ ;
II6 NEXT M
I2Ø N = INT (M(E))
122 X$ = STR$ (N)
124 PRINT "!" ; SPC (4 - LEN (X$)) ; X$
I15 PRINT ; " - - - - - - - - - - - - - - - - - - "
I3Ø NEXT E
I5Ø DATA ØØ , I2, I4, 3Ø, 4Ø, 35, IØ
I6Ø DATA 12 , I3, I3, 31, 39, 4Ø, 11
I7Ø DATA 24 , I3, I5, 34, 38, 35, 12
I8Ø DATA 36,  12, I6, 36, 3Ø, 29, 9
I9Ø DATA 48,  I3, I7, 37, 25, 3Ø, I3
2ØØ DATA 6Ø,  II, I8, 39, 2Ø, 29, IØ
2Ø5 STOP : GOTO 21Ø
21Ø REM GRAPHE POUR CHAQUE MESURE
215 E = 2 : M = 1
218 PRINT ; TAB (I6) ; "VARIABLE Y"
22Ø PRINT ; SPC (2) ; 1 ; "......." ; IØ ; "......." ; 2Ø ;
    "......." ; 3Ø ; "......." ; 4Ø
225 FOR X = Ø TO 6Ø STEP 12
23Ø X$ = STR$ (X)
24Ø PRINT ; SPC (2 - LEN(X$)) ; X$ ; TAB (T(M,E)) ; "*"
245 PRINT
25Ø M = M + 1
26Ø NEXT X
27Ø E = E + 1 : M = 1
```

```
28Ø IF E > 7 THEN 3ØØ
29Ø STOP : GOTO 218
3ØØ END
```

Ce programme sera exécuté en tapant l'instruction RUN puis RETURN. Les résultats affichés aprés exécution du programme sont les suivants :

```
1 ! Ø !12! 24! 36! 48! 6Ø! I .
- - - - - - - - - - - - - - -
2 ! I2!I3!_I3!_12!_I3!_11!12_
3 ! I4!13!_15!_16!_17!_18!15_
4 ! 3Ø!31!_34!_36!_37!_38!34_
5 ! 4Ø!39!_38!_3Ø!_25!_2Ø!32_
6 ! 35!4Ø!_35!_29!_3Ø!_29!33_
7 ! IØ!11! 12!  9! 13! 1Ø!1Ø
```

] BREAK IN 2Ø5

] CONT (Instruction permettant la reprise des programmes interrompus par l'instruction. stop)

```
            VARIABLE Y
I.......IØ .......2Ø .......3Ø .......4Ø
Ø          *
12          *
24          *
36         *
48          *
6Ø        *
```

] BREAK IN 29Ø

Pour passer aux autres paramètres, il suffit de taper l'instruction CONT puis RETURN.

II.2 : PROGRAMME :

On a commencé par le pas 4ØØ pour laisser la place au programme de chargement eventuel de données à partir de la cassette.

```
4ØØ FOR I = 1 TO 7
41Ø READ J$
42Ø NEXT I
43Ø REM TRAITEMENT DE DONNEES
44Ø REM SAISIE DE DONNEES
45Ø/DIM T(5,7) , R(7,24), M(7), N(7,24)
46Ø K = 1
47Ø B = 1
48Ø X = 12
49Ø I = 1
5ØØ IF B > 24 GOTO I21Ø
51Ø M = 1
52Ø E = 1
53Ø READ T(M,E)
54Ø IF E = 1 THEN 61Ø
55Ø E = E + 1
56Ø IF E ≤ 7 THEN 53Ø
57Ø M = M + 1
58Ø X = X + 12
59Ø IF M ≤ 5 THEN 52Ø
6ØØ GOTO 66Ø
6Ø5 REM ELIMINATION DE MESURES MANUELLES
61Ø IF X = T(M,E) THEN 55Ø
62Ø FOR Y = 1 TO 6
63Ø READ J$
64Ø NEXT Y
650 GOTO 52Ø
66Ø REM MESURES SUR 24 HEURES
67Ø REM TRAITEMENT DE DONNEES
675 REM CALCUL DES MOYENNES SUR CHAQUE HEURE
68Ø FOR E = 1 TO 7
69Ø M(I) = Ø
```

```
7ØØ FOR M = 1 TO 5
71Ø M(I) = M(I) + T(M,E)
72Ø NEXT M
73Ø M(I) = M(I)/5
74Ø ON I GOSUB 1Ø8Ø, 11ØØ, 112Ø, 114Ø, 116Ø, 116Ø, 118Ø
75Ø I = I + 1
76Ø NEXT E
77Ø I = 1
78Ø A = 1
79Ø REM SORTIE DES RESULTATS SUR CHAQUE HEURE
8ØØ FOR E = 1 TO 7
81Ø PRINT ; SPC (2 - LEN(STR$(E))) ; E ;
82Ø FOR M = 1 TO 5
83Ø X$ = STR$ (T(M,E))
84Ø PRINT "!" ; SPC (4 - LEN (X$)) ; X$ ;
85Ø NEXT M
86Ø N = M(I)
87Ø X$ = STR$ (N)
88Ø IF N = INT (N) THEN 92Ø
89Ø N = INT (N*2 + Ø.5)/2
9ØØ X$ = STR$ (N)
91Ø GOTO 93Ø
92Ø X$ = X$ + ".Ø"
93Ø R(A,B) = N
94Ø PRINT "!" ; SPC (6 - LEN(X$)) ; X$
95Ø PRINT " - - - - - - - - - - - - - - - - - - - - "
96Ø I = I + 1
97Ø A = A + 1
98Ø NEXT E
99Ø K = K + 1
1ØØØ B = B + 1
1Ø1Ø FOR N = 1 TO 1ØØØ
1Ø2Ø Y = 1
1Ø3Ø NEXT N
```

```
1Ø4Ø PRINT
1Ø5Ø PRINT " - - - - - - - - - - - - - - - - - - - "
1Ø6Ø PRINT
1Ø7Ø GOTO 49Ø
1Ø8Ø M(I) = K
1Ø9Ø RETURN
11ØØ M(I) = M(I)/2Ø - 2Ø
111Ø RETURN
112Ø M(I) = M(I)/2Ø
113Ø RETURN
114Ø M(I) = M(I)/1ØØ
115Ø RETURN
116Ø M(I) = M(I)/16Ø
117Ø RETURN
118Ø M(I) = M(I)/16.38
119Ø RETURN
12ØØ REM SORTIE DES RESULTATS SUR 24 HEURES
121Ø FOR N = 1 TO 1ØØØ
122Ø Y = 1
123Ø NEXT N
124Ø J = 6
125Ø I = 1
127Ø A = 1
1275 C = Ø
128Ø N$ = "HEURE"
129Ø PRINT ; SPC (5 - LEN(N$ )) ; N$ ; "!" ;
13ØØ GOTO I43Ø
131Ø N$ = "TAMB"
132Ø GOTO I29Ø
133Ø N$ = "TCELL"
134Ø GOTO 129Ø
135Ø N$ = "WGH"
136Ø GOTO 129Ø
137Ø N$ = "WEL"
138Ø GOTO 129Ø
```

```
I39Ø N$ = "WD"
14ØØ GOTO 129Ø
141Ø N$ = "SEUIL"
142Ø GOTO 129Ø
143Ø FOR B = I TO J
144Ø X$ = STR$ (R(A, B))
145Ø PRINT SPC ( 4 - LEN(X$)) ; X$ ; "!" ;
146Ø NEXT B
147Ø PRINT ; "    "
148Ø PRINT " - - - - - - - - - - - - - - - - - -"
149Ø A = A + 1
15ØØ C = C + 1
151Ø ON C GOTO 131Ø, 133Ø, 135Ø, 137Ø, 139Ø, 141Ø
152Ø FOR N = 1 TO IØØØ
153Ø Y = 1
154Ø NEXT N
155Ø I = I + 6
156Ø J = J + 6
1565 PRINT
1567 PRINT
157Ø IF J > 24 THEN 159Ø
158Ø GOTO 127Ø
159Ø STOP : GOTO I6ØØ
16ØØ REM MISE A L'ECHELLE
161Ø A = 2
162Ø GOSUB 174Ø
163Ø A = A + 1
164Ø GOSUB 178Ø
165Ø A = A + 1
166Ø GOSUB 182Ø
167Ø A = A + 1
168Ø GOSUB I86Ø
169Ø A = A + 1
17ØØ GOSUB I9ØØ
171Ø A = A + 1
172Ø GOSUB 178Ø
```

```
173Ø GOTO I94Ø
174Ø FOR B = 1 TO 24
175Ø R(A,B) = Ø.5 * R(A, B)
176Ø NEXT B
177Ø RETURN
178Ø FOR B = 1 TO 24
179Ø R(A, B) = Ø.4 * R(A, B)
I8ØØ NEXT B
1Ø1Ø RETURN
182Ø FOR B = 1 TO 24
183Ø R(A,B) = R(A,B) + 15
184Ø NEXT B
185Ø RETURN
186Ø FOR B = 1 TO 24
187Ø R(A,B) = R(A,B) + 2Ø
188Ø NEXT B
189Ø RETURN
19ØØ FOR B = 1 TO 24
191Ø R(A,B) = R(A,B) + 1Ø
192Ø NEXT B
193Ø RETURN
194Ø REM GRAPHE POUR CHAQUE MESURE SUR 24 HEURES
195Ø A = 2
196Ø C = Ø
197Ø N$ = "TAMB"
198Ø A$ = "Y = .5*YR"
199Ø B = 1
2ØØØ PRINT ; SPC(1) ; 1 ; SPC(5 - LEN(N$)) ; N$ ; ".." ;
     1Ø ; "......." ; 2Ø ; A$ ; 3Ø ; "......." ; 4Ø
2Ø1Ø FOR X = 1 TO 24
2Ø2Ø X$ = STR$ (X)
2Ø3Ø N(A,B) = INT(R(A,B))
2Ø35 IF N(A,B)<1 THEN GOSUB 271Ø
2Ø4Ø PRINT ; SPC (2 - LEN (X$)) ; X$ ; TAB(N(A,B)) ; "*"
2Ø5Ø B = B + 1
2Ø6Ø NEXT X
```

```
2Ø7Ø A = A + 1
2Ø8Ø IF A > 7 THEN 229Ø
2Ø9Ø FOR N = 1 TO 2ØØØ
21ØØ Y = 1
211Ø NEXT N
2115 PRINT
212Ø C = C + 1
213Ø ON C GOTO 214Ø , 217Ø, 22ØØ, 223Ø, 226Ø
214Ø N$ = "TCELL"
215Ø A$ = "Y = .4*YR"
216Ø GOTO I99Ø
217Ø N$ = "WGH"
218Ø A$ = "Y = YR + 15"
219Ø GOTO I99Ø
22ØØ N$ = "WEL"
221Ø A$ = "Y = YR + 2Ø"
222Ø GOTO I99Ø
223Ø N$ = "WD"
224Ø A$ = "Y = YR + 1Ø"
225Ø GOTO I99Ø
226Ø N$ = "SEUIL"
227Ø A$ = "Y = .4*YR"
228Ø GOTO I99Ø
229Ø FOR N = 1 TO 2ØØØ
2292 Y = 1
2294 NEXT N
23ØØ REM MOYENNE POUR CHAQUE MESURE SUR 24 HEURES
231Ø I = 1
232Ø A = 2
2325 M(I) = Ø
233Ø FOR B = 1 TO 24
234Ø M(I) = M(I) + R(A,B)
235Ø NEXT B
236Ø M(I) = M(I)/24
237Ø I = I + 1
```

```
238Ø A = A + 1
239Ø IF A > 7 THEN 241Ø
24ØØ GOTO 2325
241Ø I = 1
242Ø PRINT ; TAB (9) ; "MOYENNES SUR 24 HEURES"
243Ø PRINT
244Ø N = M(I)
245Ø X$ = STR$ (N)
246Ø IF N = INT (N) THEN 25ØØ
247Ø N = INT (N*2 + Ø.5)/2
248Ø X$ = STR$ (N)
249Ø GOTO 251Ø
25ØØ X$ = X$ + ".Ø"
251Ø ON I GOTO 252Ø, 255Ø, 258Ø, 261Ø, 264Ø, 267Ø
252Ø PRINT ; "TAMB" ;"- - - - - - - - - - - - - - - - " ; X$
253Ø PRINT
254Ø I = I + 1 : GOTO 244Ø
255Ø PRINT ; "TCELL" ; " - - - - - - - - - - - - - - " ; X$
256Ø PRINT
257Ø I = I + 1 : GOTO 244Ø
258Ø PRINT ; "WGH" ; "  - - - - - - - - - - - - - - - " ; X$
259Ø PRINT
26ØØ I = I + 1 : GOTO 244Ø
261Ø PRINT ; " WEL" ; " - - - - - - - - - - - - - - - " ; X$
262Ø PRINT
263Ø I = I +1 : GOTO 244Ø
264Ø PRINT ; "WD" ; " - - - - - - - - - - - - - - - - " ; X$
265Ø PRINT
266Ø I = I + 1 : GOTO 244Ø
267Ø PRINT ; "SEUIL" ; " - - - - - - - - - - - - - - - " ; X$
268Ø PRINT
269Ø PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - "
27ØØ END
271Ø N (A,B) = N(A,B) + 1
272Ø RETURN
```

III - ENREGISTREMENT SUR CASSETTE :

III.1 : L'interface cassette de la carte-mère :

A l'arrière de la carte principale, prés du connecteur vidéo se trouvent deux petits boîtiers marqués "IN" et "OUT". Ce sont deux petites prises dans lesquelles on peut broncher un cable possedant une paire de prises mâles à chaque extrémité. L'autre extrémité du cable peut se broncher sur un magnétophone standard, ce qui nous permet de sauvegarder des programmes sur des cassettes audio pour les relire ensuite.

Le connecteur marqué "OUT" est relié à un autre commutateur situé sur la carte principale, ce commutateur peut-être basculé en se referant à l'adresse $C02$. Une référence à cette adresse commandera au potentiel de la prise "OUT" de monter de zéro à 25 mV ou de tomber de 25 mV à zéro. En utilisant de façon répétitive cette adresse, un programme peut produire un son sur l'enregistrement.

L'autre connecteur marqué "IN" peut-être utilisé pour écouter une cassette enregistée. On l'utilise pour écouter les sons provenant de la bande, les décoder en données qui sont alors stockées en mémoire.

Le circuit d'entrée necessite un signale d'un volt (crête à crête), obtenu sur la prise écouteur du magnétophone, et qu'il traduit en une suite de 1 et de $. A chaque fois que le signale fournit en entrée passe d'une valeur positive à une valeur négative, le circuit d'entrée change d'état : s'il envoyait jusque là du 1, il envoie alors du $ et réciproquement.

III.2 : ETUDE DE L'ENREGISTREMENT :

L'enregistrement sur cassette peut se faire de deux manière différentes, soit en utilisant les instructions du langage BASIC, soit le moniteur.

Le dernier cas est moins interessant si on veut enregistrer un programme, puisqu'aprés lecture, on perd totalement le contrôle du système, au contraire son utilisation serait utile dans l'étude du code fréquentiel que l'on verra dans la troisième partie.

III.2.1 : Utilisation du moniteur :

On peut à l'aide du moniteur sauvegarder de grandes

quantités de données et de programmes sur cassette puis les relire par simple commande.

Les commandes qui réalisent ces fonctions sont les suivantes :

- WRITE : Nous permet de sauvegarder sur cassette le contenu d'un bloc de 1 à 65536 mémoires à l'aide d'un magnétophone courant. Pour sauvegarder un bloc de mémoires sur cassette, il faut préciser au moniteur l'adresse de début, l'adresse de fin, et les faire suivre de la W (Write).

{début} . {fin} W

- READ : Qui permet de relire pour réintroduire dans l'APPLE un bloc de mémoire sauvegardé sur cassette. Il n'est pas obligé de replacer le bloc de valeurs dans l'endroit où il se trouvait lors de la sauvegarde.

On peut demander au moniteur de le ranger dans n'importe quel bloc de même taille de la mémoire de l'APLLE.

Le format de la commande READ est identique à celui de la commande WRITE à l'exception de W remplacé par R.

{début} . {fin} R

III.2.2 : Utilisation du langage BASIC :

Pour sauvegarder dans ce cas un programme il suffit de taper l'ordre SAVE, aprés RETURN le curseur disparait de l'écran, un premier beep indiquant le début de l'enregistrement se fait entendre aprés une dizaine de secondes et un second beep, une dizaine de secondes aprés le premier, en même temps que réapparait le curseur (l'intervalle du temps séparant les deux beep dépend de la langueur du programme).

L'opération inverse de chargement LOAD d'un programme a partir de la cassette vers la mémoire de l'ordinateur est plus délicate et demande un reglage des niveaux de volume et de tonalité, en général volume moyen et tonalité aigue. D'autre part, il est nécessaire de connaître assez précisement la position sur la bande où a été enregistré le programme qu'on veut charger.

III.3 : Commandes et programme d'enregistrement et de lecture des données sur cassette :

III.3.1 : Sauvegarde de tableaux de données sur cassette :

Un tableau dimensionné par DIM P(X,Y,Z,...) est enregistrable sur cassette par l'instruction STORE P, et, de façon inverse l'instruction RECALL P rappelle en mémoire les valeurs concervées sur cassette.

On utilisera un tableau à une dimension, ce qui est le cas pour un fichier séquentiel en mémoire vive.

Dans ce cas les données ne peuvent être de type alphanumérique, il n'est pas nécessaire d'utiliser le même nom de tableau pour relire les données, mais il est obligatoire que le tableau recepteur ait les mêmes dimensions que le tableau émetteur.

III.3.2 : Sauvegarde des chaînes de caractères sur cassette :

Le traitement des chaînes est plus souple que celui des données numériques, c'est la raison pour laquelle on fait la transformation valeurs numériques - chaînes de caractères dans certains programmes, et il serait trés utile de connaître le programme d'enregistrement ou de lecture d'une chaîne de caractère

Le sous programme d'enregistrement de données sur cassette débute en $FECD, s'appelle par CALL - 259.

Le sous programme de lecture de données sur cassette débute en $FEFD, s'appelle par CALL - 3Ø7.

Les paramètres demandés par ces sous programmes sont :

- L'adresse de début de la zone mémoire,
- l'adresse de fin de la zone mémoire,

à enregistrer ou à charger.

Ils sont à charger dans les adresses $3C, $3D et $3E, $3F. La longueur de la zone des chaînes de caractères est déterminée par différence entre HIMEM et DC contenues dans $73, $74, et $6F, $7Ø.

Cette langueur aprés calcul sera enregistrée la première sur la cassette, en la placant temporairement en $3Ø et $31.

Le programme d'enregistrement et de lecture sur cassette d'une chaîne de caractères est donné ci-dessous :

```
1Ø REM SAUVEGARDE DE CHAINES DE CARACTERES
15 REM SUR CASSETTE
2Ø DIM A$ (1ØØØ)
3Ø FOR K = 1 TO 96Ø : READ A$ (K) : NEXT K
4Ø GOSUB 15Ø
5Ø PRINT "LES CHAINES SONT ENREGISTREES"
6Ø PRINT "POUR LES RECHARGER EN RAM"
7Ø PRINT "TAPER GOTO 1ØØ, REMBOBINEZ"
8Ø PRINT "DEMARRER LA CASSETTE PUIS RETURN"
9Ø END
1ØØ REM CHARGEMENT EN MEMOIRE
11Ø DIM B$(1ØØØ)
12Ø GOSUB 29Ø
13Ø FOR K = 1 TO 96Ø : PRINT B$(K) : NEXT K
14Ø END
15Ø REM SAUVEGARDE DE A$
16Ø PRINT "PREPARER L'ENREGISTREMENT"
17Ø PRINT "COMMENCEZ L'ENREGISTREMENT"
18Ø GET Z$
19Ø PP X = FRE (Ø) : STORE A$ : REM ENREGISTRE LES VRAIS
    POINTEURS DE A$
2ØØ XH = PEEK (115) + PEEK(116) * 256 : REM HIMEM
21Ø XDC = PEEK (111) + PEEK(112) * 256 : REM DC
22Ø XL = XH - XDC : GOSUB 38Ø
23Ø POKE 3Ø, XL - INT (XL/256)*256
24Ø POKE 31, XL/256
25Ø CALL - 3Ø7 : REM ENREGISTREMENT DE XL
26Ø POKE 6Ø, PEEK (111) : POKE 61, PEEK (112)
27Ø POKE 62, PEEK (115) : POKE 63, PEEK (116)
275 CALL - 3Ø7 : REM ENREGISTREMENT DES CHAINES
28Ø PRINT "C'EST ENREGISTRE" : RETURN
```

```
29Ø RECALL BØ : REM CHARGEMENT DES POINTEURS
3ØØ GOSUB 38Ø : CALL - 259 : REM LONGUEUR
3Ø5 X = PEEK (3Ø) + PEEK (31) * 256
31Ø  X = PEEK (115) + PEEK (116) * 256 - X
32Ø POKE 6Ø, X - INT (X/256) * 256
33Ø POKE 61, X/256
34Ø POKE 62, PEEK(115)
35Ø POKE 63, PEEK(116)
36Ø CALL - 259 : REM CHAINES LUES
37Ø RETURN

38Ø POKE 6Ø, 3Ø : POKE 61, Ø : POKE 62, 31 : POKE 63, Ø
39Ø RETURN
```

Pour ces deux cas, il y'aura des petites modifications dans le programme principale lors du traitement de données enregistrées sur cassette.

On s'intersse uniquement au premier cas concernant la sauvegarde d'un tableau de données. Les instructions à ajouter et à modifier sont les suivantes :

```
25Ø REM LECTURE ET AFFICHAGE DE DONNEES
26Ø DIM A % (1ØØØ)
27Ø RECALL A %
28Ø FOR J = 1 TO 968
29Ø PRINT A % ; "  " ;
3ØØ NEXT J
31Ø REM ELIMINATION DES ZEROS
315 REM DES DEBUTS DE CYCLES
32Ø DIM B % (1ØØØ)
33Ø T = 1
34Ø J = 1
35Ø I = 1
36Ø IF I = T THEN 4Ø5
37Ø B % (J) = A % (I)
38Ø I = I + 1
39Ø J = J + 1
```

```
4ØØ GOTO  36Ø
4Ø5 I = I + 1
41Ø T = T + 8
415 IF J > 847 THEN 425
42Ø GOTO 36Ø
425 PRINT
427 PRINT
428 PRINT " - - - - - - - - - - - - - - - - - - - - - - - - - - - - "
455 J = 8
53Ø T (M,E) = B % (J)
62Ø
63Ø J = J + 6
64Ø
```

Le pas vide signifie la suppression de l'instruction correspondante.

TROISIEME PARTIE

ETUDE DE L'INTERFACE CASSETTE

L'mportance dans cette étude est de déterminer d'abord les deux codes d'enregistrement de L'apple II et de l'enregistreur ALEP.ED 7620 ensuite de proposer un schéma synoptique pour l'interface cassette.

I - PRESENTATION DE L'ENREGISTREUR ALEP.ED 7620. :

I - 1 Déscription Générale :

L'enregistreur ALEP.ED 7620 est conçu env vue de l'enregistrement sur cassette Standard toutes les 12 minutes, de la valeur d'une grandeur physique.

Il est prévu 7 voies d'enregistrement affectées suivants le tableau ci-après.

Les voies 2 et 3 représentent des mesures instantanées, les voies 3 à 6 les valeurs moyennes sur la période d'intégration, soit 12 minutes.

L'ensemble des circuits électroniques est supporté par deux cartes de circuit imprimé :

a) - Carte 7619 pour la partie analogique et les alimentations .
b) - Carte 7620 pour la partie digitale : base de temps et logique.

I - 2 : Fonctionnement :

L'intervalle entre les enregistrements est fixé à 12 minutes.le compteur d'intervalle actionne par l'intermédiaire d'un monostable une bascule qui autorise par sa sortie, l'alimentation du magnétophone et autorise la décade de séquence préliminaire, cette décade génére les signaux afférents aux opérations suivantes :

1°) - Au bout de 3 secondes, le transfert du contenu des compteurs de stockage dans les compteurs de lecture.

2°) - Au bout de 3,5 secondes, la remise à zéro des compteurs de stockage

3°) - Au bout de 4 secondes, actionne la bascule qui commande le déroulement de la bande et autorise la scrutation des voies.

4°) - Se remet en position d'attente, la décade de scrutation est alors libérée et autorise le passage des informations vers la tête d'enregistrement, successivement : identification de début de cycle, codage horaire, voies de mesure, puis stoppe le cycle.

I - 3 : Commandes manuelles :

1°) - ARRET - MARCHE : Utilisé pour la mise en service.

2°) - Remise à zéro base de temps : permet de fixer l'origine de la rampe de codage horaire à l'instant de la mise en service.

3°) - Cycle manuel : utilisé au début d'une période d'enregistrement à titre de vérification du fonctionnement ainsi que pour l'introduction du papier dans le mécanisme d'impression de l'imprimante IBM 7612.

TABLEAU DE NORMALISATION DES VOIES :

VOIE N°	FONCTION	ECHELLE	DECODAGE	OBSERVATIONS
1	Codage Horaire	0 à 24 Heures	0 à 1440	Voie interne
2	Thermomètre	-20 à + 80°C	0 pour - 20°C 400 pour 0°C 2000 pour 80°C	Mesure instantanée
3	Thermomètre	0 à 100°C	0 à 2000	Mesure instantanée
4	Pyranomètre	0 à 16 mv	0 à 1600	Mesure intégrée sur 12 mn
5	Wattmètre	0 à 10 v	0 à 1600	" " "
6	Pyrhéliomètre	0 à 10 mv	0 à 1600	" " "
7	Seuil	0 à 100%	0 à 1638	Voie interne.

Note 1 : la voie 7 permet de connaitre la fraction de la durée d'intégration durant laquelle le signal sur la voie 6 à été supérieur à 40 % de la pleine échelle (Seuil ajusté d'origine à 4 mv).

Note 2 : Afin de satisfaire à la durée d'intégration spécifiée, l'intervalle entre enregistrement est fixé à 12 minutes.

II - DETERMINATION DES CODES D'ENREGISTREMENT :

Pour la détermination de ces codes, on a utilisé un oscilloscope à mémoire, le TEKTRONIX 7633.

II - 1 : Forme des données sur le support :

Les formes les plus courantes d'enregistrements de données sur le support sont les suivantes :

- Forme "ASC II" (également appelée "Symbolique" ou "Teletype").
- Forme "Binaire".

En forme ASC II, les données sont inscrites en employant un codage du caractère identique à celui du terminal.

En forme binaire, les données sont inscrites en employant un code spécifique à l'ordinateur (la plupart des systèmes d'exploitation imposent une limite maximum à la longueur des enregistrements, souvent une valeur numérique est représentée sur deux "mots" d'ordinateur).

Le choix de l'une ou l'autre forme, quand il est laissé à l'utilisateur, entraine en particulier les conséquences suivantes :

- La forme ASC II permet de créer un fichier puis de l'éditer sous le contrôle d'un programme utilitaire standard, l'emploi de cette forme donne la possibilité de correction de données.

- Les données numériques occupent plus de place sur le support quand elles sont exprimées en ASC II que lorsqu'elles sont en binaire, cette dernière forme est donc plus économique.

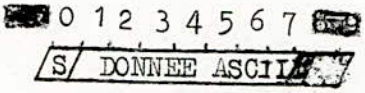
- Quand les données sont fournies en ASC II puis enregistrées en binaire, l'ordinateur doit effectuer la transformation de code (également pour la présentation des données au terminal, après traitement) cela peut amener à consommer du temps d'unité centrale.

Le tableau qui suit, donne le code binaire et son équivalent ASC II pour les chiffres de 0 à 9 :

CARACTERE	CODE BINAIRE	CODE ASC II
0	0000	0011 0000
1	0001	0011 0001
2	0010	0011 0010
3	0011	0011 0011
4	0100	0011 0100
5	0101	0011 0101
6	0110	0011 0110
7	0111	0011 0111
8	1000	0011 1000
9	1001	0011 1001

II - 2 : Cas de l'Apple II :

Le caractère issu du clavier est sur 8 bits et possède la forme suivante :



Les 7 bits de droitecontiennent le code ASC II de la touche frappée. le huitième bit, à l'extrême gauche, est appelé le "KEYBORD STROBE", il est généré par le décodeur des touches du clavier et prend la valeur 1 chaque fois qu'une touche a été frappée.

Pour la lecture sur cassette, seul le bit le plus à gauche est utile et représente le signal lu sur la cassette et transmis depuis la sortie EARPHONE j'usqu'au connecteur marqué IN à l'arrière de l'Apple, ce bit sera testé par un programme du Moniteur et décodé lors de l'instruction READ entre autres.

On va voir que la forme d'enregistrement sera différente si on utilise le langage Basic Applesaft.

II – 2 – 1 : Code des Fréquences :

A l'aide du Moniteur, on a enregistré la totalité de la RAM qui contient, des suites alternés des 0 et 1 lorsqu'elle est vide.

Les résultats obtenus au moyen de l'oscilloscope sont les suivants :

Fréquences des Zéros : F0 = 2 khz.

Fréquences des Uns : F1 = 1 khz.

II – 2 – 2 : Forme d'enregistrement de données :

1 – Réprésentation interne des nombres :

Comme toute unité arithmetique d'ordinateur, celle du microprocesseur de l'Apple II opère sur des registres de calcul de longueur fixe et limitée, la longueur des registres est de 8 Bits, pour augmenter cette longueur, et augmenter la précision et l'etendue des résultats de calcul, l'Apple fait intervenir des sous – programmes d'arithmétique étendue.

Les nombres entiers, par exemple, sont enregistrés en mémoire sur deux octets, les nombres réels sont enregistrés sur 5 octets, étudions leur principe de réprésentation.

– Représentation des nombres entiers :

Les entiers positifs sont représentés par simple numérotation binaire.sur 15 bits, et donc leur étendue va de 0 à $2^{15} - 1 = 32767$. le bit le plus à gauche , donc le Sezième, est mis à Zéro.

Les entiers négatifs sont représentés par leur complément à $\left(2^{16}\right) = 65536$ soit : – 1 devient (65535) 10 = (FFFF) 16

– 2 devient (65534) 10 = (FFFE) 16

jusqu'à : – 32767 devient (32769) 10 = (8001) 16.

Cette représentation est utilisée pour faciliter les opérations arithmétiques, le bit le plus à gauche est toujours à 1 pour les entiers négatifs

- Représentation des nombres réels :

La notation en virgule flottante diminue considérablement le nombre de chiffres pour représenter un nombre fractionnaire ou réel.

En Applesoft, il est possible d'enregistrer en mémoire des nombres réels allant jusqu'à 9 chiffres significatifs et pouvant être, en valeur absolue

Aussi Petit que 10^{-38}

Aussi grand que 10^{+38}

Pour obtenir une telle précision et un tel domaine de variation, les nombres réels sont codés en mémoire avec 5 octets repartis en 1 octet pour l'exposant et 4 octets pour la mantisse.

Si E est la valeur décimale de l'exposant, M la valeur décimale de la mantisse et N la valeur décimale du nombre représenté.

- $127 < E \leq 127$

$2^{-32} < M < 0,5$

Alors : $N = (M+0,5)\ 2^{E-128}$

est la formule permettant de déterminer la valeur absolue de N représentée en mémoire par E et M.

2 - Forme d'enregistrement de données :

Dans la mémoire RAM de l'Apple, on a crée 10 tableaux contenant chacun une suite d'un seul chiffre de 0 à 9, a l'aide de l'instruction STORE on les a enregistrées sur cassette. Dans un premier cas comme des réels ;

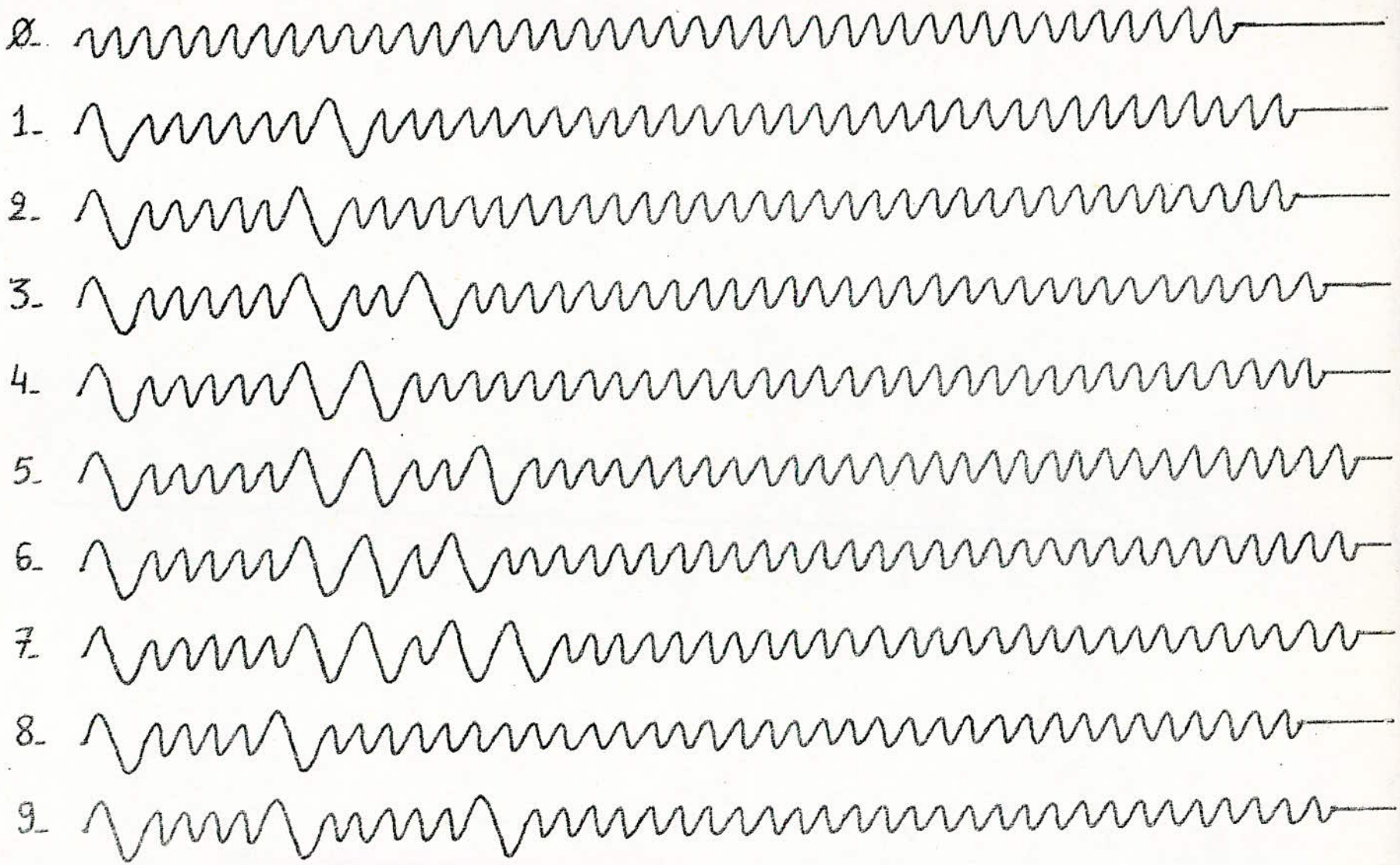
Dans le deuxième cas, comme des entiers positifs.

Les résultats obtenus à l'aide de l'oscilloscope à mémoire sont représentés sur les graphes suivants :

D'après le code des fréquences, on aura la correspondance binaire suivante pour chaque chiffre.

FORME D'ENRÉGISTREMENT DE DONNÉES

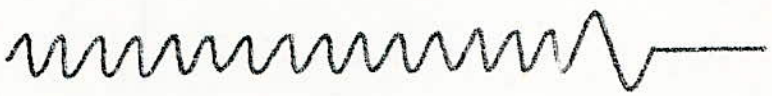
1er CAS

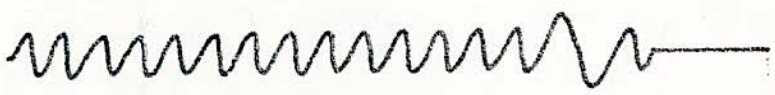


FORME D'ENRÉGISTREMENT DE DONNÉES

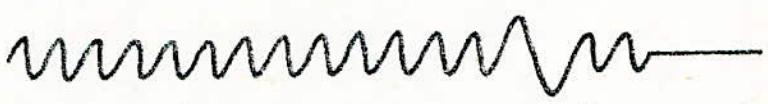
2ème CAS

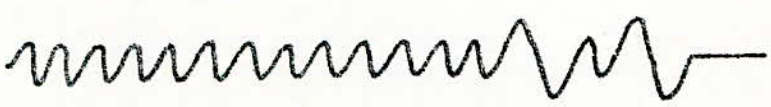
Ø_ 

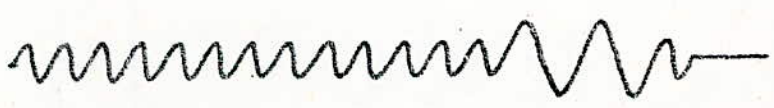
1_ 

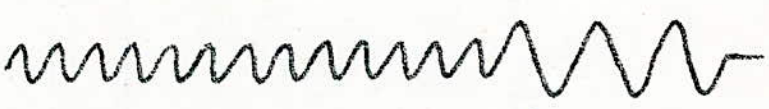
2_ 

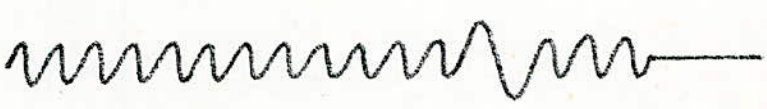
3_ 

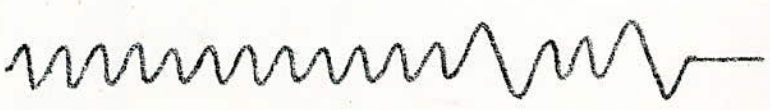
4_ 

5_ 

6_ 

7_ 

8_ 

9_ 

Premier cas :	Exposant	Mantisse			
0 –	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
1 –	1000 0001	0000 0000	0000 0000	0000 0000	0000 0000
2 –	1000 0010	0000 0000	0000 0000	0000 0000	0000 0000
3 –	1000 0010	0100 0000	0000 0000	0000 0000	0000 0000
4 –	1000 0011	0000 0000	0000 0000	0000 0000	0000 0000
5 –	1000 0011	0010 0000	0000 0000	0000 0000	0000 0000
6 –	1000 0011	0100 0000	0000 0000	0000 0000	0000 0000
7 –	1000 0011	0110 0000	0000 0000	0000 0000	0000 0000
8 –	1000 0100	0000 0000	0000 0000	0000 0000	0000 0000
9 –	1000 0100	0001 0000	0000 0000	0000 0000	0000 0000

En mémoire, la mantisse et l'exposant sont représentés sous forme héxadécimale, et pour calculer N_{10}, on doit faire la transposition ; $(E)_{16} \longrightarrow E_{10}$ et $M_{16} \longrightarrow M_{10}$; comme on peut la voir dans l'exemple qui suit concernant le chiffre 1.

(E) 16 (M) 16

/ 81 / / ØØ / ØØ / ØØ / ØØ /

$E_{10} = 8 \times 16 + 1 = 129$

$E - 128 = 129 - 128 = 1$

$2^{E-128} = 2^{1} = 2$

$M_{10} = Ø$

$N = (0,5) \times 2 = 1$

Les résultats trouvés ci-dessus, concordent parfaitement avec la représentation interne des nombres réels.

Deuxième Cas :

0 –	0000 0000	0000 0000
1 –	0000 0000	0000 0001
2 –	0000 0000	0000 0010
3 –	0000 0000	0000 0011
4 –	0000 0000	0000 0100
5 –	0000 0000	0000 0101
6 –	0000 0000	0000 0110
7 –	0000 0000	0000 0111
8 –	0000 0000	0000 1000
9 –	0000 0000	0000 1001

de même, ces résultats concordent aussi avec la représentation interne des nombres entiers positifs.

- Conclusion :

D'après les résultats trouvés précédemment, on remarque que l'enregistrement des nombres réels ou entiers a une forme binaire, mais pour notre étude ce dernier cas est plus interessant car il occupe moins de place : deux bytes uniquement, en plus, le codage est simple et ne nécessite pas une formule particulière, ce qui s'implifie énormément le transcodage enregistreur - Apple.

II - 3 : Cas de l'enregistreur :

Le diagramme des états logiques est donné ci-après.

Les différents signaux de commande sont les suivants :

DC : début de Cycle.

ECH : Echantiffonage des voies instantanées (Voies 2 et 3).

TR : Impulsion de transfert des données.

RZC : Remise à Zéro des compteurs de stockages.

DS : Début de scrutation de voies.

CV : Commutation de voie chaque 0,75 Seconde.

DV : Durée de voie.

ID : Impulsion d'identification.

SV : Scrutation de voie.

FC : Fin de Cycle d'enregistrement.

La fréquence d'enregistrement de données est unique et est égale à :

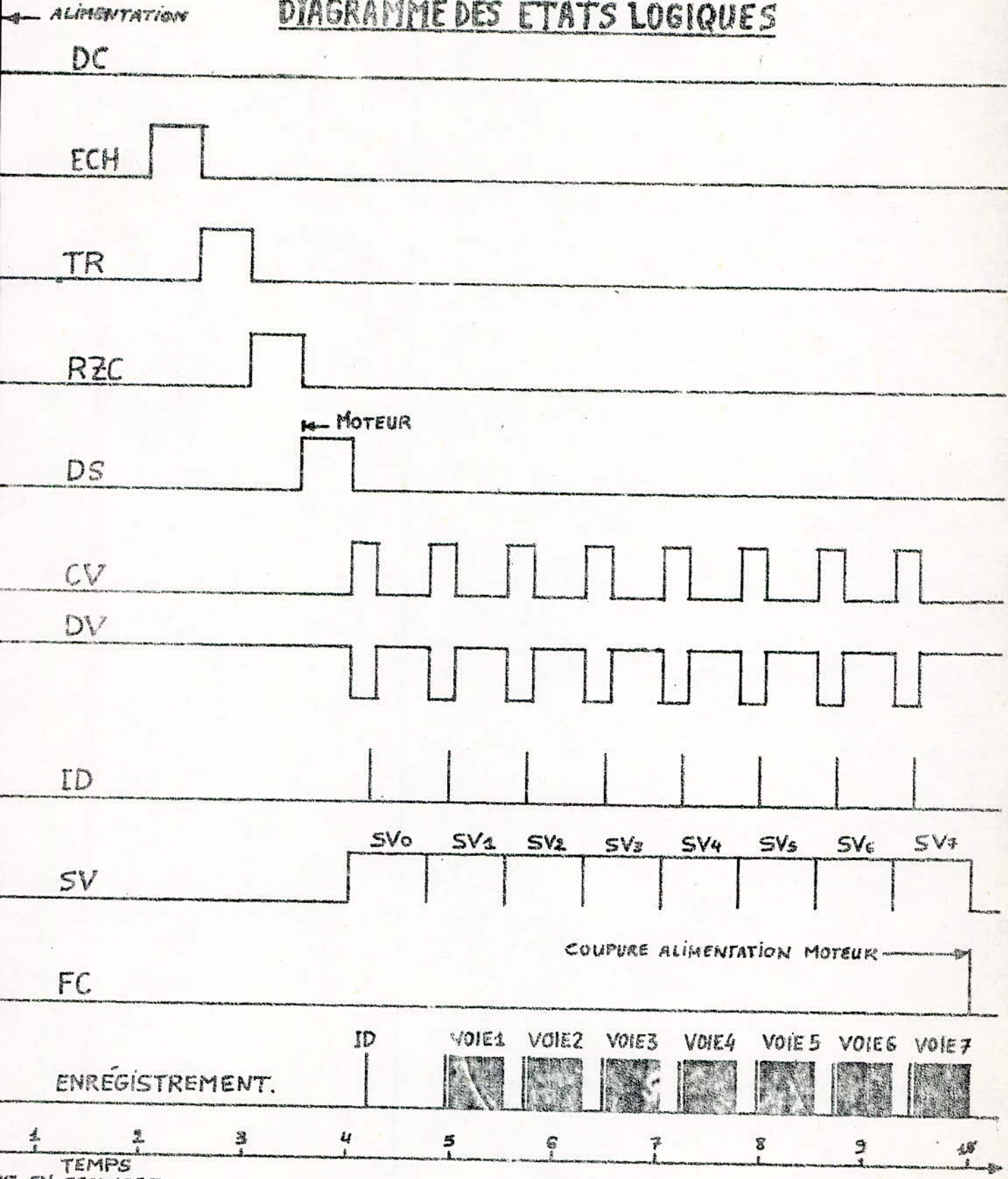
$F = 5$ khz

et, le codage se fait par périodes, c'est à dire à chaque nombre correspond un nombre équivalant de périodes.

Par simple calcul, on peut vérifier que le nombre maximum, que peut contenir une voie ne peut pas dépasser une valeur égale à : $\frac{0,5}{0,2.10^{-3}} = 2500.$

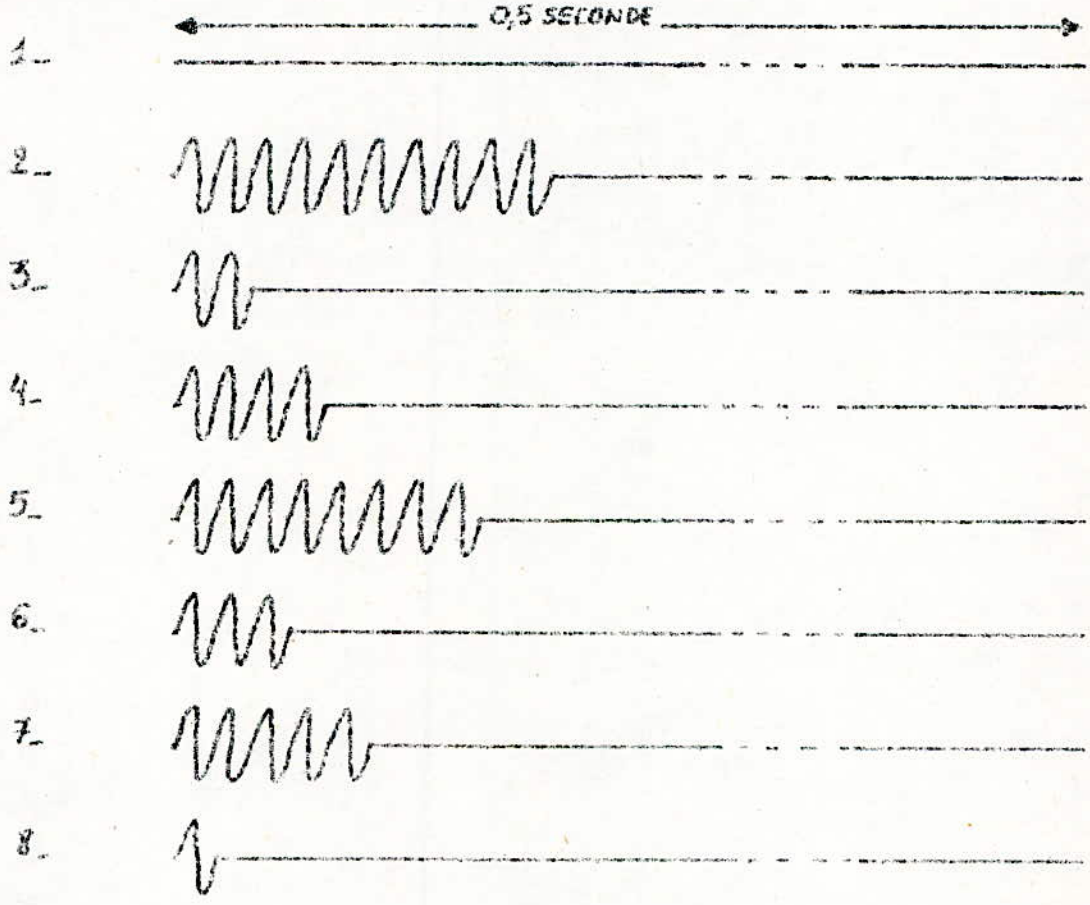
Cette valeur est largement suffisante quisque l'enregistrement maximal ne dépasse pas 2000.

DIAGRAMME DES ÉTATS LOGIQUES



_ FORME D'ENREGISTREMENT DE DONNEES

A l'aide de l'oscilloscope on a pu remarquer qu'il ya uniquement Variation en longueur des differents enregistrements. La figure suivante en donne un exemple.



Dans cet exemple, la commande d'enregistrement a été manuelle. il s'aura parfaite concordance si on compte le nombre de périodes pou chaque enregistrement et on le compare avec celui de l'imprimante qui a donnée simultanément la suite des nombres suivants:

0000
0010
0002
0004
0008
0003
0005
0001

III – Proposition d'un schéma synoptique :

Sur la base des résultats trouvés, on peut envisager différentes solutions et le schéma synoptique donné ci-après représente l'une d'entre elles.

III – 1 : Problème de décodage :

Comme on la vue, le code utilisé par l'enregistreur est relativement simple et pour passer au binaire il suffet d'utiliser un compteur, mais il yaura certaines contraintes, car le signal issu de la cassette est noyé presque completement dans le bruit, ce qui necessite une opération préliminaire de filtrage pour extraire le signal utile.

Le compteur doit utiliser des registres à 16 bits pour être conforme avec la représentation des nombres entiers dans la mémoire de l'Apple. l'horloge de 1,33 HZ sert à la remise à Zéro du compteur, chaque fois qu'une donnée entre en mémoire, et à l'incrementation du compteur d'adressés cette horloge peut être synchronisée par les impulsions d'identification présentes dans le signal.

III – 2 : Problème d'acquisition de données :

Sur le schéma on peut remarquer que le problème essentiel se pose au niveau du système d'acquisition de données, et cette dernière peut s'effectuer de deux manières différentes.

1 – Soit par l'Apple II plus.

2 – Soit par un système externe.

Dans le premier cas on éxploite directement le signal binaire issu du compteur et il yaura deux possibilités.

– En utilisant l'entrée IN de l'Apple, puisque un programme peut vérifier l'etat du circuit d'entrée cassette en inspectant l'adresse 49248 (équivalente à $ C060) si la valeur qui s'y trouvé est supérieure ou égale à 128, l'etat est "1", sinon l'etat est "0", bien que les programme Basic puissent lire l'etat du circuit d'entrée cassette, mais la vitesse de ces programmes est bien trop lente pour pouvoir attribuer un sens à une telle lecture.

on peut vérifier ça par l'execution du programme suivant :

```
10 FOR  I = 1 TO 1000.
20 X = PEEK (-15288).
30 PRINT X.
40 NEXT I.
```

et en injectant simultanement un signal de frome quelconque mais d'amplitude suffisante sur l'entrée IN de l'Apple.

- En utilisant l'entrée parallèle des connecteurs periphériques : le système Apple peut communiquer avec des périphériques grâce à des circuits d'interface spéciaux et des programmes de gestion des signaux de commandes particuliers d'entrée / Sortie, par l'intermédiaire des huit connecteurs de périphériques numérotés de Ø à 7.

Chaque connecteur comprend 50 broches permettant de disposer de signaux d'adresse (10 bits), de données (8 bits), de selection d'adresse (donc de périphériques), de generer des interruptions, de deconnecter des ROM de la carte mère etc...

Cette archotecture d'E/S fait que les péripheriques sont vus du microprocesseur comme des emplacements memoire auquels toutes les possibilités d'adressage de la mémoire sont applicables.

A chaque connecteur (Sauf le connecteur Ø) le système Apple fait correspondre 256 adresses de son espace mémoire, cette page de 256 positions contient un programme en ROM réalisant les fonctions d'entrée/Sortie spécifique du périphérique branché.

Cette ROM est implantée sur la carte d'interface, lui donnant ainsi une certaine " intelligence".

Le début de cette page a pour adresse $CSØØ, ou S et le numéro du connecteur ou a été branchée la carte d'interface.

De plus chaque connecteur dispose de 16 positions repartiés entre $CØ8Ø et $ CØFF (pour l'ensemble des 8), et chaque carte d'interface peut utiliser ces adresses à sa guise, pour réaliser des fonctions particulières d'entrées/Sortie sur le peripherique branché à cette carte.

Enfin un éspace de 2Ø48 adresses ($C8ØØ à SCFØØ) est utilisable par tout péripherique nécessitant un programme d'E/S plus élaboré qui sera figé en ROM sur la carte d'interface correspondante.

Chaque interface peut contenir cette ROM de 2K octets, mais une seule à la fois en "Ligne" pour le système Apple à un moment donné finalement il est réservé des mémoires en RAM pour chaque connecteur memoire localisés entre $4ØØ et $7FF. chacun dispose de 8 octets (Sauf le connecteur Ø) utilisables comme "Scratch - pad" (memoires temporaires), ces positions se trouvent à l'interieur de la page text ou GR, mais les fonctions graphiques n'interperent aucunement sur elles .

Dans le deusième cas, c'est à dire avec l'acquisition externe de données, la solution devient relativement compliquée comme la montre le syboptique de la figure.

Sur ce schéma, l'oxillateur F3 = 8ØØ HZ permet de génerer le signal de synchronosation qui se trouve au début de chaque enregistrement, il est d'une durée T = 2 x 10 secondes.

Après les 10 premières secondes, il ya un vide de 0,1 seconde correspondant au beep de l'enregistrement, ce dernier est entendu chaque fois qu'on a affaire à une lecture ou écriture de données par l'Apple.

Directement après le signal de synchronisation commence l'enregistrement de données qui doivent sortir en serie de la memoire RAM de 2K bytes, en commençant par le bit de poids le plus fort.

Le choix de cette capacité memoire est justifie par le nombre de données qu'on peut avoir sur 24 heures.

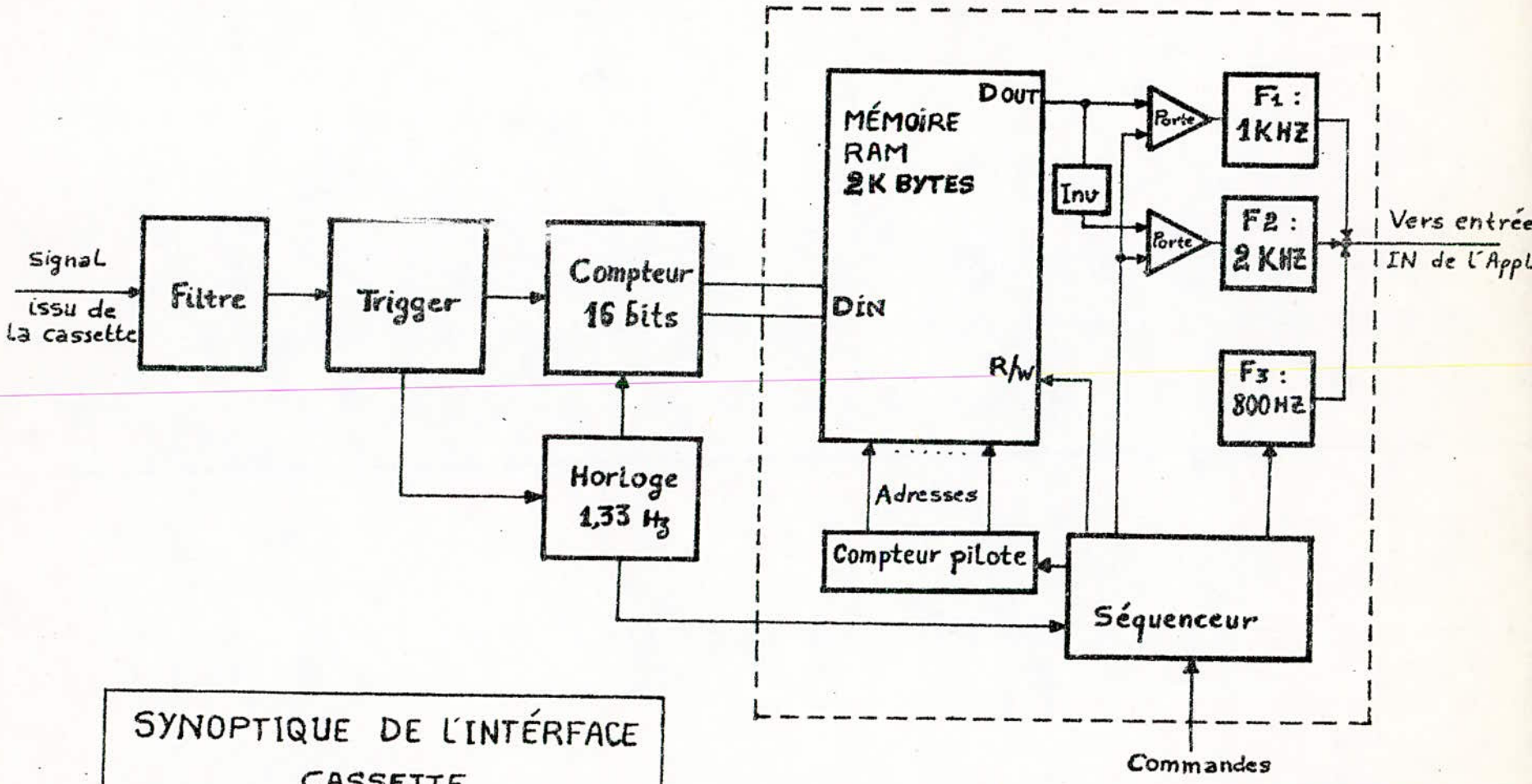
Selon la nature du bit (1 ou Ø) il yaura action de l'un ou l'autre des deux oxillateurs correspondant à F 1 = 1KHZ et F2 = 2 KHZ.
ces derniers delivrent à chaque fois une seule periode en commançant
Par l'alternence positive ou négative, et celle ci demeure inchangée le long de tout l'enregistrement.

Le Séquenceur permettera l'organisation des différentes opérations portant sur les instructions de lecture /ecriture en RAM, incrementation du compteur d'adresses entre autres.

Les différents oxillateurs doivent fournir des amplitudes relativement fixes telles que si on limite l'amplitude crête à crête des oxillations de frequences de 8ØØ HZ à 1 volt, celle de 1 KHZ doit être de 0,75 V, enfin celle de 2 KHZ sera alors 0,35 V.

Le signal obtenu peut etre enregistrée sur cassette audio, ou appliqué sur l'entrée IN de l'Apple pour avoir le traitement de données.

ACQUISITION DE DONNÉES



SYNOPTIQUE DE L'INTÉRFACE
CASSETTE

CONCLUSION

Comme on l'a vu la réalisation de l'interface cassette peut se faire de trois manières différentes, toute fois la partie transcodage reste commune, la seule différence réside au niveau du système d'acquisition de données qui peut etre faite soit par l'entrée serie ou parallèle de l'Apple II, ou bien par la conception d'un système spécial.

Apparemment on peut dire que les deux premières solutions sont plus aisées du fait que la bonne partie du travail sera faite par du Soft, mais la dernière reste particulièrement interessante, car elle permettera non seulement d'avoir une interface pour l'Apple, mais pour l'enregistreur lui même, ceci dans le cas où on veut enregistrer des données directement exploitable par le micro ordinateur, avec une economie appreciable de cassettes.

Enfin vu le temps qui nous a ete imparti, il sera impossible de faire la réalisation pratique, mais nous ésperons que d'autres étudiants à la lumière de nos resultats n'auront aucune difficulté à terminer ce travail, car outre son utilité pratique, la pluralité de solutions le rend plus attrayant du point de vue scientifique.

- BIBLIOGRAPHIE -

1°) - LA PRATIQUE DE L'APPLE II . VOLUME 1.

NICOLE BREAUD - POULIQUEN.

2°) - LA DECOUVERTE DE L'APPLE SOFT. TOME 1

D. SCHRAEN - F.LEVY

3°) - LE LANGAGE BASIC ET SES EXTENSIONS

J.P. LAMOITIER.

4°) - MANUEL DE REFERENCE DE L'APPLE

5°) - DOCUMENT DE L'ENREGISTREUR ED 7620.