

# Parole et micros

---

Page blanche

# Parole et micros

---

Synthèse vocale

Hervé Benoit  
Michel Weissgerber

**cedic/nathan**

6-10 boulevard Jourdan, 75014 Paris

**L'exploitation dans un but commercial des réalisations décrites dans ce livre devra être soumise à l'autorisation préalable des auteurs.**

---

**L'éditeur remercie la société RTC pour l'autorisation qui lui a été donnée de reproduire les schémas extraits de la documentation du MEA 8000 ainsi que la photo du système de codage OM8010.**

---

**Directeur d'édition: Serge Pouts-Lajus  
Couverture: Claire Baujat  
Maquette: Michèle Beaucamp**

---

**Editions Cedic/Nathan  
6-10 boulevard Jourdan, 75014 Paris  
Téléphone: (1) 45 65 06 06**

**Copyright © Cedic, 1985**

**Ce volume porte la référence  
ISBN 2-7124-1701-1**

*Toute reproduction, même partielle, de cet ouvrage est interdite. Une copie ou reproduction par quelque procédé que ce soit, photographie, photocopie, microfilm, bande magnétique, disque ou autre, constitue une contrefaçon passible des peines prévues par la loi du 11 mars 1957 sur la protection des droits d'auteur.*

# Sommaire

---

Introduction .....	9
<b>Chapitre 1. Principes de la synthèse de parole .....</b>	<b>13</b>
Pourquoi la synthèse de parole ? .....	13
La synthèse de parole à formants .....	15
<b>Chapitre 2. Le synthétiseur intégré MEA 8000 .....</b>	<b>21</b>
Particularités du MEA 8000 .....	21
Description du circuit .....	25
La partie "synthétiseur à formants" .....	26
Circuit d'interface et de traitement des données .....	27
Le circuit de sortie .....	29
Mise en œuvre du MEA 8000 .....	31
Registre tampon d'entrée .....	31
Registre d'état .....	31
Registre de commande .....	32
Modes de fonctionnement .....	32
Envoi des données vocales .....	34
<b>Chapitre 3. Logiciel de commande .....</b>	<b>37</b>
Principe .....	37
Application aux microprocesseurs 6800 et 6802 .....	42
Application au microprocesseur 6809 .....	45
Application aux microprocesseurs 6502 et 6510 .....	47
Application au microprocesseur Z-80 (ou Z-80A) .....	51
<b>Chapitre 4. Vocabulaire et phonétique .....</b>	<b>55</b>
Création du vocabulaire codé .....	55
Quel vocabulaire ? Sous quelle forme ? .....	57
Un peu de phonétique .....	58
La prosodie ? Ce n'est pas si simple .....	60
Intonation .....	60
Rythme .....	62
Les phonèmes du MEA 8000 .....	63
Et le chant ? .....	65

<b>Chapitre 5. Applications</b> .....	67
<b>Applications aux machines THOMSON</b> .....	67
Réalisation matérielle .....	69
Logiciel .....	73
<b>Utilisation sous ASSEMBLEUR</b> .....	74
Utilisation de l'IRQ .....	77
Les phonèmes sous ASSEMBLEUR .....	90
<b>Utilisation sous BASIC</b> .....	98
Le tout BASIC .....	98
Minimum de binaire .....	100
Totalité binaire .....	102
Les phonèmes sous BASIC .....	103
Votre premier jeu parlant .....	106
Y'a une pie dans le poirier ! .....	109
Utilisation sous FORTH .....	110
Utilisation sous LOGO .....	112
Utilisation sous LSE .....	113
<b>Applications sur d'autres machines</b> .....	115
Fonctionnement .....	115
<b>Applications sur APPLE II, COMMODORE et ORIC</b> .....	117
APPLE IIe ou II+ .....	117
COMODORE 64 .....	121
ORIC 1 et ATMOS .....	131
<b>Applications sur AMSTRAD, MSX, VG 5000 et MZ-700</b> .....	136
AMSTRAD CPC 464/664/6128 .....	136
Ordinateurs MSX .....	141
PHILIPS VG 5000 .....	145
SHARP MZ-700 .....	150

<b>Chapitre 6. PHONETRAM</b> .....	165
Introduction .....	165
Principe .....	166
Concaténation des phonèmes .....	166
Affinage phonique .....	166
Exploitation de PHONETRAM .....	168
Entrée des phonèmes .....	169
Les commandes .....	170
Logiciel .....	180
Le programme BASIC .....	180
Le programme source ASSEMBLEUR .....	182
Affichage d'écran .....	183

<b>Chapitre 7. NORMAPHON</b> .....	251
Utilisation de NORMAPHON .....	253
<b>Annexes</b> .....	257
Vocabulaire .....	257
Caractéristiques du MEA 8000 .....	307
Glossaire .....	311
Adresses utiles .....	314

Page blanche

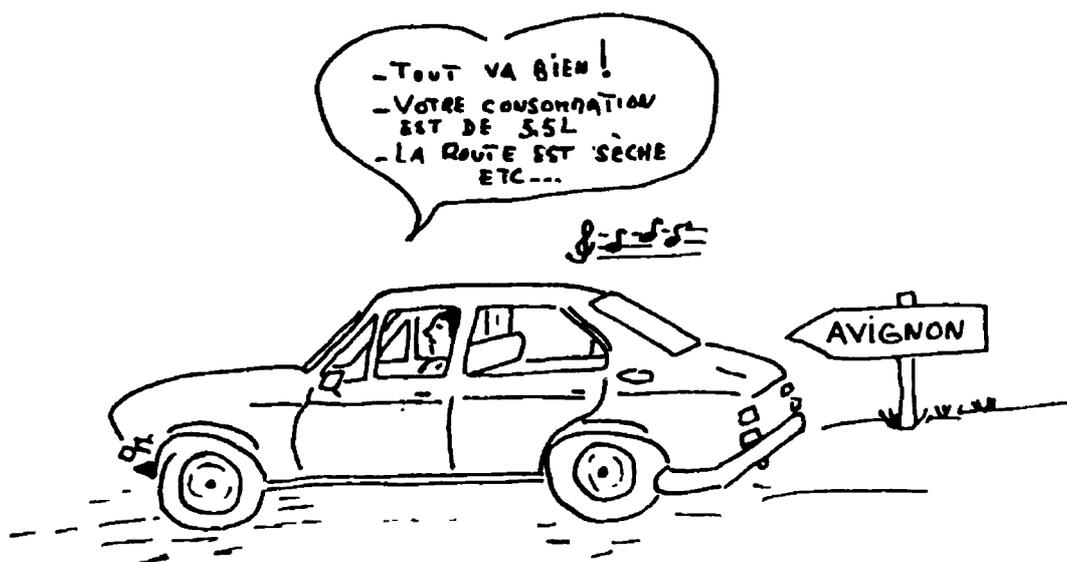
# Introduction

---

Faire parler une machine ou un objet est un rêve que l'homme a toujours caressé. La première réalisation sérieuse connue date de la fin du XVIIIème siècle: la machine du Hongrois Wolfgang von Kempelen, basée sur des résonateurs acoustiques et dont l'auteur jouait avec virtuosité. Au XIXème siècle, l'anglais J. Faber construisit une machine qu'il savait faire chanter et chuchoter. La première machine parlante électronique fut le Voder, présenté en 1939 aux U.S.A. par H. Dudley des *Bell Laboratories*.

Ces trois machines ne connurent pas d'application pratique, sans doute parce que leur utilisation nécessitait un apprentissage très compliqué. Dans les années 60-70, le Vocodeur à canaux, dont le premier prototype fut également réalisé par Dudley, fut l'un des premiers systèmes utilisés en pratique pour générer des messages vocaux synthétiques.

Ce n'est cependant que l'arrivée des synthétiseurs de parole intégrés, à la fin des années 70, qui a permis l'utilisation industrielle de la parole synthétique et l'a fait connaître au grand public. La première réalisation ayant connu un succès commercial très important est le célèbre "Speak and spell" de Texas Instruments sorti en 1978. Une version française, la "Dictée magique", a été introduite peu après. Quelques années plus tard, on a vu apparaître des automobiles dont les messages d'alerte étaient donnés sous forme vocale, en plus des habituels voyants lumineux du tableau de bord.



Malgré leur vocabulaire limité, ces applications n'auraient pas été viables économiquement sans l'utilisation d'un synthétiseur de parole, en raison de l'énorme mémoire ROM qu'elles auraient nécessité. En réduisant la taille mémoire nécessaire dans un rapport de l'ordre de cinquante à cent, les synthétiseurs de parole ont radicalement changé cet état de choses.

C'est pourquoi nous pouvons vous proposer, dans ce livre, de mettre en œuvre par vous-même, sur votre micro-ordinateur, un synthétiseur de parole intégré d'un type particulier, dit "à formants". Il s'agit du circuit MEA 8000 commercialisé en France par R.T.C.

Après une explication des principes généraux de la synthèse de parole, ce livre détaille plus particulièrement la synthèse par formants et son application dans le MEA 8000. Il décrit ensuite les principes de l'interfaçage avec un microprocesseur, en détaillant le cas des principaux processeurs utilisés dans les ordinateurs domestiques (6502/6510, Z-80/Z-80A, 6809), ainsi que la routine de base en langage machine activant le MEA 8000. Ceci permettra aux possesseurs d'une machine non traitée dans cet ouvrage, mais utilisant l'un des processeurs ci-dessus, de pouvoir aisément lui adapter le MEA 8000.

Ce livre montre également les différents types de vocabulaire utilisables, avec leurs avantages et leurs inconvénients selon l'application envisagée. Il traite plus particulièrement de l'utilisation d'un dictionnaire de phonèmes, en donnant quelques bases de phonétique permettant de les utiliser au mieux.

La partie "pratique" de cet ouvrage est consacrée à l'étude détaillée de l'application matérielle et logicielle à des machines actuellement très répandues en France, avec pour chacune d'entre elles la possibilité de création de phrases quelconques à partir de phonèmes. D'autres logiciels particuliers sont également décrits sur certaines machines, notamment sur les TO7, TO7/70, TO9 et MO5: le programme d'édition graphique "PHONETRAM" et "NORMAPHON" qui assure la normalisation des codes vocaux produits par PHONETRAM.

Le lecteur trouvera en annexe, sous forme de "listings", un vocabulaire relativement important, permettant de développer certaines applications avec une qualité supérieure aux phonèmes. Un glossaire des principaux termes employés en synthèse de parole termine l'ouvrage.



*Dessins : Robert Stroppiana*

Page blanche

# Chapitre 1

---

## Principes fondamentaux de la synthèse de parole

### Pourquoi la synthèse de parole ?

Avant de décrire l'adaptation du synthétiseur aux différents micro-ordinateurs, nous commencerons par une explication des principes généraux de la synthèse de la parole, suivie de la description de leur mise en application dans le synthétiseur à formants MEA 8000.

En premier lieu, on peut se demander pourquoi utiliser un synthétiseur de parole, alors que d'autres méthodes de reproduction a priori plus simples et plus directes existent. Deux raisons principales y conduisent :

- Si l'on veut disposer d'un accès aléatoire et immédiat à tous les éléments composant un vocabulaire donné, les moyens électromécaniques traditionnels (bandes ou disques magnétiques) sont inappropriés. Il n'est pour s'en convaincre que de regarder la complexité et le coût d'une

réalisation telle que l'horloge parlante, dont le vocabulaire est pourtant relativement limité. Il faut donc pouvoir stocker la parole dans un dispositif entièrement statique tel qu'une mémoire à semi-conducteurs, par exemple.

Ceci nécessite donc la "numérisation" du signal vocal ; pour obtenir une qualité comparable à celle du téléphone, il faut disposer d'une bande passante de l'ordre de 4 kHz, et d'une dynamique d'au moins 40 dB. C'est ce que l'on peut atteindre par une conversion analogique/numérique sur 10 à 12 bits avec échantillonnage à une fréquence de 8 kHz (théorème de Shannon). Le débit binaire résultant peut être réduit à 64 kb/s, valeur normalisée pour les transmissions téléphoniques de type M.I.C. (Modulation par Impulsions Codées).

- Le codage numérique pur (à 64 kb/s par exemple) est trop "gourmand" en mémoire : en effet, à ce débit, la mémoire vive d'un micro-ordinateur de 64 k-octets ne pourrait contenir que 8 secondes de parole, même sans aucun programme d'application !!

On a donc cherché depuis longtemps à réduire ce débit en exploitant la redondance très importante du signal vocal qui transporte un débit d'information utile de l'ordre de 70 bits par seconde, soit près de 1000 fois moins que le résultat de la numérisation du signal ; cette redondance a pour effet de rendre ce signal très résistant au bruit ambiant et aux distorsions qui peuvent affecter sa transmission.

Pour réduire le débit d'informations à transmettre, tous les synthétiseurs tirent parti du fait que la voix humaine peut seulement émettre certains sons compatibles avec le système physiologique de génération de la voix, dont nous allons étudier brièvement le fonctionnement.

Pratiquement, tous les synthétiseurs auront donc un schéma-bloc général dérivé de ce modèle physiologique ; nous étudierons plus particulièrement le modèle "à formants", dont le MEA 8000 fait partie.

# La synthèse de parole à formants

La figure 1 représente une coupe de l'appareil vocal humain et sa représentation schématique simplifiée.

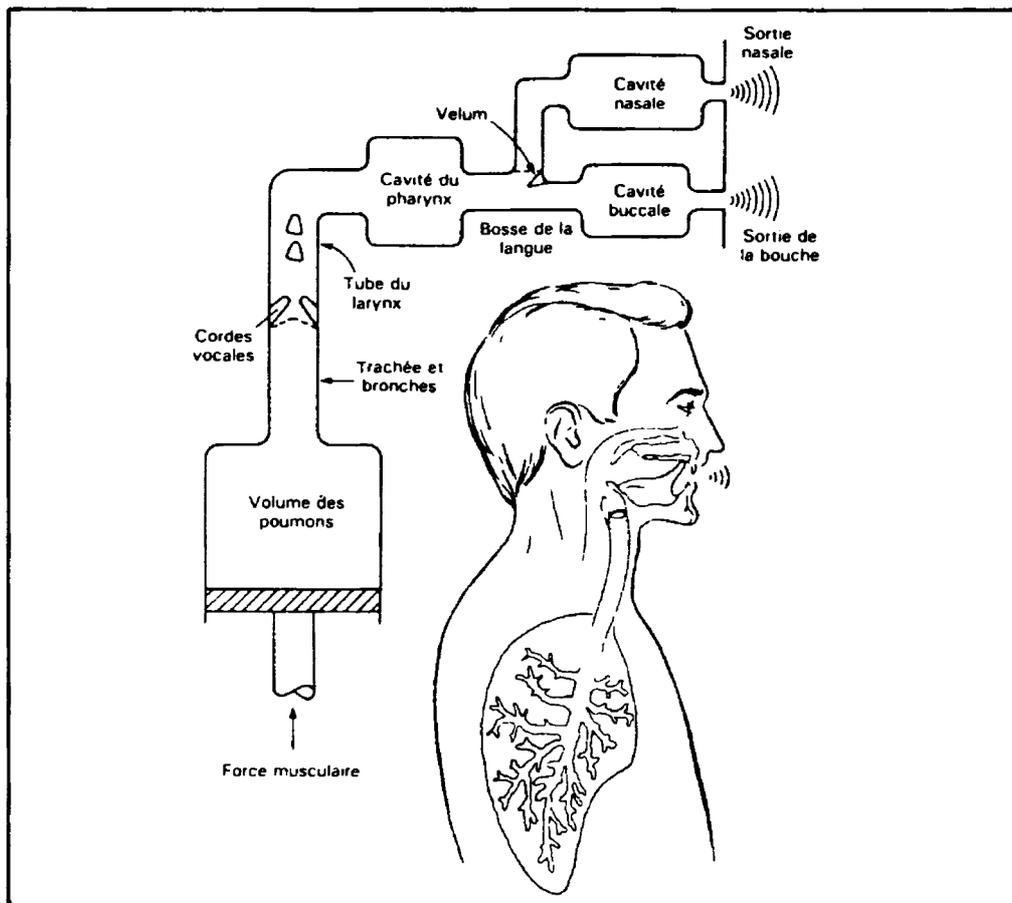


Figure 1 - Représentation simplifiée de l'appareil vocal

L'énergie qui servira à produire la voix est fournie sous forme de pression d'air par les poumons que l'on peut assimiler à une pompe. L'augmentation de la pression de l'air provoque l'ouverture des cordes vocales initialement closes. Il s'ensuit une brusque chute de pression, provoquant la fermeture des cordes vocales ; ceci entraîne une nouvelle augmentation de pression qui ouvrira de nouveau les cordes vocales, et ainsi de suite.

Ce mécanisme crée ainsi un train périodique d'impulsions de pression en dents de scie qui excite le conduit vocal situé au-dessus des cordes vocales. Les sons créés par ce processus sont dit "voisés" et correspondent à toutes les voyelles et à certaines consonnes dites sonores (b, d, g, l, m, n, r, v, z).

Le signal périodique ainsi généré est riche en harmoniques dont la décroissance est de l'ordre de 12 dB par octave ; sa fréquence fondamentale est appelée "pitch" dans la littérature anglo-saxonne et nous utiliserons par la suite ce terme ou celui de "fondamental" pour la désigner.

Il existe un autre mode de génération de sons vocaux dans lequel les cordes vocales sont toujours entr'ouvertes : de cette manière, l'air passe à travers elles de façon continue, sans les faire vibrer, en causant une turbulence dans le conduit vocal. Les sons produits de cette façon sont dits "non voisés" et ne correspondent qu'à des consonnes telles que les fricatives (ch, f, s...) et certaines plosives (k, p, t...). Ces consonnes sont dites sourdes.

La parole est constituée d'une suite continue de sons voisés ou non, dont l'amplitude et le pitch, qui caractérisent la "source" de signal, varient en permanence, et quelquefois assez rapidement. Le signal émis par la source décrite ci-dessus est ensuite "filtré" par le conduit vocal constitué des cavités pharyngienne, buccale et nasale. Cette dernière n'est généralement pas prise en compte dans les synthétiseurs intégrés, pour simplifier leur réalisation.

On peut alors, en synthèse vocale, assimiler le conduit vocal à un tube de diamètre constant, dont les résonances principales sont représentées sur la figure 2.

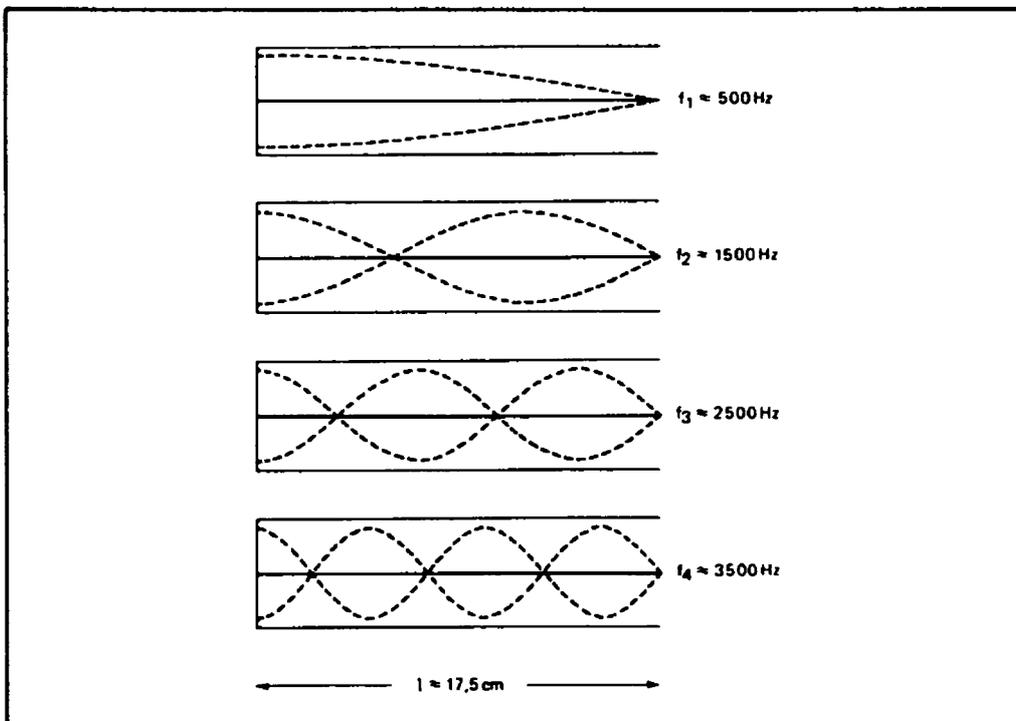


Figure 2 - Résonances d'un tube cylindrique

Ce tube est pratiquement fermé à l'extrémité constituée par les cordes vocales, et ouvert à l'autre par la bouche qui diffuse le son vers l'extérieur. La réponse en fréquence d'un tel tube est caractérisée par un nombre de résonances équidistantes dont les fréquences sont données par la relation :

$$f(N) = 340 (2N - 1) / 4L$$

où  $N = 1, 2, 3, 4 \dots$  et  $L =$  longueur du conduit vocal (en mètres).

Ces fréquences de résonance sont appelées "formants" du conduit vocal. A l'intérieur de la bande de 0 à 4000 Hz, on trouve en général quatre formants pour une voix masculine et trois pour une voix féminine, en raison de la longueur plus réduite du conduit vocal chez la femme. La figure 3 illustre la position relative des formants dans les deux cas.

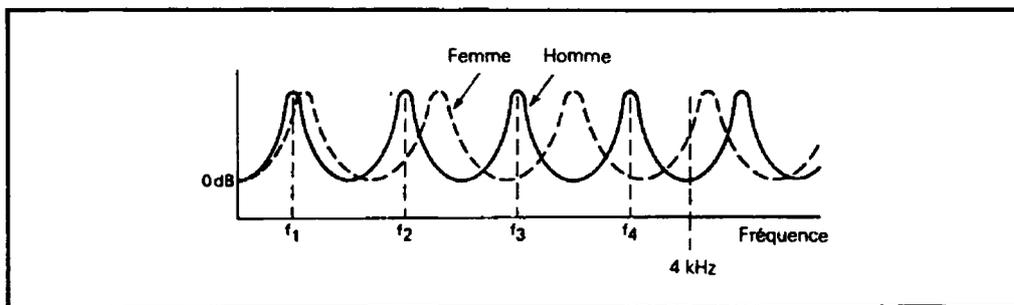


Figure 3 - Position des formants

Au cours de la parole, la forme du conduit vocal varie constamment : par exemple, pour le son "e", la cavité pharyngienne est grande alors que la cavité buccale est petite, ce qui a pour effet d'accroître la fréquence du formant n° 2. Lorsque l'on prononce un "a", la situation est inverse, ce qui réduit la séparation entre les formants n° 1 et 2. La figure 4 illustre ces deux situations.

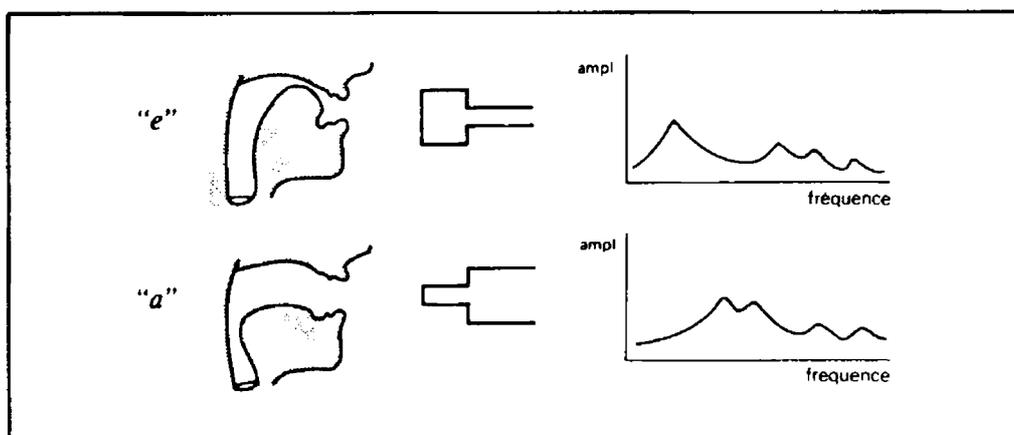


Figure 4 - Spectre des formants pour le "a" et le "e"

Chaque formant est en outre caractérisé par sa bande passante qui correspond à une résonance plus ou moins prononcée : plus la bande est étroite, plus la résonance est importante.

Ce sont les formants qui caractérisent réellement le son émis à un instant donné, et les trois premiers sont les plus importants pour l'intelligibilité du message.

Le fondamental (pitch), ainsi que, dans une certaine mesure, le voisement et l'amplitude, peuvent être considérés comme des informations "secondaires" pour la signification du message, essentiellement déterminée par l'évolution du conduit vocal et donc des formants. Les variations du pitch sont le facteur principal de l'intonation. Le non-voisement (en dehors des consonnes spécifiques) caractérise le chuchotement et l'accent tonique est déterminé par les variations instantanées de l'amplitude, dont de très rapides variations caractérisent également certains sons tels que les plosives.

De toutes les considérations précédentes, on peut déduire le schéma-bloc général d'un synthétiseur à formants (figure 5), qui n'est que la réalisation électronique simplifiée du modèle de départ.

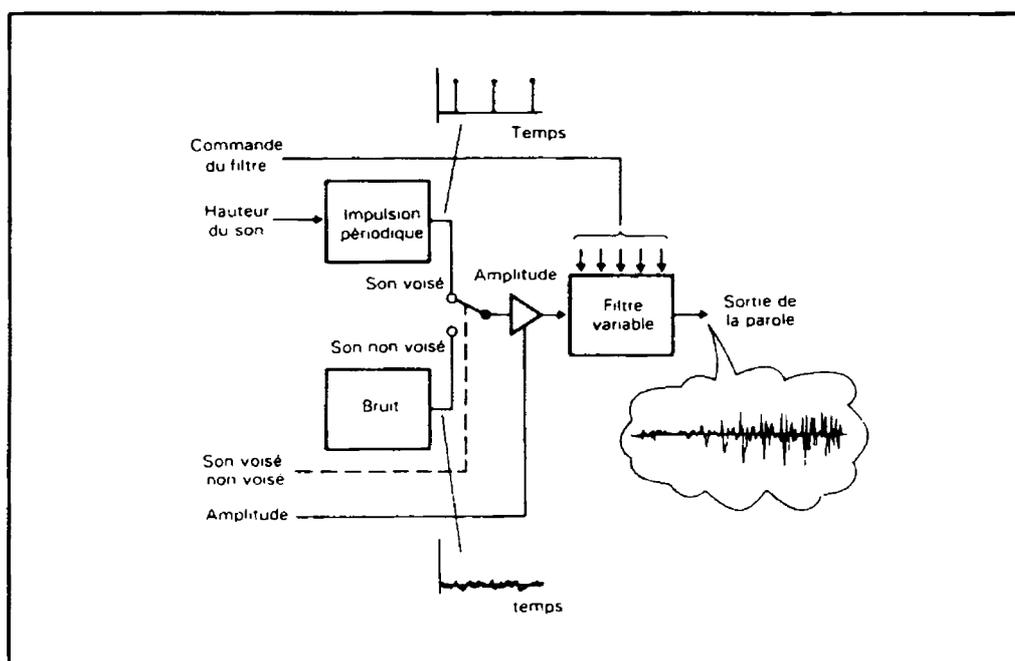


Figure 5 - Schéma-bloc d'un synthétiseur de parole

Le synthétiseur se compose donc d'une source de signal périodique, de fréquence programmable et égale au pitch de sons voisés. Pour la génération des sons non voisés, il dispose également d'une source de bruit blanc de 0 à 4000 Hz. Un commutateur permet de sélectionner l'une ou

l'autre source, et une commande de gain permet de déterminer l'amplitude du signal d'excitation. Enfin, un réseau de filtres programmables permet la simulation du conduit vocal. Chacun de ces filtres est programmable en fréquence et en bande passante et l'ensemble permet de modeler le spectre du signal conformément à la parole originale.

En résumé, le son émis par le synthétiseur à un instant donné est déterminé par l'ensemble des paramètres suivants :

— Fréquence fondamentale (pitch)	}	caractérisent la source d'excitation
— Son voisé ou non voisé		
— Amplitude (ou énergie)		
— Fréquence centrale des formants	}	caractérisent le conduit vocal
— Bande passante des formants		

L'obtention d'une reproduction fidèle de la parole à synthétiser nécessite une actualisation périodique de tous ces paramètres, considérés comme fixes dans une "fenêtre" dont la durée détermine la durée de trame sonore. Le conduit vocal humain étant un système "mécanique", ses variations sont relativement lentes, et l'actualisation des paramètres du synthétiseur peut être réalisée de manière satisfaisante avec une périodicité de l'ordre de 10 à 20 millisecondes.

Cette période (durée de trame) doit être suffisamment longue de façon à contenir assez d'échantillons pour le calcul des paramètres lors de l'analyse du signal vocal, mais pas trop afin de pouvoir reproduire de manière satisfaisante ses variations les plus rapides.

Afin de ne pas provoquer de brusques changements lors du passage d'une trame à la suivante, une interpolation linéaire est souvent effectuée sur les paramètres entre deux trames consécutives. L'ensemble des fonctions du synthétiseur est entièrement réalisé en technique numérique, le signal de sortie étant restitué au moyen d'un convertisseur numérique/analogique (C.A.N.). Les messages à reproduire sont stockés dans une mémoire ROM ou RAM selon l'application.

Page blanche

# Chapitre 2

---

## Le synthétiseur de parole intégré MEA 8000

Ce chapitre est consacré à la description détaillée du MEA 8000, à son interfaçage avec le processeur de contrôle, ainsi qu'aux principes généraux de sa programmation.

### Particularités du MEA 8000

---

Le MEA 8000 est un synthétiseur à formants et correspond de ce fait aux caractéristiques générales de ceux-ci, expliquées au chapitre précédent.

Du point de vue physique, il s'agit d'un circuit périphérique de microprocesseur ou microcontrôleur à bus de données 8 bits ; ses principales caractéristiques sont résumées à l'annexe 1, où l'on trouvera également son brochage et le schéma de principe général d'application. Les particularités essentielles du MEA 8000 sont les suivantes :

- **Quatre formants, chacun réalisé sous forme d'un filtre numérique du second ordre à trois multiplicateurs (voir figure 6).**

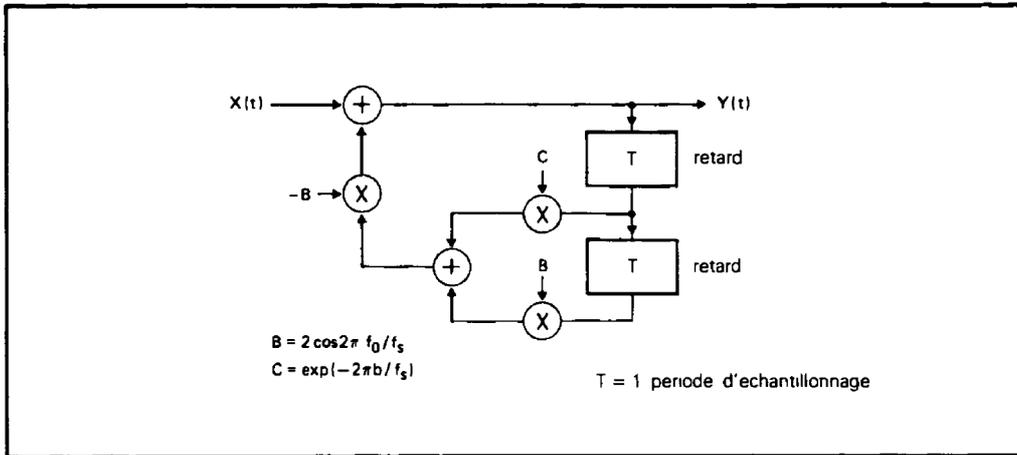


Figure 6 - Filtre numérique du 2<sup>e</sup> ordre

La mise en cascade de ces quatre résonateurs reproduit le modèle du conduit vocal; elle est représentée par la figure 7.

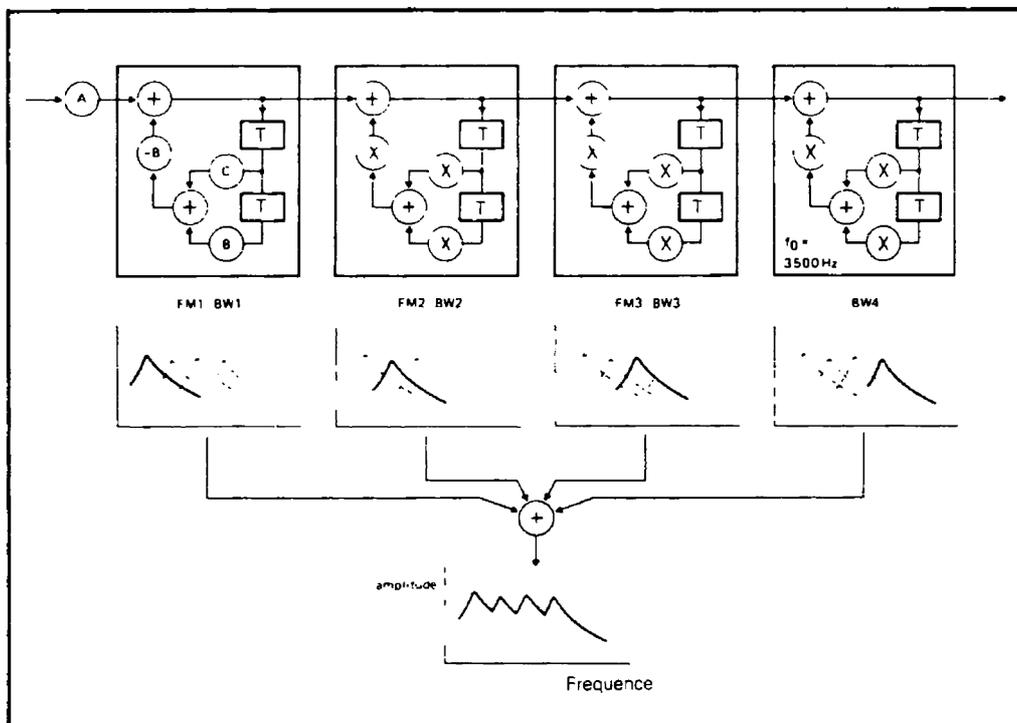


Figure 7 - Filtre complet simulant le conduit vocal

Dans le cas du MEA 8000, les trois premiers formants sont programmables en fréquence centrale (FM1, FM2, FM3) et en bande passante (BW1, BW2, BW3). Le quatrième formant n'est programmable qu'en bande passante (BW4), sa fréquence centrale étant fixée à 3500 Hz.

Outre les sons purement vocaux, le synthétiseur est capable de reproduire d'autres types de sons tels que des bruits ou de la musique.

Une méthode simple pour générer des sons musicaux consiste à utiliser comme générateur la source de bruit et à accorder les formants 1 et 2 sur la même fréquence (celle de la note à générer) pour créer un filtre très sélectif; il est également possible d'utiliser le générateur périodique en accordant les formants sur des harmoniques de celui-ci pour créer le timbre recherché.

L'enveloppe du son (attaque et décroissance) peut être obtenue en créant une courbe d'amplitude appropriée. Le nombre et la durée des trames utilisées déterminera celle de la note.

Il est toutefois certain que les performances du MEA 8000 en ce domaine n'égalent pas celles des synthétiseurs musicaux spécialisés qui offrent une gamme d'octaves plus étendue et plusieurs voix simultanées.

La génération de bruits spéciaux (explosion, coup de feu, train, etc.) est possible en utilisant le générateur de bruit et en jouant sur les formants, l'amplitude et la répétition des sons ; quelques sons de ce type sont fournis à titre d'exemple en fin d'ouvrage.

- **Durée de trame programmable** ; dans le cas du MEA 8000, la durée de trame est programmable en quatre valeurs de 8, 16, 32, 64 ms au moyen de 2 bits du code de trame qui en compte 32 en tout. Ceci permet de tirer parti de certains passages où les paramètres varient lentement ou pas du tout en utilisant une durée de trame de 32 ou 64 ms.

A l'inverse, l'utilisation des trames de 16, et, plus rarement, de 8 ms permet une reproduction fidèle des passages tels que les plosives (sons p, t, k par exemple), caractérisées par une évolution rapide des paramètres vocaux. L'utilisation judicieuse de la durée de trame lors du codage permet d'obtenir une parole de qualité avec un débit de l'ordre de 1000 bits/seconde.

- **Convertisseur numérique/analogique de sortie intégré** ; ce convertisseur incorpore un circuit de suréchantillonnage qui intercale sept échantillons interpolés entre deux échantillons à 8 kHz fournis par la partie "synthèse" du circuit. Ceci a pour effet de fournir, en sortie du circuit, un signal dont la fréquence d'échantillonnage est de 64 kHz, très supérieure à la fréquence limite d'audition, et donc beaucoup plus facile à éliminer du signal de sortie (un simple filtre RC passe bas suffit dans la majorité des cas).

Outre les fonctions liées à la synthèse proprement dite, le circuit incorpore un générateur d'horloge à quartz (4,00 MHz), une interface microprocesseur et un circuit de sortie analogique. La figure 8 représente le synoptique simplifié du MEA 8000.

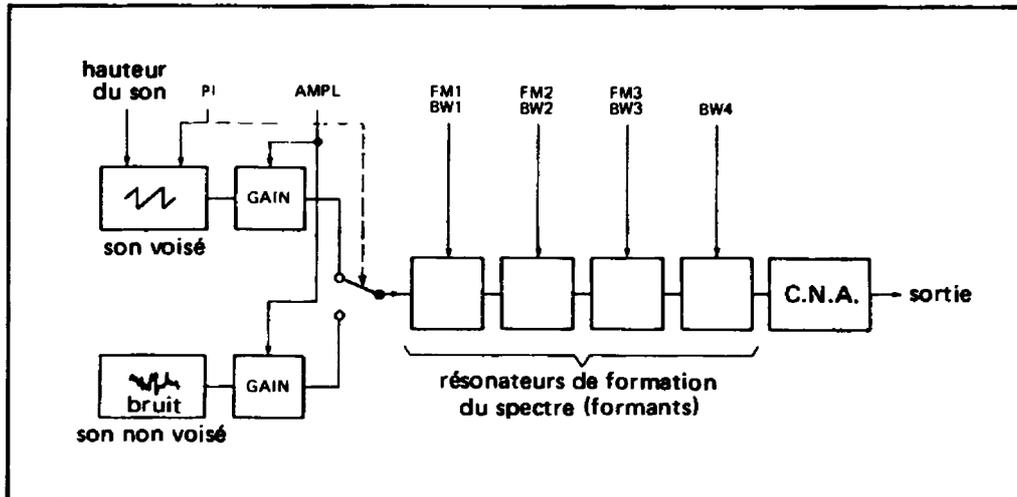


Figure 8 - Synoptique fonctionnel du synthétiseur MEA 8000

# Description du circuit

Le synoptique détaillé du MEA 8000 est représenté par la figure 9 sur laquelle on peut distinguer trois grands blocs fonctionnels.

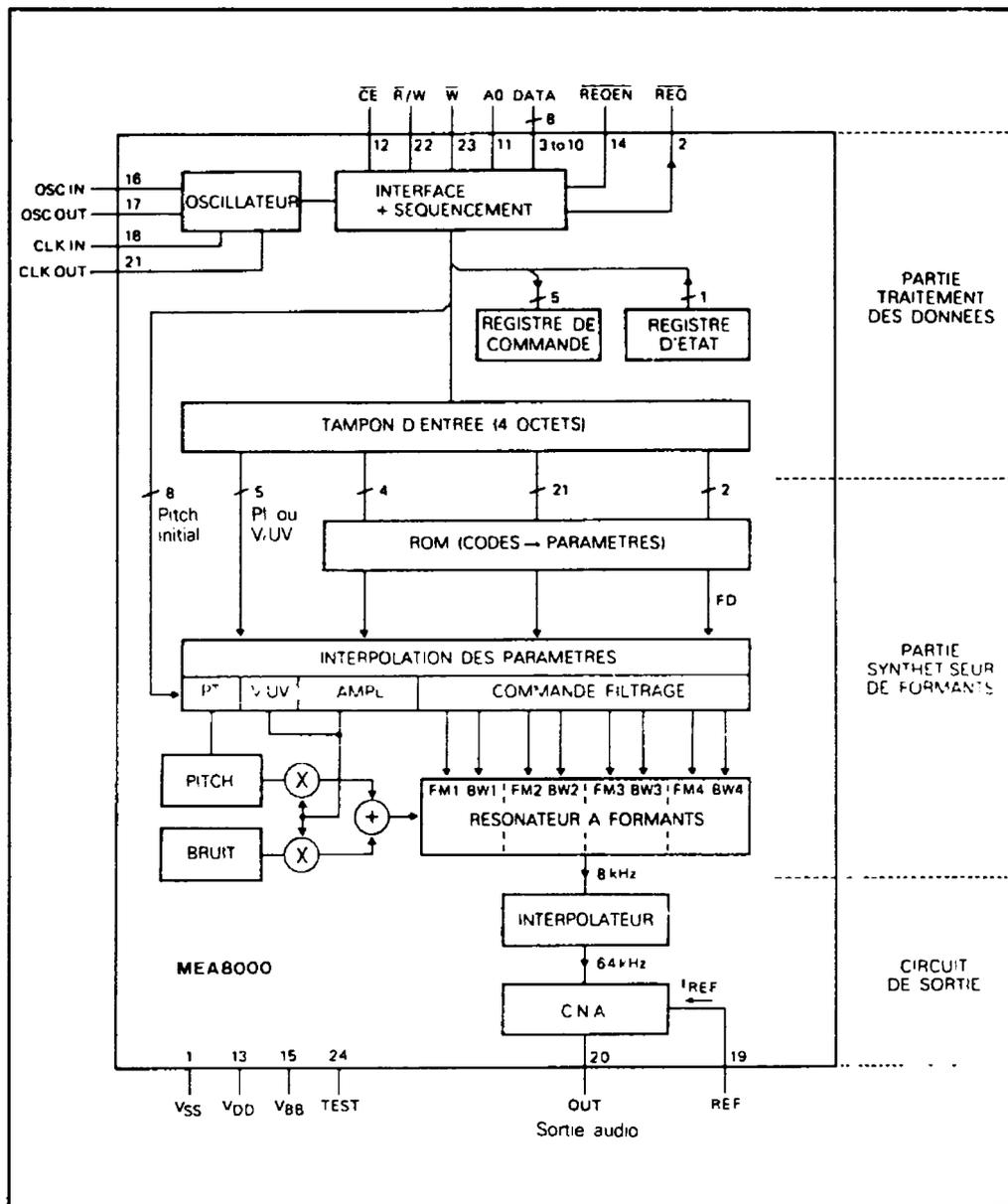


Figure 9 - Synoptique complet du synthétiseur MEA 8000

## **La partie “synthétiseur à formants”**

Elle est située au centre du synoptique. La source périodique, la source de bruit et les quatre résonateurs à formants décrits au chapitre précédent consistent en fait en un ensemble de multiplicateurs et d'additionneurs à 16 bits qui calcule les échantillons vocaux au rythme de 8 kHz. Les échantillons de 16 bits résultants sont tronqués à 11 bits avant application au circuit de sortie.

Le synthétiseur est contrôlé par 11 paramètres représentant la hauteur (pitch) et sa variation pour les sons voisés, ou la source de bruit pour les sons non voisés, l'amplitude ainsi que les quatre fréquences centrales et les quatre bandes passantes des formants.

Chaque paramètre est défini sur 14 bits, d'où un total de 154 ( $11 \times 14$ ) bits pour commander l'ensemble du synthétiseur sur une trame. Pour réduire le nombre de bits nécessaires à la définition d'une trame, chaque paramètre est codé au moyen d'une quantification appropriée; ceci permet de réduire à 27 le nombre des bits nécessaires au codage des paramètres du conduit vocal.

La logique d'interpolation de paramètres calcule la différence entre deux valeurs consécutives et réalise une interpolation linéaire entre elles pour lisser les transitions; l'intervalle d'interpolation est obtenu par décodage des deux bits FD qui déterminent la durée de trame. Ce sont les sorties de cette logique qui commandent toutes les fonctions de synthèse (pitch, voisé/non voisé, amplitude, formants).

D'autre part, le codage du pitch est effectué en valeur relative (variation en Hz sur un intervalle de 8 ms); ceci permet de réduire de 8 à 5 le nombre de bits nécessaire au codage de ce paramètre, autorisant une variation de  $-15$  à  $+15$  Hz/8 ms soit près de  $\pm 2$  kHz/s, ce qui est suffisant. La valeur particulière 16 sélectionne le générateur de bruit pour les sons non voisés. Une trame est donc entièrement définie au moyen de 32 bits.

Cette méthode de codage implique de fournir la valeur initiale du pitch en début d'expression, sous forme d'un premier octet, avant l'envoi des données relatives à la première trame à prononcer. Chaque trame est ensuite définie par quatre octets.

## Circuit d'interface et de traitement des données

C'est la partie supérieure du synoptique de la figure 9. Comme nous l'avons vu au chapitre précédent, les paramètres du synthétiseur doivent être actualisés de façon périodique. Ces données sont fournies au circuit par un bus parallèle à 8 bits et envoyées dans un registre tampon de 32 bits lorsque  $A0 = 0$ . La figure 10 indique l'affectation des différents bits du code de trame.

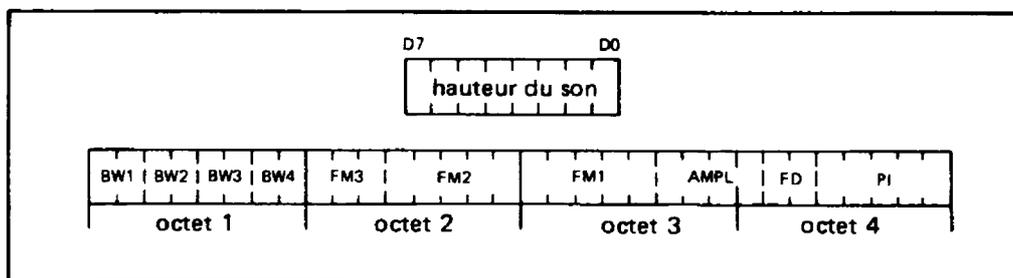


Figure 10 (a) - Les différents bits du code de trame

Code	Bits	Paramètre
Hauteur	8	Valeur initiale de la hauteur du son
FD	2	Durée de la trame de parole
PI	5	Incrément (vitesse de variation) de la hauteur du son ou sélection du générateur de bruit
AMPL	4	Amplitude
FM1	5	Fréquence du 1er formant
FM2	5	Fréquence du 2ème formant
FM3	3	Fréquence du 3ème formant
FM4	0	Fréquence du 4ème formant (fixe)
BW1	2	Largeur de bande du 1er formant
BW2	2	Largeur de bande du 2ème formant
BW3	2	Largeur de bande du 3ème formant
BW4	2	Largeur de bande du 4ème formant

Figure 10 (b) - et leur signification

La figure 11 représente les valeurs correspondantes des paramètres en fonction du code de trame.

Code binaire	FD (ms)	Hauteur (Hz)	PI Hz/8 ms	Ampl.	FM1 (Hz)	FM2 (Hz)	FM3 (Hz)	BW (Hz)
0	8	0	0	0	150	440	1179	726
1	16	2	1	0.008	162	466	1337	309
2	32	4	2	0.011	174	494	1528	125
3	64	6	3	0.016	188	523	1761	50
4		8	4	0.022	202	554	2047	
5		10	5	0.031	217	587	2400	
6		12	6	0.044	233	622	2842	
7		14	7	0.062	250	659	3400	
8		16	8	0.088	267	698		
9		18	9	0.125	286	740		
10		20	10	0.177	305	784		
11		22	11	0.250	325	830		
12		24	12	0.354	346	880		
13		26	13	0.500	368	932		
14		28	14	0.707	391	988		
15		30	15	1.00	415	1047		
16		32	bruit		440	1110		
17		34	-15		466	1179		
18		36	-14		494	1254		
19		38	-13		523	1337		
20		40	-12		554	1428		
21		42	-11		587	1528		
22		44	-10		622	1639		
23		46	-9		659	1761		
24		48	-8		698	1897		
25		50	-7		740	2047		
26		52	-6		784	2214		
27		54	-5		830	2400		
28		56	-5		880	2609		
29		58	-3		932	2842		
30		60	-2		988	3105		
31		62	-1		1047	3400		
.		.	.					
.		.	.					
255		510						

La fréquence de FM4 est fixée à 3 500 Hz. La colonne BW (largeur de bande) s'applique aux quatre filtres. Pour obtenir les valeurs exactes, il convient de multiplier par 1 0244 les chiffres portes dans les colonnes hauteur du son et PI (incrément de la hauteur du son).

Figure 11 - Valeur des différents paramètres en fonction du code de trame

Le circuit possède également un registre de commande vers lequel les données sont aiguillées lorsque  $A0 = 1$ . Ceci permet de déterminer le mode de fonctionnement du synthétiseur par l'envoi d'une commande "STOP" (voir "Mise en oeuvre du MEA 8000"). Un registre d'état à 1 bit (REQ) permet de connaître l'état du synthétiseur pour l'envoi des données vocales en mode actif. Des signaux de contrôle  $\overline{CE}$ ,  $\overline{W}$ ,  $\overline{R/W}$  permettent un interfacement simple avec le processeur de commande.

## Le circuit de sortie

C'est la partie inférieure du synoptique de la figure 9. Après un arrondi à 8 bits, les échantillons à 8 kHz provenant de la sortie des filtres numériques sont appliqués à un interpolateur linéaire qui intercale 7 échantillons interpolés entre deux échantillons consécutifs à 8 kHz ; ceci accroît donc la fréquence effective d'échantillonnage en sortie à 64 kHz et permet un post-filtrage analogique très simple.

Ces échantillons de 8 bits à 64 kHz attaquent ensuite le convertisseur analogique/numérique de la figure 12, dont le courant de référence  $I$  est défini par la résistance externe connectée à la borne REF (borne 19). Les 4 bits de poids fort déterminent le temps de conduction d'un générateur de courant  $16 \times I$ , et les 4 bits de poids faible celui d'un générateur de courant  $I$ . Ceci permet donc un total de  $16 \times 16 = 256$  combinaisons nécessaires à la conversion 8 bits.

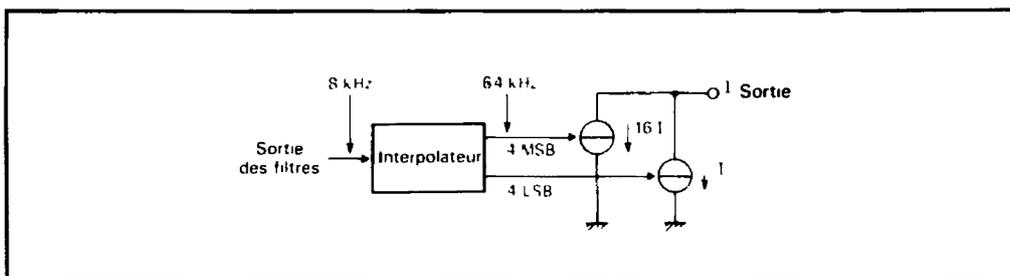


Figure 12 - Interpolateur et C.N.A. de sortie

L'amplificateur de puissance externe peut être réalisé de façon très simple au moyen du schéma de la figure 13 pour attaquer une ligne, un casque ou un petit haut-parleur d'impédance 50 à 100 ohms ; il assure également le filtrage de sortie.

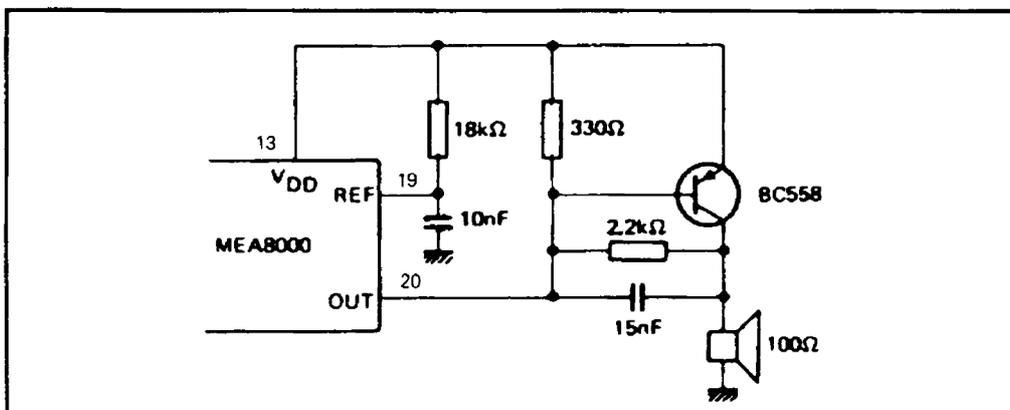


Figure 13 - Un amplificateur audio ultra-simple

La figure 14 représente un schéma utilisant un amplificateur intégré type TDA 1011 pouvant fournir jusqu'à 4 watts à H.P. de 4 ohms. Tout autre amplificateur peut convenir et l'entrée Péritel d'un téléviseur, en particulier, est une solution tout indiquée lorsqu'on utilise ce type de moniteur pour la visualisation.

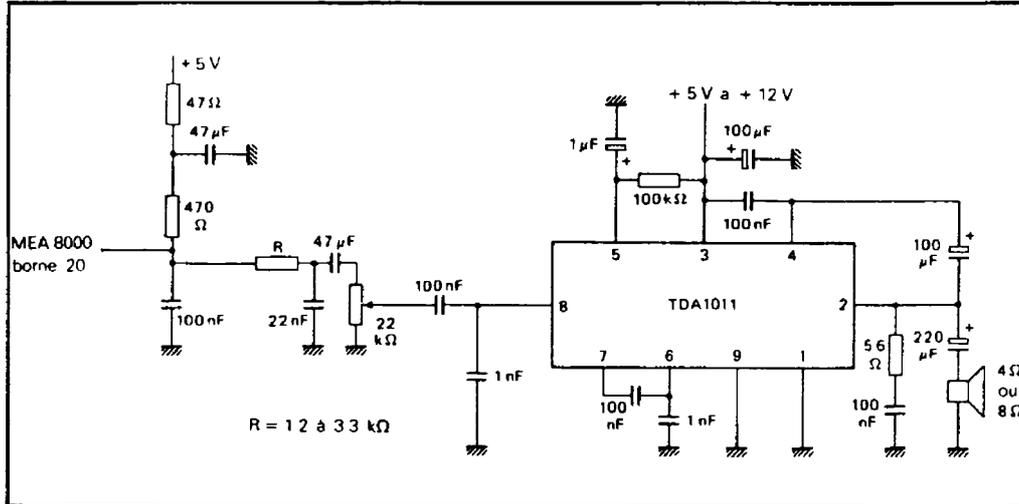


Figure 14 - Un autre amplificateur, plus puissant

# Mise en œuvre du MEA 8000

---

Le MEA 8000 est un circuit périphérique de microprocesseur ou microcontrôleur et peut être adressé comme un circuit d'entrée/sortie ou une mémoire, grâce aux signaux de contrôle  $\overline{CE}$ ,  $\overline{W}$ ,  $\overline{R/W}$ , à l'entrée d'adresse A0 et au signal  $\overline{REQ}$ .

Le registre de commande et le tampon d'entrée sont à écriture seule, alors que le registre d'état est à lecture seule. La table de vérité de la figure 15 indique la manière d'y accéder. Les caractéristiques temporelles des signaux sont données en annexe au paragraphe "Caractéristiques principales du MEA 8000".

$\overline{CE}$	$\overline{W}$	$\overline{R/W}$	A0	Opération
0	0	1	0	ECRITURE DE DONNEES
0	0	1	1	ECRITURE DE COMMANDE
0	X	0	X	LECTURE D'ETAT
0	1	1	X	} BUS DE DONNEES TROIS ETATS
1	X	X	X	

Figure 15 - Accès aux registres du MEA 8000

## Registre tampon d'entrée

Ce registre de 32 bits (4 octets) contient les codes d'une trame de parole et peut être adressé lorsque A0 = 0.

## Registre d'état

Ce registre n'est constitué en fait que d'un seul bit appelé REQ. Il peut être lu sur le bit 7 du bus de données lorsque  $\overline{R/W}$  est à l'état bas et apparaît également (inversé) sur la borne  $\overline{REQ}$  si elle a été validée. Lorsque le bit REQ est à "0", le MEA 8000 est occupé et ne peut recevoir de nouvelles données. Il indique qu'il demande ses données par la mise à "1" du bit REQ. La borne  $\overline{REQ}$  (active bas) peut être validée par mise à la masse de la borne 14 ( $\overline{REQEN}$ ) ou par logiciel en mettant "1" le bit ROE du registre de commande avec  $\overline{REQEN}$  à l'état haut. Ceci permet donc l'utilisation du MEA 8000 en interruption ou en scrutation.

## Registre de commande

Le MEA 8000 dispose d'un registre de commande de 5 bits dans lequel un mot de commande peut être écrit lorsque  $A0 = 1$ . Le tableau ci-dessous indique le rôle des différents bits de ce registre :

D4	D3	D2	D1	D0
STOP	CONT enable	CONT	ROE enable	ROE
0 = sans effet	0	X	0	X
1 = STOP	1	0 = ARRET LENT	1	0 = $\overline{REQ}$ validée
	1	1 = CONTINU	1	1 = $\overline{REQ}$ inhibée

D5, D6, D7 inutilisés. ROE = valid. de la sortie  $\overline{REQ}$ . X = indifférent.

**STOP (D4)** : si ce bit est à l'état "1", le synthétiseur s'arrêtera dès la réception du mot de commande (initialisation).

**CONT (D2)** : ce bit ne peut être modifié que si le bit CONT enable (D3) est à l'état "1"; si CONT est à l'état "1", le synthétiseur répètera indéfiniment la dernière trame en cas d'interruption dans la réception des codes.

**ROE (D0)** : ne peut être modifié que si le bit ROE enable (D1) est à l'état "1". ROE = "1" entraîne l'apparition du bit REQ (inversé) sur la borne  $\overline{REQ}$ . Le même résultat est obtenu en mettant la borne  $\overline{REQEN}$  à "0".

A la mise sous tension, les bits CONT et ROE sont mis à "0", ce qui équivaut à l'envoi du mot de commande 0001 1010 (Hex 1A) ; c'est donc la lecture de REQ sur le bus de données (D7) et la procédure "arrêt lent" qui sont sélectionnés par défaut.

## Modes de fonctionnement

Le MEA 8000 a deux modes de fonctionnement :

- **Mode INACTIF** : caractérisé par une sortie silencieuse et le bit d'état REQ à "1". Ce mode est provoqué par la mise sous tension ou l'envoi d'une commande STOP qui provoque un arrêt immédiat du synthétiseur, ou bien à la fin d'une procédure d'arrêt lent (voir ci-dessous). Dans ce cas, la trame est répétée avec une amplitude réduite avant l'arrêt du synthétiseur.
- **Mode ACTIF** : ce mode est activé par la réception de l'octet de pitch initial, et le circuit prononce ensuite les sons correspondant aux trames reçues. Pour un fonctionnement normal, les 4 octets de la trame suivante doivent être reçus avant la fin de la trame en cours de prononciation. Si tel

n'est pas le cas, le circuit entame l'une des deux procédures suivantes selon l'état du bit CONT :

— ARRET LENT si CONT = "0" (répétition de la trame avec amplitude réduite, puis retour au mode INACTIF).

— CONTINU si CONT = "1" : dans ce cas, le circuit répète indéfiniment la dernière trame reçue jusqu'à la réception d'une nouvelle trame ou d'une commande STOP.

La figure 16 montre l'évolution du signal de sortie en cas d'interruption des codes selon que l'une ou l'autre procédure a été choisie.

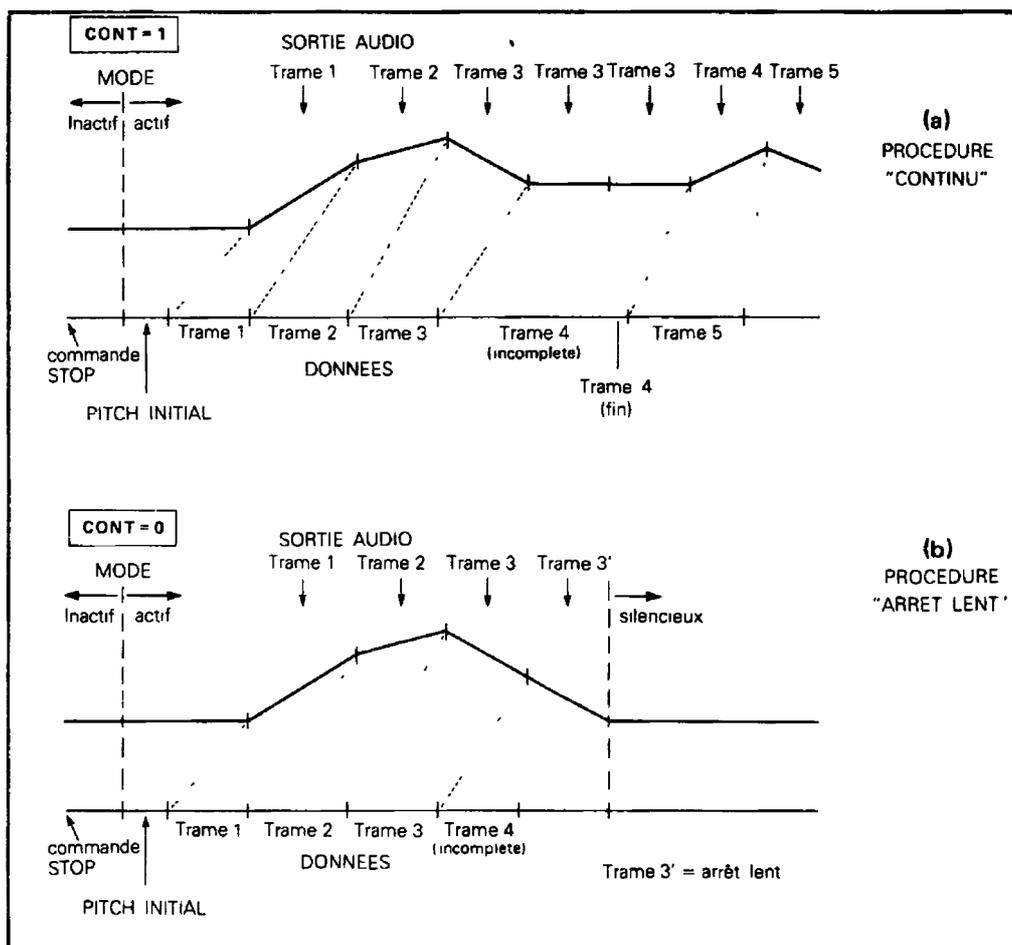


Figure 16 - Evolution du signal de sortie (a) continu, (b) arrêt lent

La figure 17 correspond au cas de fonctionnement "normal", avec arrêt du synthétiseur par une commande STOP. Afin que la dernière trame du message ne soit pas raccourcie, cette commande ne doit pas arriver avant la fin de la prononciation de celle-ci. Le moyen le plus simple d'y parvenir consiste à envoyer une trame silencieuse (constituée par exemple de 4 octets à "00") après la dernière trame utile du message, et d'envoyer ensuite la commande STOP.

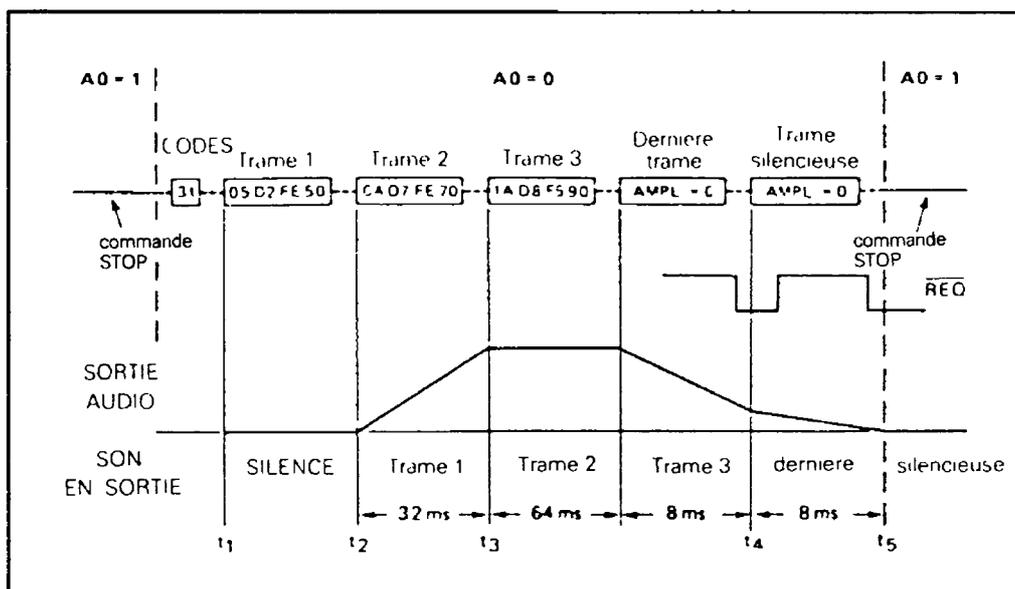


Figure 17 - Evolution du signal de sortie (cas de fonctionnement normal)

## Envoi des données vocales au MEA 8000

Les codes vocaux peuvent être envoyés vers le registre tampon d'entrée du synthétiseur lorsque  $\overline{CE}$ ,  $\overline{W}$  et  $A0$  sont à "0" et  $\overline{R/W}$  à "1", alors que le bit REQ est à l'état "1". REQ est remis à "0" à chaque opération d'écriture d'un octet.

A partir du mode INACTIF, le premier octet reçu est interprété comme la valeur initiale du fondamental (pitch initial) et le bit REQ repasse à l'état "1" au bout d'un maximum de 8 ms. Les quatre octets suivants sont interprétés comme les codes de la première trame du message et, bien que le bit REQ passe à l'état "1" entre chaque octet, il suffit de lire son état à la fin de chaque trame pour envoyer la suivante.

Le MEA 8000 commence donc à parler après le cinquième octet reçu. Pour que le fonctionnement soit normal (ininterrompu), il faut que les octets de la trame suivante soient tous arrivés au MEA 8000 avant la fin de la prononciation de la trame en cours. Compte tenu de la durée de trame minimale de 8 ms, cela ne pose aucun problème pour tout microprocesseur programmé en langage-machine, mais interdit l'emploi du BASIC pour cette partie du programme.

La lecture de REQ par scrutation de D7 interdit au microprocesseur toute autre tâche pendant que le MEA 8000 parle ; pour éviter cela, on devra utiliser la borne  $\overline{\text{REQ}}$  pour actionner une entrée d'interruption. Dans ce cas, la synthèse ne demandera que quelques pour cent du temps du microprocesseur qui pourra vaquer à d'autres occupations comme si de rien n'était.

Ces données nous permettent maintenant de concevoir un logiciel pour commander le MEA 8000 ; c'est l'objet du chapitre suivant.

Page blanche

# Chapitre 3

---

## Logiciel de commande

### Principe

---

Le vocabulaire correspondant à l'application est stocké en mémoire ROM ou RAM selon l'organisation représentée par la figure 18. En général, plusieurs expressions, mots ou éléments de vocabulaire (phonèmes par exemple) sont stockés dans un espace-mémoire déterminé. Afin de faciliter l'accès à ce vocabulaire et pour rendre le programme de commande indépendant de celui-ci, on dispose en début de la zone mémoire de vocabulaire une table d'adresse (directory) indiquant l'adresse de début de chacun de ses éléments. Chacune de ces adresses est codée sur deux octets, poids fort en tête; ceci permet donc de trouver l'adresse de l'adresse d'une expression quelconque à partir de son numéro d'ordre N, par la relation :

$$AN = A0 + 2 \times N$$

où A0 = Adresse de début de table, AN = Adresse de l'adresse de l'expression N.

La table d'adresses se termine par la séquence "FF 00" ou "FF FF" selon le dispositif de codage utilisé.

Chaque expression débute elle-même par un "en tête" de 4 octets, dont les deux premiers représentent le nombre total d'octets de l'expression (en-tête compris) et le quatrième le pitch initial. Le troisième octet représente le pitch en fin d'expression ou est à 00 selon le dispositif de codage; il n'est en général pas exploité par le logiciel de synthèse. Les trames qui suivent étant elles-mêmes codées sur 4 octets, la longueur d'une expression est toujours un multiple de 4 octets.

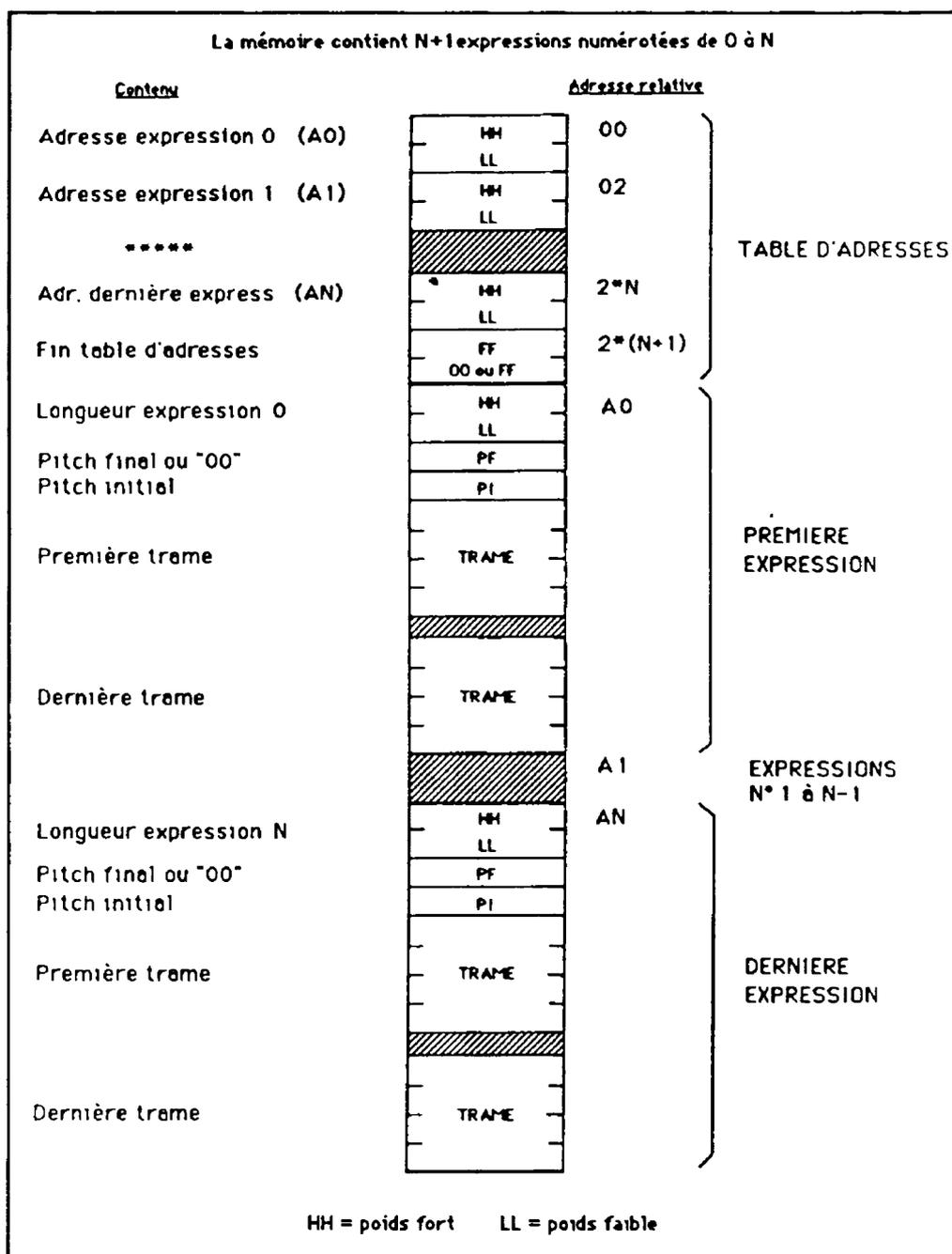


Figure 18 - Organisation de la mémoire de vocabulaire

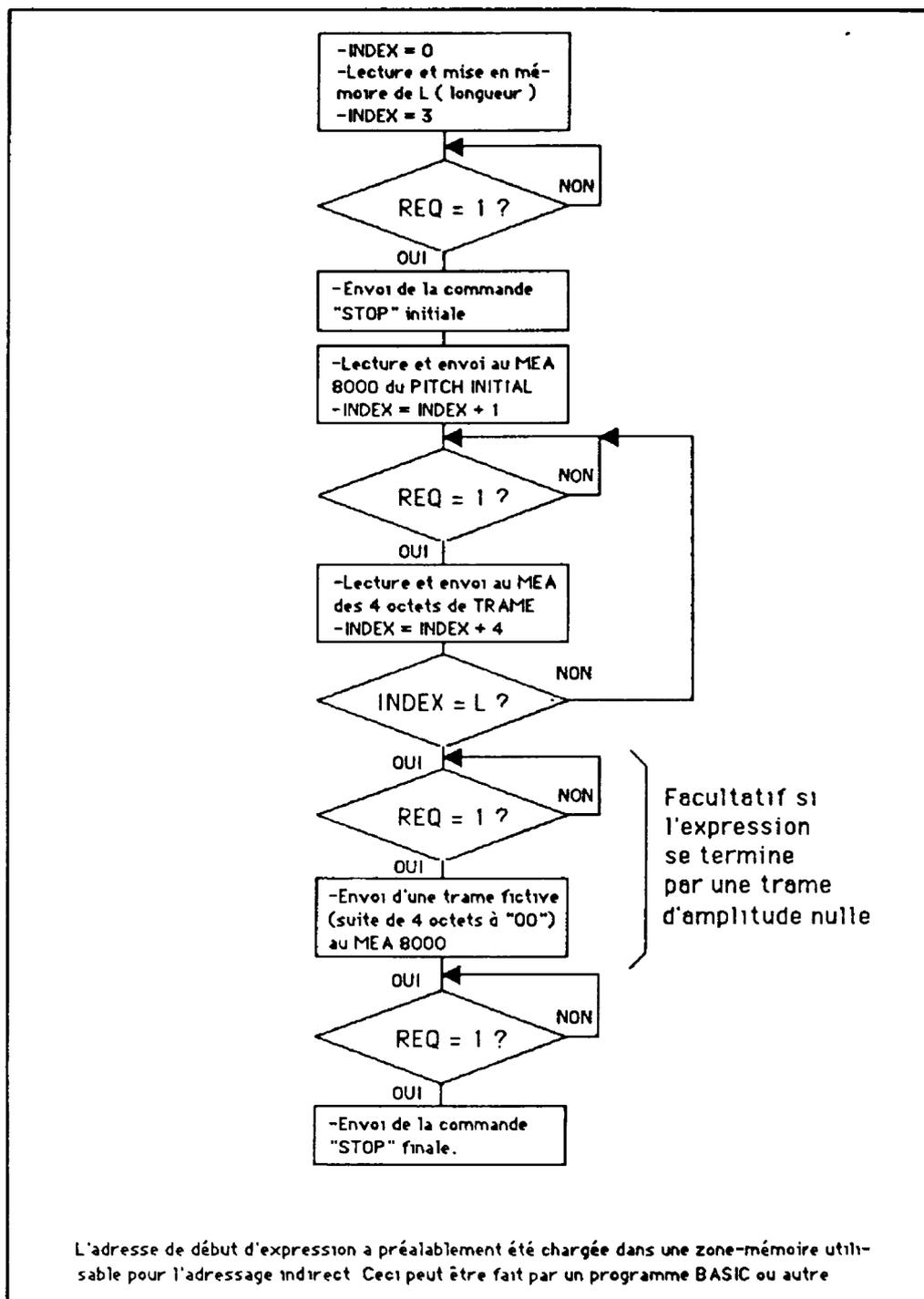


Figure 19 - Organigramme général de la routine de parole

Connaissant ce qu'il faut envoyer au synthétiseur et la façon dont les informations sont stockées dans la mémoire, il est nous maintenant possible d'en déduire l'organigramme général d'une routine permettant de faire prononcer au MEA 8000 une expression en mémoire.

Cette routine devra être réalisée en langage-machine, le BASIC n'étant pas assez rapide pour cette application. Son organigramme est représenté par la figure 19. L'adresse de début de l'expression à prononcer est supposée avoir été préalablement chargée à un endroit approprié pour l'adressage indirect.

Après avoir testé l'état du bit "REQ", on initialise le synthétiseur par l'envoi d'une commande STOP (A0 = 1) qui déterminera en outre la procédure d'arrêt suivie en cas d'interruption dans l'envoi des codes ("continu" ou "arrêt lent").

Il faut ensuite lire les deux octets de la longueur d'expression et les stocker dans un registre ou en mémoire. Il est commode, pour adresser les octets successifs de l'expression, d'utiliser un **index** qui pourra être un registre permettant l'adressage indexé. Cet index sera donc positionné à 0 au départ pour pointer la longueur d'expression; après lecture de la longueur, on le positionnera à 3 pour pointer l'octet de pitch initial.

C'est le premier octet de données vocales à envoyer au MEA 8000 (A0 à "0"), après quoi on incrémente l'index pour pointer l'octet suivant. On teste de nouveau le bit REQ avant l'envoi des trames vocales. Les quatre octets de la trame peuvent être envoyés en bloc sans test de REQ entre chacun d'eux lorsque l'on travaille par scrutation de D7, après quoi on incrémente l'index de 4.

On peut aussi envoyer chaque octet et incrémenter de 1 l'index, ce qui permet d'utiliser la même routine pour l'envoi du pitch initial et des octets de trame; c'est la solution que nous avons choisie.

Après chaque trame, il faut tester si l'index a atteint la valeur de la longueur de l'expression; si ce n'est pas le cas, on teste de nouveau REQ, on envoie la trame suivante, et ainsi de suite. Lorsque les deux valeurs sont égales, on est arrivé à la fin de l'expression et on peut envoyer une trame silencieuse (00 00 00 00) suivie, après test de REQ, de la commande STOP finale.

Cette trame "fictive" n'est pas indispensable si on a pris la précaution, lors du codage, de terminer chaque expression par une trame d'amplitude nulle. C'est le cas pour le vocabulaire fourni en annexe et les expressions créées par les logiciels de composition phonétique décrits qui permettent de se passer de trame fictive dans la routine de base.

**On peut se rendre compte, sur l'organigramme de la figure 19, que certaines opérations sont effectuées à plusieurs endroits du programme, ce qui justifie leur réalisation au moyen d'un sous-programme :**

- STOP: Envoi de la commande STOP**
- TREQ: Test de l'état de REQ**
- TRAM: Envoi d'un octet de données et incrémentation d'index.**

**Le programme principal définit le séquençement des opérations et assure directement la réalisation de celles ne figurant qu'une seule fois (initialisation, lecture de la longueur, test de fin).**

# Application aux microprocesseurs 6800/6802

Bien que plus très employés dans l'industrie, ces deux microprocesseurs (6800 et 6802) sont toujours utilisés dans de nombreux lycées et collèges techniques, car très simples, ce qui est essentiel pour des étudiants qui découvrent les "micros". Pour en avoir fait l'expérience, la synthèse de parole est une application spectaculaire, donc très motivante en milieu scolaire. C'est pourquoi nous donnons les bases de réalisation autour de ces deux microprocesseurs.

Bien que l'utilisation en mode "périphérique" soit tout à fait possible avec l'entrée d'interruption  $\overline{IRQ}$ , nous nous bornerons à l'utilisation en mode "mémoires". Les "fanas de l'IRQ" pourront toujours s'inspirer de l'exemple donné avec le 6809 qui n'est, en fait, qu'un 6800 amélioré!

Le MEA 8000 est perçu, par le micro, comme deux emplacements mémoires qui doivent être consécutives, car un seul bit (A0) permet d'accéder aux registres internes.

A0=0	Registre de données
A0=1	Registre de commande

Figure 20

## Schéma de base

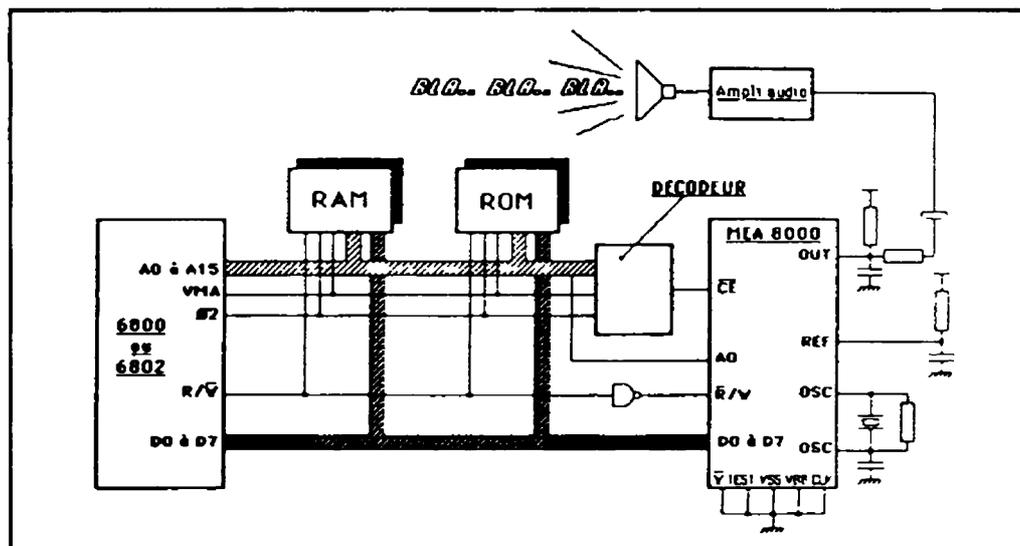


Figure 21

La validation du circuit ( $\overline{CE}$ ) est générée par le circuit de décodage (décodeur) qui doit répondre à l'équation suivante:  $\overline{CE} = X \cdot \overline{VMA} \cdot \Phi 2$  où X = Somme logique des adresses appropriées,  $\overline{VMA}$  = Valid Memory Address (signal de validation de l'accès aux memoires),  $\Phi 2$  = Deuxième phase d'horloge.

$\overline{R/W}$  est issu de  $R/\overline{W}$  après passage dans un inverseur. Les données (D0 à D7) sont reliées directement au bus DATA du micro.

## Logiciel

### Routine de prononciation d'une expression de longueur limitée à 255 octets

Adr.	Op.code	Etiqu.	Mnémo	Commentaire
			ORG \$9F00	
		DONN	EQU \$E7FE	
		RCOM	EQU \$E7FF	
9F00	01	VAL	FCB 1	Rang dans la table
9F01	C6 01	DEBUT	LDA B #1	Premier mot dans la table
9F03	F7 9F00		STA B VAL	Charge VAL
9F06	BD 9F0A		JSR PARL	S/PROG PARL
9F09	3F		SWI	
9F0A	B6 9F00	PARL	LDA A VAL	Compare le rang de la table
9F0D	81 15		CMP A #21T	21 mots dans la table
9F0F	22 26		BHI FIN	Si plus grand: FIN
9F11	48		ASL A	2 octets par adresse
9F12	CE A000		LDX #\$A000	Début de la table
9F15	BD 9F3B		JSR ADXA	Calcul la pos. de l'adr. relat.
9F18	A6 01		LDA A 1,X	Adresse relative
9F1A	CE A000		LDX #\$A000	Début de la table
9F1D	BD 9F3B		JSR ADXA	Calcul l'adresse absolue
9F20	A6 01		LDA 1,X	Longueur du mot
9F22	C6 1B		LDA B #\$1B	Arrêt lent + REQ
9F24	F7 E7FF		STA B RCOM	Registre de commande
9F27	08		INX	
9F28	08		INX	Saute les 3 octets suivants
9F29	08		INX	
9F2A	80 03		SUB A #3	Moins 3 sur la longueur
9F2C	E6 00	RET	LDA B 0,X	Hauteur initiale ou données
9F2E	08		INX	Positionne X
9F2F	7D E7FF	AR3	TST RCOM	Demande d'octet ?
9F32	2A FA		BPL AR3	Non. On boucle sur AR3
9F34	F7 E7FE		STA B DONN	Envoie l'octet demandé
9F37	4A		DEC A	Décompte les octets
9F38	26 F7		BNE RET	Si non zéro, on continue
9F3A	39	FIN	RTS	

### Sous-programme

9F3B 16	ADXA	TAB		Pour conserver A
9F3C 27 04		BEQ	AV1	Test si zéro; Oui alors RTS
9F3E 08	AR1	INX		Plus un sur X
9F3F 5A		DEC B		Moins un sur B
9F40 26 FB		BNE	AR1	Si non égal a 0; Alors AR1
9F42 39	AV1	RTS		

L'exemple logiciel que nous donnons a été réalisé a partir d'une application fonctionnant sur TO7. Les deux registres du MEA 8000 sont implantés aux adresses:

\$E7FE pour le registre de données (DONN)  
et \$E7FF pour le registre de commande (RCOM)

La table d'adresses est positionnée à partir de l'adresse \$A000. La constante VAL définit le rang (dans la table) du mot ou de l'expression que l'utilisateur souhaite entendre prononcer par son dispositif. Le programme "DEBUT" permet d'appeler le sous-programme "PARL" et détermine, en chargeant VAL, le mot qui sera traité par PARL.

PARL compare la valeur de VAL avec le nombre de mots codés disponibles en mémoire (21 dans notre exemple). Si VAL est supérieure, rien ne se passe et on retourne au programme principal. Le sous-programme "ADXA" additionne la valeur de l'index X et l'accumulateur A. Ceci afin de calculer l'adresse de la valeur qu'il convient d'ajouter au début de la table (\$A000) pour pointer sur l'adresse du premier octet du mot choisi. Cette fonction est réalisée par un deuxième appel du sous-programme "ADXA".

Après avoir chargé l'accumulateur A avec la longueur du mot, le MEA 8000 est positionné en mode "ARRET LENT et  $\overline{REQ}$  VALIDE", par l'envoi du code hexadécimal \$1B dans le registre de commande. Le pointeur X est incrémenté de trois et l'accumulateur A, spécifiant la longueur, est décrémenté d'autant. La hauteur initiale est chargée dans B et le pointeur incrémenté de 1. Le bit 7 du registre de commande est testé afin de vérifier s'il y a bien une demande d'octet de la part du MEA 8000.

Dans l'affirmative, on envoie la donnée. Dans le cas contraire, on boucle en AR3 en attendant l'accord d'envoyer du MEA 8000. Puis on teste la fin du mot, sinon on retourne en RET.

# Application au microprocesseur 6809

Le 6809 est le microprocesseur employé sur les machines THOMSON (TO7, TO7/70, MO5, TO9) et TANDY (TRS/80). Il permet, suivant l'application souhaitée, de considérer le MEA 8000 soit comme deux emplacements mémoires consécutives, soit comme un périphérique qui utilise les interruptions, assez nombreuses sur le 6809 (IRQ, FIRQ, NMI), pour interrompre le déroulement du programme en cours afin d'être servi en données vocales. La figure ci-dessous indique le schéma général de raccordement du MEA 8000 au 6809.

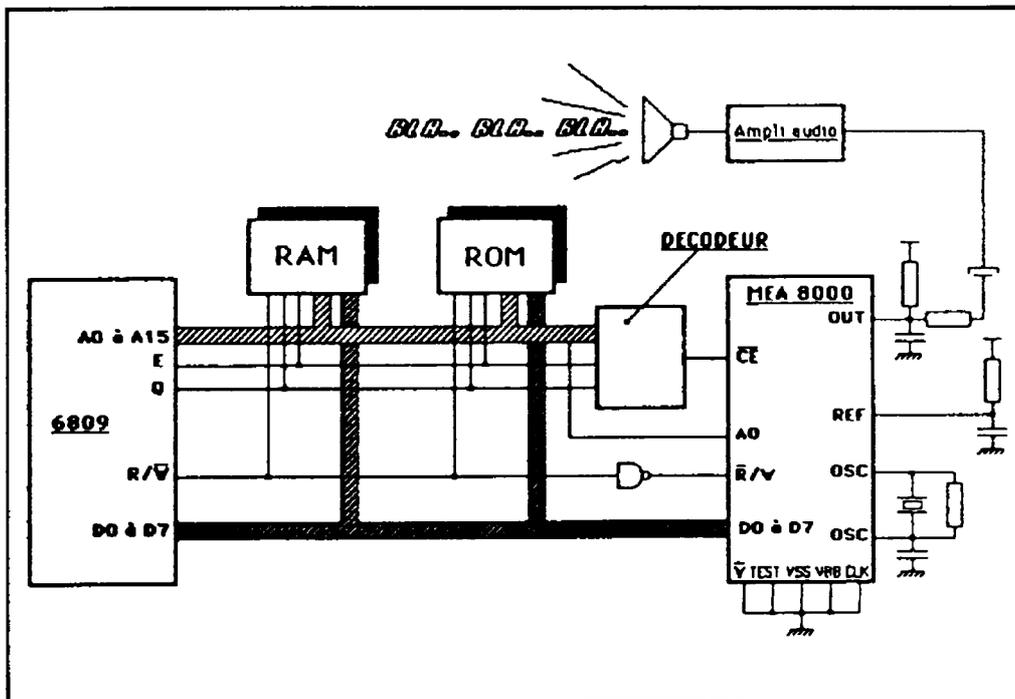


Figure 23

L'entrée A0 du MEA 8000 est connectée directement au 6809.  $\overline{CE}$  est issu d'un décodage approprié des adresses A1 à A15, mais aussi de E qui est, avec Q, un signal qui détermine le cycle de mémorisation des données. R/W du 6809 est inversé par une porte NON avant d'être appliqué au MEA 8000. Le bus de données est raccordé directement aux bornes équivalentes du MEA 8000.

Au point de vue logiciel, nous donnerons comme exemple une application qui "tourne" sur TO7. Les adresses du MEA 8000 ont été réservées, en tenant compte des emplacements disponibles sur TO7, aux locations \$E7FE et \$E7FF. La table des données vocales, précédée du directory (table d'allocation d'adresses relatives) est implantée à partir de l'adresse \$A000.

Le programme "PARL", implanté en \$9FD3, commence par comparer le rang de l'expression proposée avec le nombre de mots disponibles en mémoire et calcule l'adresse du premier octet de l'expression à prononcer.

Le pointeur, chargé dans U, acquis, il convient d'initialiser le MEA 8000 dans le mode de fonctionnement souhaité (mode ARRET LENT et  $\overline{REQ}$  VALIDE - code \$1B - dans notre exemple). Le registre X est ensuite chargé avec la longueur du mot et ajusté en fonction de la valeur du pointeur (U) à cet instant.

Le premier octet correspondant à la hauteur initiale est chargé dans l'accumulateur B, puis le bit 7 du registre de commande est testé afin de valider le chargement du registre de données. Le registre X est décrémente à chaque octet transmis au synthétiseur. Un test après chaque décrémentation permet de détecter la fin du message. Comme ce programme est "emprunté" à l'application de base sur TO7, les listings (source et objet) sont visibles dans le paragraphe concernant cette machine (page 75 ).

# Application aux microprocesseurs 6502/6510

Ces microprocesseurs étant utilisés dans de nombreuses machines très répandues (APPLE IIe et II+, COMMODORE 64, ORIC 1 et ATMOS...) ainsi que d'autres moins connues, les routines de base pourront trouver d'autres applications que les utilisations particulières décrites plus loin. Le 6502 est un processeur relativement ancien qui ne dispose pas d'instructions d'entrée-sortie spécialisées. Le synthétiseur sera donc vu par le microprocesseur comme une mémoire et occupera deux adresses consécutives (A0 = 1 et A0 = 0). La figure 24 indique le schéma général d'interfaçage:

- A0 est directement connectée au MEA 8000 (borne 11)
- A1 à A15 par un décodage approprié fournissent le signal  $\overline{\text{CE}}$  (borne 12); ce même signal peut être éventuellement utilisé pour dévalider une mémoire occupant les mêmes adresses.
- $\overline{\text{R/W}}$  du 6502 est appliqué après inversion à  $\overline{\text{R/W}}$  du MEA 8000 (borne 22)
- D0 à D7 sont connectés directement aux bornes 10 à 3 du MEA 8000.

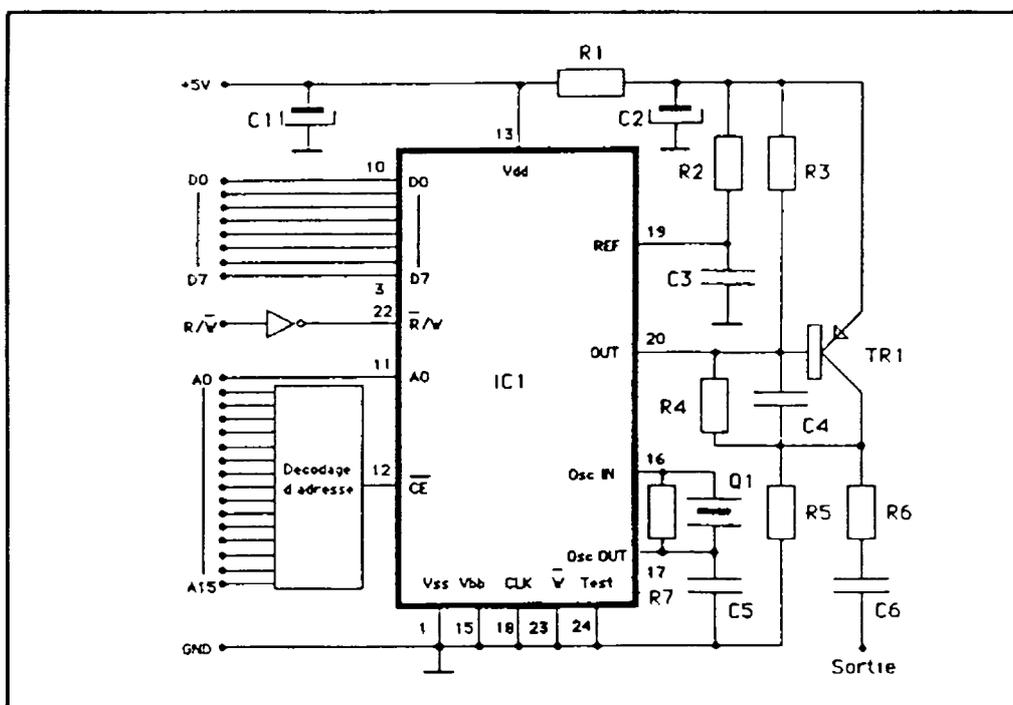


Figure 24 - Schéma général d'interfaçage avec le 6502

## **Routine de prononciation d'une expression de longueur < 256 o.**

Pour le premier exemple, qui correspond à une application sur ORIC, les adresses hexadécimales décodées \$03FE et \$03FF ont été choisies pour le MEA 8000 et le programme est implanté à l'adresse \$B000. Afin de pouvoir aisément adresser les octets successifs composant l'expression, nous utiliserons l'adressage indirect post-indexé; celui-ci n'est toutefois utilisable qu'en page zéro (adresses hexadécimales de \$00 à \$FF).

Dans ce mode d'adressage, le registre d'index est obligatoirement Y. On chargera donc au préalable l'adresse de début de l'expression à prononcer en page zéro, poids faible en tête à une adresse paire. Ici les adresses hexadécimales \$64 et \$65 sont utilisées.

Enfin, pour simplifier cette première application, nous ne lirons que l'octet de poids faible de la longueur d'expression, ce qui ne permettra de prononcer que des expressions de moins de 256 octets (c'est le cas de la plupart des mots isolés du vocabulaire fourni en annexe); cet octet sera stocké à l'adresse hexadécimale \$B0FF et comparé au registre d'index Y après chaque trame pour détecter la fin de l'expression qui est atteinte lorsque ces deux valeurs sont égales. Le listing source commenté de ce programme est donné ci-dessous.

<b>Adr.</b>	<b>Op.code</b>	<b>Etq.</b>	<b>Mnémo.</b>	<b>Commentaire</b>
-------------	----------------	-------------	---------------	--------------------

### **Programme principal**

B000	A0 01		DEBU: LDY \$01	Index = 1 (longueur p. faible)
B002	B1 64		LDA (64),Y	Acc = longueur expression
B004	8D FF B0		STA B0FF	B0FF = long. expression
B007	A0 03		LDY \$03	Index = 3 (pitch initial)
B009	20 2D B0		JSR TREQ	Test de REQ
B00C	20 35 B0		JSR STOP	Initialisation MEA 8000
B00F	20 3B B0		JSR TRAM	Envoi pitch initial + incr.Y
B012	20 2D B0	NEXT:	JSR TREQ	Test de REQ
B015	20 3B B0		JSR TRAM	1 <sup>e</sup> octet trame + incr.Y
B018	20 3B B0		JSR TRAM	2 <sup>e</sup> octet trame + incr.Y
B01B	20 3B B0		JSR TRAM	3 <sup>e</sup> octet trame + incr.Y
B01E	20 3B B0		JSR TRAM	4 <sup>e</sup> octet trame + incr.Y
B021	CC FF B0		CPY B0FF	Index = Longueur? (Y = L?)
B024	30 EC		BMI NEXT	Si NON, tr. suivante (NEXT)
B026	20 2D B0		JSR TREQ	Si OUI, test de REQ
B029	20 35 B0		JSR STOP	Et envoi cde STOP finale
B02C	60		RTS	Fin de l'expression

### **Sous-programme de test du bit d'état (REQ)**

```
B02D A9 80   TREQ: LDA $80   Acc = 80H (1000 0000)
B02F CD FF 03   CMP 03FF   Mot état = Acc? (REQ = 1?)
B032 D0 F9     BNE TREQ   Si NON, recommencer
B034 60        RTS
```

### **Sous-programme d'envoi du mot de commande (STOP)**

```
B035 A9 1A   STOP: LDA $1A   Arrêt lent + REQ sur D7
B037 8D FF 03   STA 03FF   Envoi commande (A0 = 1)
B03A 60        RTS
```

### **Sous-programme d'envoi d'un octet de données vocales (TRAM)**

```
B03B B1 64   TRAM: LDA (64),Y   Acc = octet à envoyer
B04D 8D FE 03   STA 03FE   Envoi donnée vocale (A0 = 0)
B040 C8       INY       Incrément. index (Y = Y + 1)
B041 60        RTS
```

### **Routine de prononciation d'une expression de longueur quelconque**

Ce programme correspond à une application sur APPLE et utilise le même principe d'interfaçage, mais le décodage d'adresse est réduit à sa plus simple expression par connexion de la borne DEVICE SELECT de l'APPLE au CE du MEA 8000. De ce fait, les adresses décodées sont \$C0C0 et \$C0C1 si la carte est connectée sur le slot n° 4. Ici, le programme permet la prononciation d'une expression de longueur et d'adresse de début quelconques.

L'adresse de début sera préalablement chargée en page zéro aux adresses \$FE (poids faible) et \$FF (poids fort), de façon à permettre l'adressage indirect post-indexé par Y. Cette indexation ne pouvant se faire que sur 256 octets, nous incrémenterons le contenu de l'adresse \$FF chaque fois que Y passera par 0 ("retenue"), ainsi que le registre X qui servira à faire la comparaison sur le poids fort de la longueur pour détecter la fin de l'expression. Le programme débute à l'adresse \$6EF3 (DEBU) par la lecture de la longueur de l'expression et sa sauvegarde aux adresses \$6FFE (poids fort) et \$6FFF (poids faible). On peut le commencer à l'adresse \$6F00 (DEBU2) si ces opérations sont effectuées par un autre programme appelant celui-ci.

Les sous-programmes TREQ et STOP sont identiques à ceux du précédent, à l'exception des adresses, et seuls le programme principal et le sous-programme TRAM diffèrent en raison de l'utilisation des registres X et Y. Le listing source commenté est représenté à la page 50.

Adr.	Op.code	Etiqu.	Mnémo.	Commentaire
<b>Programme principal</b>				
6EF3	A0 00	DEBU:	LDY \$00	Y = 0 (index p. fort longueur)
6EF5	B1 FE		LDA (FE),Y	Lecture poids fort longueur
6EF7	8D FE 6F		STA 6FFE	Sauvegarde en 6FFE
6EFA	C8		INY	Y = 1 (index p. faible long.)
6EFB	B1 FE		LDA (FE),Y	Lecture p. faible longueur
6EFD	8D FF 6F		STA 6FFF	Sauvegarde en 6FFF
6F00	A2 00	DEBU2:	LDX \$00	X = 0 (poids fort ad. relative)
6F02	A0 03		LDY \$03	Y = 3 (pitch initial)
6F04	20 2D 6F		JSR TREQ	Test de REQ
6F07	20 35 6F		JSR STOP	Initialisation MEA 8000
6F0A	20 3B 6F		JSR TRAM	Envoi pitch initial + incr.Y
6F0D	20 2D 6F	NEXT:	JSR TREQ	Test de REQ
6F10	20 3B 6F		JSR TRAM	1 <sup>e</sup> octet trame + incr.Y
6F13	20 3B 6F		JSR TRAM	2 <sup>e</sup> octet trame + incr.Y
6F16	20 3B 6F		JSR TRAM	3 <sup>e</sup> octet trame + incr.Y
6F19	20 3B 6F		JSR TRAM	4 <sup>e</sup> octet trame + incr.Y
6F1C	CC FF 6F		CPY 6FFF	Y = Poids faible longueur?
6F1F	D0 EC		BNE NEXT	Si NON, tr. suivante (NEXT)
6F21	EC FE 6F		CPX 6FFE	X = Poids fort longueur?
6F24	30 E7		BMI NEXT	Si NON, tr. suivante (NEXT)
6F26	20 2D 6F		JSR TREQ	Si OUI, test de REQ
6F29	20 35 6F		JSR STOP	Et envoi cde STOP finale
6F2C	60		RTS	Fin de l'expression
<b>Sous-programme de test du bit d'état (REQ)</b>				
6F2DA	9 80	TREQ:	LDA \$80	Acc = 80H (1000 0000)
6F2F	CD C1 C0		CMP C0C1	Mot état = Acc? (REQ = 1?)
6F32	D0 F9		BNE TREQ	Si NON, recommencer
6F34	60		RTS	
<b>Sous-programme d'envoi du mot de commande (STOP)</b>				
6F35	A9 1A	STOP:	LDA \$1A	Arrêt lent + REQ sur D7
6F37	8D C1 C0		STA C0C1	Envoi commande (A0 = 1)
6F3A	60		RTS	
<b>Sous-programme d'envoi d'un octet de données vocales (TRAM)</b>				
6F3B	B1 FE	TRAM:	LDA (FE),Y	Acc = octet à envoyer
6F3B	8D C0 C0		STA C0C0	Envoi donnée vocale (A0 = 0)
6F40	C8		INY	Incrément. index (Y = Y + 1)
6F41	C0 00		CPY \$00	Poids faible longueur = 0?
6F43	D0 03		BNE 6F48	Si NON, retour
6F45	E8		INX	Si OUI, incrémenter X ...
6F46	E6 FF		INC FF	Et le contenu de FF (p. fort)
6F48	60		RTS	

# Application au microprocesseur Z-80 (ou Z-80A)

Ces microprocesseurs, utilisés dans de nombreuses machines 8 bits récentes (AMSTRAD, MSX, PHILIPS VG 5000, SHARP...), sont dotés d'instructions et de signaux d'entrée-sortie spécialisés. Ceci permettra de considérer le synthétiseur comme un périphérique et non comme une mémoire, en utilisant le signal  $\overline{\text{IORQ}}$  du Z-80.

Le synthétiseur occupera deux adresses consécutives ( $A_0 = 0$  et  $A_0 = 1$ ) parmi les 256 possibles par le décodage de  $A_0$ - $A_7$ . La figure 25 représente le schéma général de connexion :

- $A_0$  est directement reliée à la borne  $A_0$  (borne 11)
- $A_1$  à  $A_7$  et  $\overline{\text{IORQ}}$  inversé, fournissent par décodage  $\overline{\text{CE}}$  (borne 12)
- $\overline{\text{RD}}$  commande  $\overline{\text{R/W}}$  du MEA 8000 (borne 22) après compensation du temps de propagation par 2 inverseurs
- $D_0$  à  $D_7$  sont directement reliées aux bornes 10 à 3 du MEA 8000.

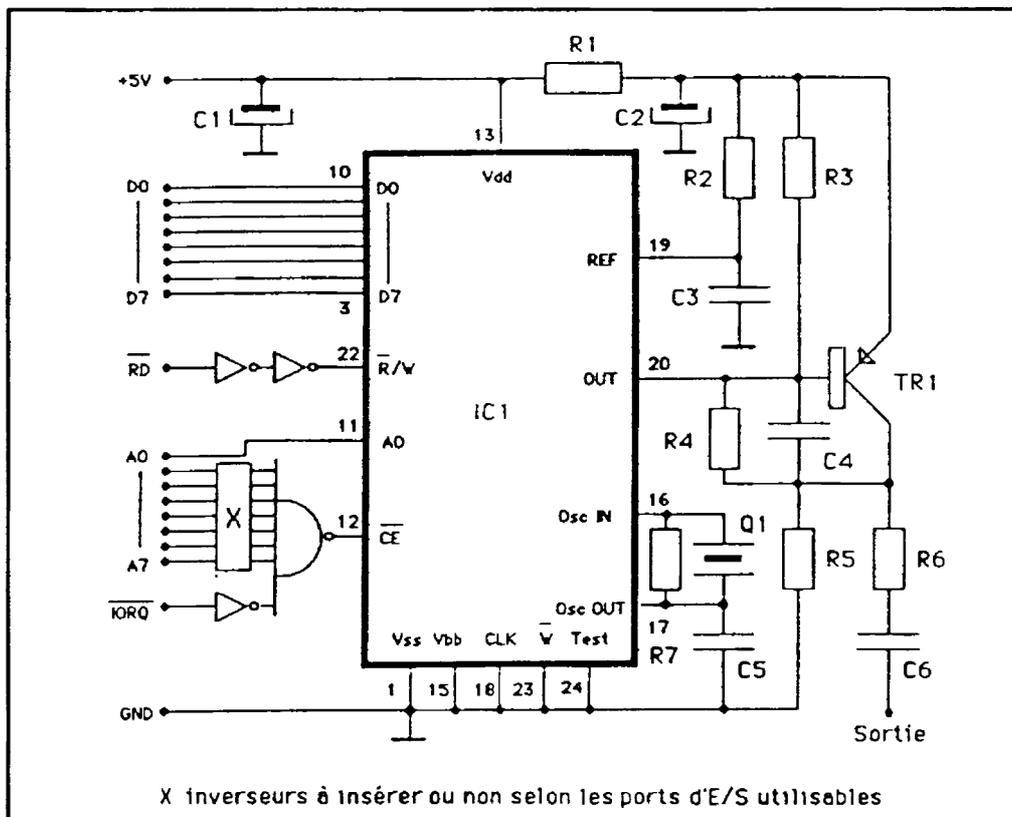


Figure 25 - Schéma général d'interfaçage avec le Z-80

### Routine de prononciation d'une expression de longueur quelconque

L'exemple proposé ici correspond à une application sur SHARP MZ-700, et les adresses d'entrée-sortie choisies sont \$DE (données) et \$DF (commande). Le programme commence à l'adresse \$A5B8 et l'adresse de début de l'expression à prononcer doit avoir été préalablement chargée aux adresses \$A5FE (poids faible) et \$A5FF (poids fort). Le programme commence par stocker l'adresse de début dans la paire de registres HL, qui sera utilisée pour pointer l'octet en cours par incrémentation. La paire de registres BC est ensuite chargée par la longueur de l'expression, et décrémentée à chaque octet; le test de sa valeur après chaque trame permettra de détecter la fin de l'expression lorsque BC deviendra nul. Pour ce programme, dont l'architecture est identique à celle des deux précédents, les mêmes noms d'étiquettes ont été utilisés. Son listing source commenté est visible ci-dessous.

Adr.	Op.code	Etiqu.	Mnémo.	Commentaire
<b>Programme principal</b>				
A5B8	00		DEBU:NOP	E5 (PUSH HL) pour VG 5000
A5B9	2A FE	A5	LD HL,(A5FE)	Adresse expression dans HL
A5BC	46		LD B,(HL)	Poids fort longueur dans B
A5BD	23		INC HL	HL pointe p. faible longueur
A5BE	4E		LD C,(HL)	Poids faible longueur dans C
A5BF	23		INC HL	
A5C0	23		INC HL	HL pointe pitch initial
A5C1	0B		DEC BC	
A5C2	0B		DEC BC	
A5C3	0B		DEC BC	BC = longueur restante
A5C4	CD EB	A5	CALL TREQ	Test de REQ
A5C7	CD F2	A5	CALL STOP	Initialisation MEA 8000
A5CA	CD F7	A5	CALL TRAM	Pitch initial + inc.HL + dec.BC
A5CD	CD EB	A5	NEXT:CALL TREQ	Test de REQ
A5D0	CD F7	A5	CALL TRAM	1° octet trame + i.HL + d.BC
A5D3	CD F7	A5	CALL TRAM	2° octet trame + i.HL + d.BC
A5D6	CD F7	A5	CALL TRAM	3° octet trame + i.HL + d.BC
A5D9	CD F7	A5	CALL TRAM	4° octet trame + i.HL + d.BC
A5DC	AF		XOR A	Acc = 0
A5DD	B9		CP C	P. faible longueur = 0 ?
A5DE	20 ED		JR NZ,NEXT	Si NON, tr. suivante (NEXT)
A5E0	B8		CP B	P. fort longueur = 0 ?
A5E1	20 EA		JR NZ,NEXT	Si NON, tr. suivante (NEXT)
A5E3	CD EB	A5	CALL TREQ	Si OUI, test de REQ
A5E6	CD F2	A5	CALL STOP	Et envoi cde STOP finale
A5E9	C9		RET	Fin de l'expression
A5EA	00		NOP	E1 (POP HL) pour VG 5000

### **Sous-programme de test du bit d'état (REQ)**

A5EB DB DF	TREQ:IN A,DFH	Acc = Mot d'état
A5ED CB 7F	BIT 7,A	Test de REQ (D7 = 1 ?)
A5EF 28 FA	JR Z,TREQ	Si NON, recommencer
A5F1 C9	RET	

### **Sous-programme d'envoi du mot de commande (STOP)**

A5F2 3E 1A	STOP:LD A,1AH	Arret lent + REQ sur D7
A5F4 D3 DF	OUT DFH,A	Envoi commande (A0 = 1)
A5F6 C9	RET	

### **Sous-programme d'envoi d'un octet de données vocales (TRAM)**

A5F7 7E	TRAM:LD A,(HL)	Acc = octet à envoyer
A5F8 D3 DE	OUT DEH,A	Envoi donnée vocale (A0 = 0)
A5FA 23	INC HL	Incrément. adresse octet
A5FB 0B	DEC BC	Décrem. longueur restante
A5FC C9	RET	

Page blanche

# Chapitre 4

---

## Vocabulaire et phonétique

Nous avons étudié, dans les chapitres précédents, les principes généraux de la synthèse de parole, leur mise en œuvre dans le MEA 8000, l'interface avec le microprocesseur et le logiciel de base permettant la prononciation d'une expression existant déjà en mémoire. Nous allons maintenant voir de quelle manière ce vocabulaire peut être constitué en fonction de l'objectif à atteindre, ainsi que les avantages et inconvénients respectifs des différentes possibilités.

### Création du vocabulaire codé

---

Pour créer le vocabulaire codé au format que nous avons vu au chapitre précédent, il faut disposer d'un système d'analyse et de codage de la parole tel que celui proposé par RTC, visible sur la photographie. Ce système se compose de deux éléments principaux :

— Un ordinateur 16 bits muni d'une interface IEEE 488 (= CEI 625, HPIB, GPIB) sur lequel un logiciel spécifique sera exécuté. Aujourd'hui, ce logiciel existe pour deux types de machines (IBM PC et HP 9816S).

— Une boîte d'adaptation (SAB) connectée à l'ordinateur par le BUS IEEE 488 et incorporant 5 blocs fonctionnels (interface d'entrée-sortie et conversion A/N et N/A, interface IEEE 488, synthèse de parole, programmation d'EPROM, alimentation).

Cet ensemble muni de son logiciel permet, à partir d'un enregistrement magnétique, la réalisation de toutes les étapes nécessaires à la création du vocabulaire codé en EPROM; la phase d'édition (correction /compression) qui est décisive pour la qualité du résultat, est grandement facilitée par la représentation graphique de tous les paramètres. Selon le mode de compression choisi (automatique ou manuel), les débits d'information seront compris entre 1 000 et 2 000 bits par seconde de parole. La partie spécifique (SAB et logiciel) est commercialisée par le fabricant sous la référence OM 8010.



*Système de codage RTC OM 8010, ici en fonctionnement sur le micro-ordinateur HP 9816S*

# Quel vocabulaire ?

## Sous quelle forme ?

---

Le vocabulaire peut être stocké sous l'une des formes suivantes :

— Phrases complètes préenregistrées et codées spécialement pour une application déterminée. Cette méthode est celle qui donne, du point de vue de la qualité acoustique globale, les meilleurs résultats car elle conserve les caractéristiques prosodiques (intonation et rythme) du message d'origine. Elle a pour inconvénient principal de nécessiter un codage spécifique relativement coûteux générant un vocabulaire figé, limité à l'application prévue. De ce fait, cette méthode est en général réservée aux applications de moyenne ou grande série, telles que l'automobile par exemple.

— Mots ou éléments de phrase isolés permettant par combinaison de constituer un certain nombre de messages. Une application assez courante de cette méthode consiste à utiliser une partie de phrase fixe avec une portion variable, constituée de chiffres par exemple. Cette méthode, si elle est un peu plus souple que la précédente, donne d'un peu moins bons résultats; en effet, d'une part un mot n'est pas prononcé de la même manière selon sa position dans la phrase, et d'autre part les liaisons existant entre les mots en français, notamment dans les chiffres composés, ne facilitent pas les choses... Le vocabulaire fourni en annexe permet néanmoins d'intéressantes applications de ce type (répondeur, horloge parlante, etc.) dont certains exemples seront décrits.

— Éléments phonétiques prédéfinis (phonèmes, diphtongues) : c'est cette dernière méthode qui retiendra le plus notre attention car c'est la seule qui permette l'accès à un vocabulaire illimité (dans une langue déterminée) sans nécessiter de codage préalable (synthèse à partir du texte). Elle implique de disposer d'un "dictionnaire" d'éléments phonétiques et des règles pour s'en servir, qui détermineront la qualité des résultats; quelques notions de phonétique sont nécessaires pour l'utiliser.

# Un peu de phonétique

---

Toute langue parlée n'utilise qu'un nombre relativement limité de sons élémentaires qui permettent par leur concaténation de composer tous les messages imaginables dans cette langue. On appelle "phonèmes" ces sons élémentaires dont le nombre peut varier d'une trentaine à une soixantaine selon la langue ou le dialecte considérés. La liste de ces phonèmes varie notablement d'une langue à l'autre, même pour des langues d'origine commune, latine ou saxonne par exemple. C'est ce qui explique les résultats très médiocres obtenus lorsque l'on essaie de synthétiser un texte à partir d'éléments phonétiques développés pour une autre langue. En ce qui concerne le français, on compte généralement 37 phonèmes, dont certains sont proches (par exemple les sons "in", "ein", "ain", "un", "eun"... que la pratique quotidienne tend à confondre).

L'utilisation de phonèmes offre l'avantage de ne nécessiter qu'un dictionnaire restreint, donc peu encombrant en mémoire, et d'être simple d'utilisation si on se contente d'une entrée phonétique des expressions à vocaliser. C'est pourquoi les principales applications décrites dans cet ouvrage, dont la destination première est l'initiation en vue d'utilisations individuelles, sont basées sur cette méthode.

Il faut cependant reconnaître que la qualité obtenue avec la synthèse par phonèmes est, par son principe, limitée. En effet, le phonème est une entité théorique qui n'existe pas dans le langage parlé réel, en raison du phénomène connu sous le nom de "coarticulation": chaque son émis est influencé par son prédécesseur et son successeur, du fait même des caractéristiques mécaniques de l'appareil vocal. Ces zones de transition entre un son et un autre sont très importantes, et on est conduit pour les respecter à définir un autre élément phonétique: le diphone ou diphonème.

Celui-ci s'étend du milieu de la partie "stable" d'un son au milieu de la partie stable de son successeur. On voit donc, si l'on considère les 37 sons élémentaires du français, qu'il y a  $37 \times 37 = 1369$  combinaisons, donc diphones, différents. En pratique, certains sont très voisins ou inusités, et l'on considère que 1200 diphones permettent d'obtenir de très bons résultats. On imagine toutefois sans difficulté que l'encombrement d'un dictionnaire de diphones sera environ 30 fois supérieur au dictionnaire de phonèmes correspondant, et que le logiciel d'utilisation sera plus complexe.

En France, le CNET (Centre National d'Etudes des Télécommunications) de Lannion a développé des dictionnaires de diphtongues et des logiciels de synthèse à partir du texte écrit, utilisables sur un synthétiseur spécifique. Outre les résultats qualitatifs dus à l'utilisation de diphtongues et à la génération d'une prosodie correcte, le logiciel de conversion orthographique/phonétique utilisé permet une entrée du texte sous forme habituelle ; il lève en effet la plupart des ambiguïtés de prononciation telles que celle de la célèbre phrase "les poules couvent au couvent". Ces résultats sont le fruit de nombreuses années de travaux d'équipes pluridisciplinaires et ne sont donc pas du domaine public.

Une adaptation au MEA 8000 sur un micro-ordinateur APPLE II en a été faite par la société MATRA ; elle est constituée d'une carte et d'un logiciel spécifiques commercialisés sous le nom de "PORTE-PAROLE".

Un dictionnaire de diphtongues, développé par l'IPO d'Eindhoven (Hollande), existe également sur le MEA 8000 pour le néerlandais.

Une autre méthode encore peu développée en France pour la synthèse à partir du texte est la synthèse "par règles" qui, à partir d'un dictionnaire de phonèmes calcule l'évolution (trajectoire) entre les paramètres de deux phonèmes successifs au moyen de modèles mathématiques ou "règles". Cette méthode a généralement recours aux formants comme paramètres, et a été utilisée avec succès aux U.S.A., en Suède et au Canada.

# La prosodie ?

## Ce n'est pas si simple...

---

Ce que l'on appelle la prosodie d'une phrase, c'est en fait essentiellement l'information supplémentaire transmise par un message parlé par rapport au même message écrit, c'est-à-dire l'intonation, l'accent tonique et le rythme de la phrase. Les principaux paramètres influençant la prosodie sont :

— La variation du fondamental (pitch) est le paramètre déterminant l'intonation, et la même phrase peut voir son caractère passer du mode affirmatif au mode interrogatif par sa seule modification. C'est le paramètre le plus important pour la prosodie globale d'une expression.

— Les variations locales de l'amplitude caractérisent l'accent tonique. Celui-ci est toutefois peu marqué en français, où toutes les syllabes sont prononcées avec des intensités assez constantes. Ce paramètre est donc relativement secondaire pour la synthèse.

— La durée des syllabes successives définit le rythme de la phrase, qui pourra être modifié en jouant sur la durée des phonèmes la composant.

Si les paramètres déterminant ces caractéristiques sont relativement bien connus, les lois régissant leurs variations en fonction de la progression de la phrase sont assez complexes, et nous nous bornerons à fournir quelques règles générales permettant d'ajouter un peu de naturel aux messages construits à partir de phonèmes, ou de modifier l'intonation d'un message prédéfini.

## Intonation

Le fondamental, qui rappelons-le est la fréquence de vibration des cordes vocales lors des sons voisés, varie entre 70 et 200 Hz pour l'homme adulte et entre 100 et 300 Hz pour la femme adulte dans la conversation courante. Il peut dépasser largement ces limites lors du chant ou de situations particulières. Dans une phrase, la courbe de variation de ce paramètre n'est en général pas uniforme et présente de nombreux changements de pente; nous allons essayer de donner quelques indications permettant d'établir une telle courbe.

- Allure générale de la courbe : on a remarqué qu'en général la courbe d'évolution du fondamental présentait une allure descendante au long de la phrase ; ceci semble vérifié dans toutes les langues, et donc probablement dû à des causes physiologiques. Le fait de créer une telle "ligne de déclinaison" donnera donc un certain naturel à la parole synthétique. Autour de cette ligne, on observe des variations de pente locales au niveau des syllabes composant le message.

- Variations de pente locales : on a observé depuis très longtemps que, dans une phrase de structure simple, si celle-ci était sur le mode affirmatif, la pente de la courbe d'intonation de la dernière syllabe, accentuée, était négative ; à l'inverse la pente est positive si le mode est interrogatif. Ainsi la phrase "Il fait beau!" aura une pente descendante sur "beau" alors qu'une pente montante sur "beau" évoquera une interrogation :

	Il fait BEAU	
Mode affirmatif	\	!
Mode interrogatif	/	?

De nombreux chercheurs se sont efforcés de déterminer les règles gouvernant les variations de pente locales à l'intérieur de la phrase. Les résultats de ces recherches ont permis d'établir certains liens entre la syntaxe et la prosodie de la phrase ; ces règles sont complexes et donc difficilement utilisables par un amateur. Il a cependant pu être établi que les variations de pente sont régies par une opposition entre la pente de la syllabe accentuée terminant la phrase (déterminée par le mode affirmatif ou interrogatif) et celle de la syllabe accentuée du mot lui faisant face.

Par exemple, pour la phrase "Geneviève partira" les syllabes accentuées sont montrées en majuscule, et la pente correspondante est indiquée au dessous pour les deux modes possibles :

	geneVIEve partiRA	
Mode affirmatif	/	\ !
Mode interrogatif	\	/ ?

Ceci est relativement simple à mettre en œuvre sur des phrases courtes du type "sujet-verbe" telle que celle ci-dessus, où l'opposition se fait entre sujet et verbe, et "sujet-verbe-complément", où l'opposition a lieu entre sujet et complément, comme ci-dessous :

	geneVIEve partira deMAIN	
Mode affirmatif	/	\ !
Mode interrogatif	\	/ ?

Dans des phrases plus longues, le principe est le même, mais on devra distinguer les oppositions à plusieurs niveaux, ce qui rend l'opération beaucoup plus délicate.

**Il n'est pas facile de mettre en évidence ce type de structure dans une phrase un peu compliquée, non plus que de découvrir les syllabes accentuées dans la phrase. Dans beaucoup de cas, on devra procéder par essais et erreurs. On aura toujours la possibilité d'imprimer une ligne de déclinaison et de fixer la pente de la dernière syllabe en fonction du mode (affirmatif ou interrogatif) recherché, si l'on veut éviter la "voix de robot".**

## **Rythme**

**La durée relative des syllabes d'une phrase en détermine le rythme qui est un facteur important du naturel du message. Les lois régissant ces durées sont encore mal connues et si l'on désire travailler ce point, il faudra procéder par tâtonnements. On pourra pour cela ajouter ou retrancher des trames aux syllabes (essentiellement à la voyelle la composant), ou modifier la durée de certaines trames.**

**Avec la version de phonèmes 4.2, codée avec une durée de trame fixe de 16 ms, il sera également possible de modifier la vitesse globale d'élocution en changeant la durée de trames sur l'ensemble de la phrase. On pourra ainsi doubler ou diviser par deux la vitesse en passant toutes les trames à 8 ou 32 ms respectivement; des vitesses intermédiaires pourront être obtenues en ne faisant cette opération qu'une trame sur deux.**

# Les phonèmes du MEA 8000

---

Nous avons vu que le français utilisait théoriquement 37 phonèmes pour composer son vocabulaire. Dans la pratique, nous en avons identifié 31 suffisamment différenciés par l'usage courant pour nécessiter un code particulier. Nous avons cependant trouvé intéressant d'ajouter à ces phonèmes 7 sons composés (diphthongues) dont la reproduction par concaténation de 2 phonèmes n'était pas très satisfaisante. D'autre part, deux silences de durée 32 et 64 ms ont été ajoutés pour servir de séparateurs, par exemple comme des signes de ponctuation. La liste de ces 40 éléments avec leur représentation écrite et un exemple d'utilisation pour ceux dont la forme écrite est ambiguë est fournie en annexe p. 258.

Dans les applications sur les diverses machines, nous utiliserons un caractère pour représenter chaque phonème, en utilisant la lettre correspondante de l'alphabet lorsque la correspondance entre lettre et phonème est unique. Pour les sons habituellement représentés par un groupe de lettres, nous utiliserons l'un des symboles spéciaux du clavier. Toutes les machines ne disposant pas exactement du même jeu de caractères, cette liste variera légèrement de l'une à l'autre.

Tous les phonèmes sont codés avec un pitch constant (120 Hz) afin qu'ils puissent s'enchaîner dans un ordre quelconque sans discontinuité ; ceci a pour effet de produire une "voix de robot" à laquelle il sera possible de donner une intonation artificielle par l'utilisation de marqueurs dans l'expression créée.

Afin de permettre une concaténation satisfaisante, les phonèmes ont été normalisés en amplitude (avec une croissance, une partie plate et une décroissance) et en durée (en général 128 ms pour les voyelles). Pour les consonnes plosives (p, t, k, b, d, g) celles-ci sont toujours précédées d'un silence dû à l'occlusion de la bouche qui les précède ; c'est pourquoi un silence de 32 ms est codé au début de ces phonèmes (version 4.2).

Enfin, deux tables de phonèmes différentes ont été codées :

— la première (version 3.3) a été codée avec une durée de trame variable de façon à minimiser son encombrement, et occupe exactement 1 kilo-octet ;

— la seconde (version 4.2) a été codée avec une durée fixe de 16 ms, ce qui augmente son encombrement de près de 50%, mais ajoute d'intéressantes possibilités, comme par exemple la variation de la vitesse d'élocution par modification de la durée de trame ; la trame fixe facilite également la conception d'un programme d'édition graphique. Ces deux tables utilisant les mêmes numéros de code pour leurs phonèmes sont compatibles en logiciel, à l'encombrement près.

## **Phonèmes, mode d'emploi**

Pour utiliser ces phonèmes, deux possibilités (au moins) nous sont offertes :

— fabriquer par concaténation une expression à un emplacement mémoire déterminé, en allant chercher les codes des phonèmes dans la table. Cette expression sera codée selon le format décrit au chapitre précédent (en-tête de 4 octets). Ceci peut être fait par un programme BASIC qui appellera la routine en langage-machine de commande du MEA 8000 pour la vocalisation ;

— vocaliser les phonèmes directement où ils se trouvent dans la table, leur séquence déterminant l'expression prononcée. Ceci implique, pour des raisons de rapidité, un programme entièrement en langage-machine, pouvant être appelé par un programme utilisateur, en BASIC par exemple.

Chacune de ces deux méthodes présente ses avantages propres, et les deux solutions seront proposées.

La première solution permet de modifier certains paramètres pour optimiser la prosodie de la phrase (l'intonation avec le pitch, le rythme en ajoutant ou retranchant des trames), et d'obtenir des effets particuliers (chuchotement par "dévoisement", vitesse d'élocution en modifiant la durée de trame). L'expression ainsi éditée pourra être sauvegardée et rappelée par un programme utilisateur. Un programme mettant en œuvre cette solution est fourni dans les chapitres suivants pour chacune des machines étudiées.

La deuxième solution offre l'avantage d'une prononciation immédiate de l'expression créée, et de ne pas nécessiter d'autre espace mémoire que celui utilisé par le dictionnaire de phonèmes et le programme L.M., quels que soient le nombre et la longueur des expressions désirées. En contre-partie, la voix sera très "robotique", aucune édition n'étant possible ici. Un programme de ce type est donné pour chacun des microprocesseurs étudiés ici (6502, Z-80, 6809).

Enfin un programme en langage-machine pour les MO5, TO7, TO7/70 baptisé "Phonetram" et permettant une édition graphique de tous les paramètres, est abondamment détaillé.

## Et le chant ?

---

Le chant est un cas particulier de parole dans lequel la hauteur n'est pas déterminée par l'intonation du message, mais par la mélodie. Le texte à chanter est une suite de syllabes qui représentent chacune une note dont la hauteur et la durée sont indiquées dans la partition. Par exemple, pour le début de "Au clair de la lune", la séquence est :

Texte	Au	clair	de	la	lu - ne
Notes	do	do	do	ré	mi ré
Durée	Nr	Nr	Nr	Nr	Bl Bl

(Nr = noire. Bl = blanche)

Donc pour la création d'un chant à partir d'un texte composé au moyen de phonèmes, il faudra d'une part adapter la durée des syllabes et d'autre part leur donner la hauteur correspondant à la note à jouer. Avec le MEA 8000, il sera possible de créer une expression correspondant au texte de la chanson au moyen des phonèmes, puis de modifier la hauteur de chacune des syllabes pour la faire correspondre à la note désirée, et enfin de définir la durée de chaque note soit par modification de la durée de trame, soit en ajoutant ou retranchant des trames à la voyelle caractérisant la note.

En pratique, la hauteur ne pourra varier que sur deux octaves environ (80 à 400 Hz avec la voix utilisée pour les phonèmes) pour rester "vraisemblable". La gamme de variation de durée sera dans un rapport 8 en jouant sur la durée de trame (soit de la croche à la ronde par exemple) ; on pourrait obtenir un rapport plus important en modifiant le nombre de trames d'une syllabe, voire en jouant sur le nombre et la durée des trames. Pour des raisons de simplicité, nous nous sommes limités à la variation de la durée de trame dans les exemples proposés.

La "chanson" ainsi créée occupe le même nombre d'octets que la même expression non chantée ; seuls les bits définissant la variation de hauteur et la durée de trame sont modifiés. Un exemple de programme permettant la création d'une expression chantée est donné pour deux des machines adaptées, et sa transposition aux autres à partir du programme de synthèse par phonèmes sera aisée.

Page blanche

# Chapitre 5

---

## Applications

### Application sur MO5, TO7, TO7/70, et TO9 (6809)

---

Nous allons étudier maintenant la réalisation, tant matérielle que logicielle, de la synthèse vocale utilisable avec les micro-ordinateurs TO7, TO7/70, TO9, MO5 et autres matériels à venir, sous réserve que THOMSON conserve le même type de raccordement au bus du microprocesseur et un microprocesseur 6809. Pour le matériel THOMSON, cette "interface vocale" intitulée PAROLE est commercialisée par CEDIC/NATHAN (voir bon de commande en annexe).

PAROLE est accompagnée d'une cassette de présentation qui vous permettra, sans programmation personnelle, d'apprécier immédiatement sa qualité phonique. Elle est la partie matérielle d'une synthèse vocale réalisée autour du circuit MEA 8000 de RTC.

Le circuit MEA 8000 est un synthétiseur vocal qui utilise la technique des formants et permet d'obtenir une qualité de parole remarquable comparée aux synthèses existantes actuellement sur le marché. Le débit d'information est de l'ordre de 1200 bits/seconde. L'interfaçage physique du MEA 8000 au TO7 est réalisé de manière classique.

**PAROLE<sup>1</sup>** peut être utilisée avec l'ensemble des langages informatiques existants actuellement ou à venir sur la gamme des micro-ordinateurs 8 bits THOMSON et permet d'adjoindre la parole à tous logiciels réalisés en BASIC, ASSEMBLEUR, LOGO, FORTH, PASCAL, LSE.

Le MEA 8000 peut être considéré par le système soit comme deux cases mémoires (mode mémoire), ou bien comme un périphérique (mode périphérique) en utilisant les commandes d'interruption. Le circuit employé en mode périphérique n'utilise au maximum, pour une locution continue, que 2 % du temps machine pour acquérir les données binaires. Ce mode permet d'intégrer la parole aux logiciels d'animation, sans en dégrader un des paramètres essentiels qu'est la vitesse d'exécution. La sortie audio du MEA 8000 est raccordée directement à l'entrée son des micros.

Afin de vous éviter de longues séances de recopie de listings, la plupart des exemples de ce chapitre, consacré aux matériels THOMSON, utilisent les codes vocaux de la cassette de présentation. Les codes vocaux se recopient, à partir de la cassette et sous BASIC, par l'instruction :

**LOADM"CASS:DEFPARL1.BIN",- &H677 pour TO7,TO7/70,TO9;**  
les codes vocaux sont alors implantés à partir \$A000 et

**LOADM"CASS:DEFPARL1.BIN",- &H677 pour MO5;**  
les codes vocaux sont alors implantés à partir \$8000.

---

<sup>1</sup> *Attention! PAROLE ne peut pas supporter la présence du modem Télétel (ils utilisent les mêmes adresses).*

# Réalisation matérielle

Pour fonctionner correctement, le MEA 8000 a besoin d'un certain nombre de signaux qui doivent être, dans notre cas, issus ou conçus à partir des signaux disponibles sur les ports d'extension des machines THOMSON. Dans la suite, nous allons "passer en revue" les accès (entrées et sorties) nécessaires au synthétiseur.

$\overline{CE}$ : Sélection du boîtier

Nous avons vu, dans les chapitres précédents, qu'en fait le MEA 8000 se comportait, vu du micro, comme deux emplacements consécutifs de mémoires.  $\overline{CE}$  est l'information, issue du décodeur, qui permet de sélectionner les deux adresses reconnaissables par le boîtier. Ces adresses ont été choisies au sommet des emplacements mémoires non affectés et réservés aux boîtiers d'extension.

Soit: \$E7FE et \$E7FF pour TO7, TO7/70, TO9  
et \$A7FE et \$A7FF pour MO5.

## Le décodeur

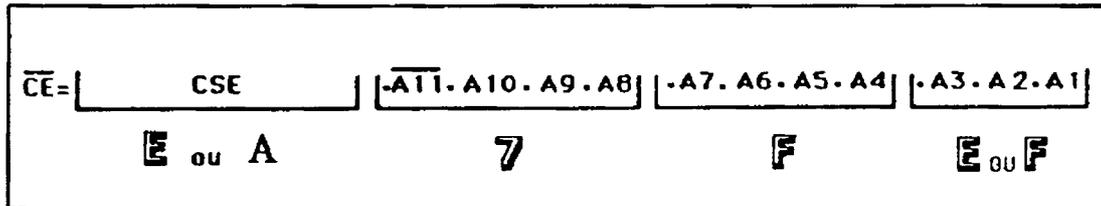
Le décodage des adresses doit répondre aux équations booléennes suivantes:

Pour TO7-T07/70-T09			
$\overline{CE} =$	$A_{15} \cdot A_{14} \cdot A_{13} \cdot \overline{A_{12}}$	$\overline{A_{11}} \cdot A_{10} \cdot A_9 \cdot A_8$	$\cdot A_7 \cdot A_6 \cdot A_5 \cdot A_4$
	$\cdot A_3 \cdot A_2 \cdot A_1$		
	<b>E</b>	<b>7</b>	<b>F</b>
			<b>E ou F</b>
Pour MO5			
$\overline{CE} =$	$A_{15} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot A_{12}$	$\overline{A_{11}} \cdot A_{10} \cdot A_9 \cdot A_8$	$\cdot A_7 \cdot A_6 \cdot A_5 \cdot A_4$
	$\cdot A_3 \cdot A_2 \cdot A_1$		
	<b>A</b>	<b>7</b>	<b>F</b>
			<b>E ou F</b>

La sortie CSE décode les adresses de poids forts, soit:

\$E000 à \$EFFF pour TO7, TO7/70, TO9  
et \$A000 à \$AFFF pour MO5.

En l'utilisant dans notre équation de décodage on obtient:



... que nous réalisons avec une porte NAND à 13 entrées (74LS133) et deux éléments d'un sextuple inverseur (74LS04).

**A0:** Sélection du registre de commande et du registre de données

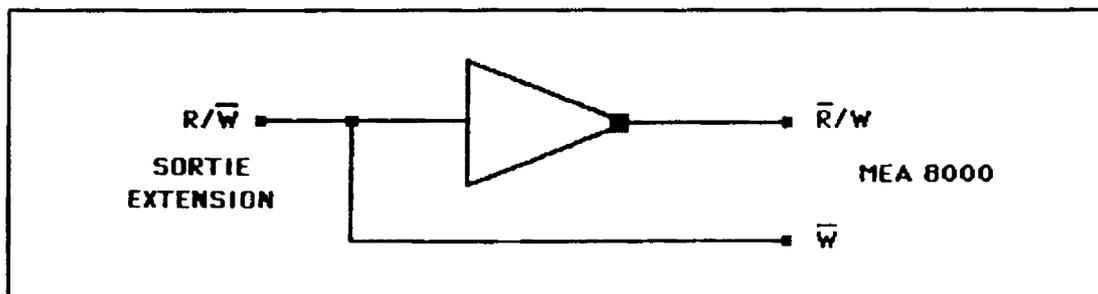
A0 est l'entrée qui permet de choisir le registre auquel on s'adresse.

A0	REGISTRE
1	REGISTRE DE COMMANDE (RCOM)
0	REGISTRE DE DONNEE (DONN)

L'entrée A0 est raccordée directement au bit de poids faible du bus d'adresse (A0).

**$\overline{R/W}$ :** Lecture ou écriture

La particularité du MEA 8000 est de disposer de deux bornes d'écriture ( $\overline{R/W}$  et  $\overline{W}$ ) afin d'être compatible avec la plupart des microprocesseurs du marché. Dans notre application (PAROLE), nous utilisons l'entrée  $\overline{R/W}$  après inversion, par une porte NON, du  $\overline{R/W}$  issu du 6809.  $\overline{W}$  est connectée à la masse. Mais le schéma suivant est aussi tout à fait admissible par le 6809.



**DATA:** Bus de données sur lequel transitent les données vocales et les informations du registre de contrôle. Comme les sorties du bus de données (D0 à D7) sont compatibles avec les entrées du MEA 8000, nous le raccorderons directement.

**REQ**: Demande d'interruption

$\overline{\text{REQ}}$  est la sortie de demande d'interruption du programme en cours par le MEA 8000. Nous avons choisi la possibilité, par strap, de la connecter à l'entrée  $\overline{\text{IRQ}}$  du 6809. Nous aurions pu utiliser également  $\overline{\text{FIRQ}}$ , mais nous avons préféré réserver  $\overline{\text{FIRQ}}$  pour une application plus prioritaire que la synthèse vocale. Nous verrons, dans l'étude du logiciel, que le choix de l'entrée  $\overline{\text{NMI}}$  (Non Masquable Interrupt) aurait été plus judicieux, mais cette entrée n'est pas disponible sur le port d'extension du MO5.

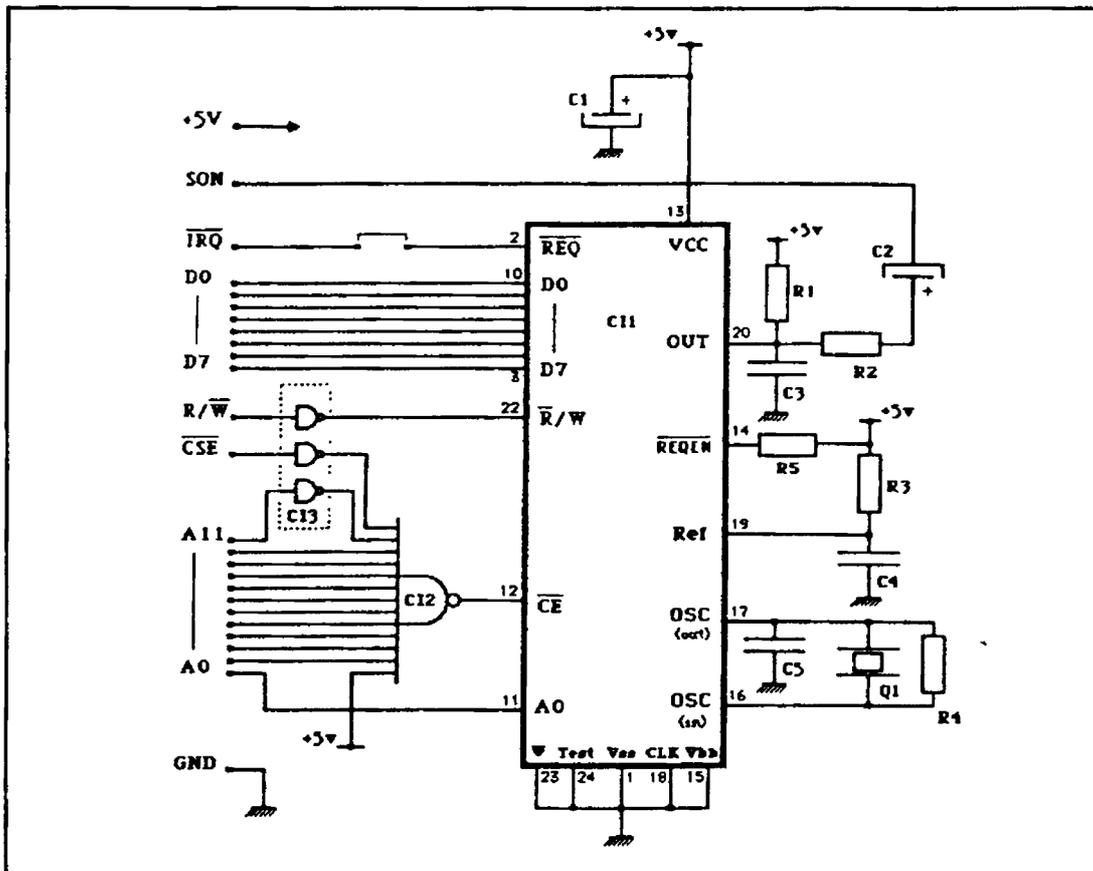
**REQUEN**: Entrée de validation de la sortie REQ

Nous n'utiliserons pas cette entrée dans nos applications, mais on doit la rendre inactive en la forçant au niveau haut par une résistance de 10 000 ohms connectée au 5V.

**OUT**: Sortie audio

La sortie son est connectée, à travers les éléments spécifiés par le constructeur, à la borne d'entrée son du port d'extension.

### Schéma général



## Connecteurs des ports d'extension

### SORTIE EXTENSION T07-T07/70-T09

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	GND	CP1	$\overline{E7C}$	$\overline{CSE}$	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	$\overline{CSD}$	$\overline{CSC}$	+5V
B	+12	GND	SON	R/V	E	D7	D6	D5	D4	D3	D2	D1	D0	$\overline{RST}$	$\overline{FIRQ}$	$\overline{NMI}$	$\overline{IRQ}$	Videq	-5V

### SORTIE EXTENSION M05

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	GND	CP1	$\overline{E7C}$	$\overline{CSE}$	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	$\overline{CSD}$	SYN	+5V
B	+12	GND	SON	R/V	E	D7	D6	D5	D4	D3	D2	D1	D0	$\overline{RST}$	$\overline{FIRQ}$	VCO	$\overline{IRQ}$	Videq	-5V

## Nomenclature

Boitier	1	Boitier de controleur de communication THOMSON CC 90-232	
Connecteur	1	SOCAPEX 254DF19 AYW8C7	
C13	1	74LS04	Sextuple inverseur
C12	1	74LS133	Porte NAND à 13 entrees
C11	1	MEA 8000 Version M 4244	Synthésiseur vocal
Q1	1	4 MHz	Quartz
C5	1	10 à 20 pF	Condensateur céramique
C4	2	10 nF	Condensateur céramique
C3	1	100 nF	Condensateur céramique
C2	1	10 $\mu$ F	Condensateur électrolytique 16V
C1	1	0,47 $\mu$ F	Condensateur tantale goutte 6V
R5	1	10 K $\Omega$	Résistance à couche carbone -1/4V - 10%
R4	1	1 M $\Omega$	Résistance à couche carbone -1/4V - 10%
R3	1	22 K $\Omega$	Résistance à couche carbone -1/4V - 10%
R2	1	680 $\Omega$	Résistance à couche carbone -1/4V - 10%
R1	1	470 $\Omega$	Résistance à couche carbone -1/4V - 10%
REPERE	Nbr	VALEURS	Désignation - Type

# Logiciel

---

Les machines THOMSON sont assez riches en langages. On compte, dans l'ordre de présentation de ce chapitre :

- l'ASSEMBLEUR
- le BASIC
- le FORTH
- le LOGO
- le LSE

Pour qu'un périphérique, ce qu'est en fait notre synthèse vocale, soit exploitable par le maximum d'utilisateurs avec le minimum d'efforts, il faut qu'il soit capable de fonctionner avec l'ensemble des langages implantés sur la machine. Ce qui implique des instructions spécifiques dans chaque langage évolué s'adressant à ce périphérique. Ce n'est évidemment pas notre situation. Toutefois, il est toujours possible d'élaborer des routines, implantables dans le programme utilisateur. C'est cette démarche que nous adopterons dans la suite de ce chapitre.

Dans toutes ces applications, autres que celles utilisant les phonèmes, nous emploierons, pour illustrer nos exemples, les codes vocaux de la cassette de présentation. Nous donnons, en annexe, le listing à l'intention des lecteurs qui souhaitent s'inspirer de ces routines pour écrire des logiciels pour d'autres machines ou qui ne disposent pas de la cassette de présentation pour faire une copie.

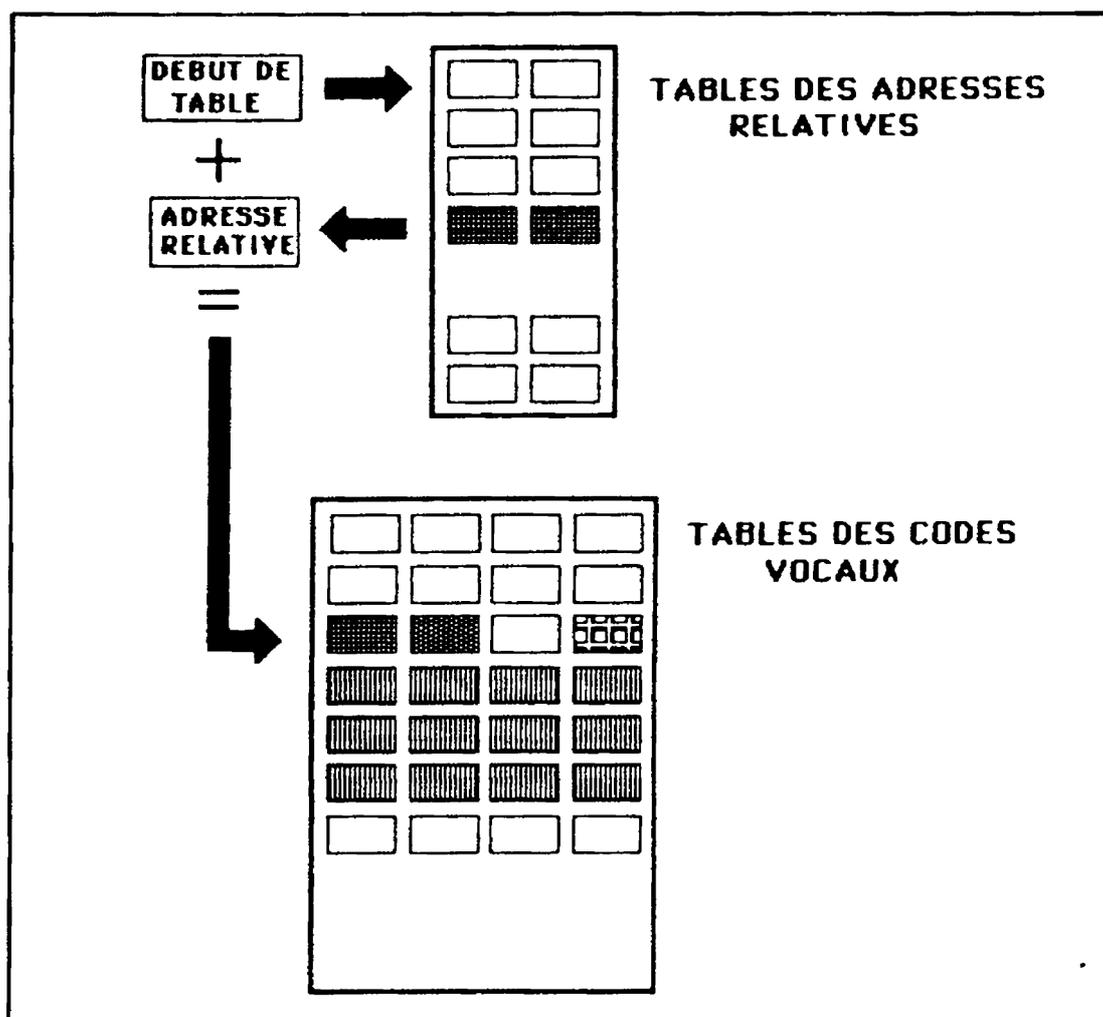
Notez : Le listing est présenté avec une implantation en 0000. Il convient d'ajouter \$A000 pour obtenir le listing TO7, TO7/70 ou TO9 et \$8000 pour obtenir le listing MO5.

# Utilisation sous ASSEMBLEUR

Nous avons vu, dans les chapitres précédents, que les codes vocaux sont une suite de paquets de 4 octets appelés trame. Une expression comporte un certain nombre de trames précédées d'un octet qui représente la hauteur initiale. Chaque représentation binaire d'une expression ou d'un mot commence par 4 octets qui donnent:

- la longueur de l'expression sur 16 bits (2 octets)
- la hauteur du son après l'expression ou 00
- la hauteur du son avant l'expression ou hauteur initiale

La valeur pointée dans la table des adresses relatives spécifie la grandeur qu'il convient d'ajouter à l'adresse du début de la table pour pointer sur le premier octet de l'expression.



En mode INACTIF (sortie silencieuse) le MEA 8000 positionne soit le bit 7 du registre de commande, soit la sortie  $\overline{REQ}$ , suivant le type de fonctionnement choisi (BIT 7 = 1,  $\overline{REQ}$  = 0). On peut dire, en quelque sorte, que le MEA 8000 "appelle" les octets dont il a besoin pour prononcer la prochaine trame. Après avoir reçu les octets nécessaires (5 pour la première trame et 4 pour les suivantes) le MEA 8000 repositionne les indicateurs jusqu'à la prochaine demande de données vocales.

Si après avoir prononcé la dernière trame, le MEA 8000 n'est pas "approvisionné" en codes vocaux, il considère que cette trame était la dernière de l'expression et repasse en mode INACTIF, s'il est configuré en mode ARRET- LENT, ou répète indéfiniment la dernière trame s'il est configuré en mode CONTINU.

Le programme ci-dessous, intitulé PARL, permet de faire prononcer par le synthétiseur une expression ou un mot dont le rang, dans la table des adresses relatives, est passé par l'octet VAL. DEBUT est le programme qui charge VAL et appelle PARL.

#### PARL (version TO7, TO7/70, TO9)

```

9FB3          ORG      $9FB3  Origine
              E7FE      DONN   EQU      $E7FE  Registre de donnees
              E7FF      PCOM   EQU      $E7FF  Registre de commande
              A000      DTAB   EQU      $A000  Debut de table
9FB3 01      VAL      FCB      1          Rang dans la table

9FB4 86      03      DEBUT  LDA      #3          Troisieme mot
9FB6 87      9FB3      STA      VAL          dans VAL
9FB9 8D      9FB0      JSR      PARL         Sous/Prog PARL
9FBC 3F      SWI

9FBD 34      57      PARL   PSHS   D,U,X,CC  Sauve le contexte
9FBF 56      9FB3      LDA      VAL          Compare le rang
* dans la table avec le nombre de mots
9FC2 81      15      CMPA   #21          21 mots dans la
* table
9FC4 22      21      BHI     FIN          Si Plus grand:FIN
9FC6 48      ASLA    2          2 octets par adresse
9FC7 CE      A000      LDJ     #DTAB      Debut de la table
9FCA EC      C6      LDD    A,U          Adresse relative
9FCC 33      CB      LEAU   D,U          Adresse absolue
9FCE C6      1A      LDB   #1A          Positionne le reg-
* istre de commande en modes:
* ARRET LENT et REQ inactif
9FD0 F7      E7FF      STB   RCOM
9FD3 AE      C1      LDX   ,U++         Longueur du mot
9FD5 33      41      LEAU  1,U          Sauter le 3em octet
9FD7 30      1D      LEAX  -3,X         Deja 3 octets de
* Passes
9FD9 E6      C0      RET   LDB   ,U+     Hauteur initiale
* ou donnees

```

9FD8	7D	E7FF	AR1	TST	RCOM	Test le bit 7 du
						* registre de commande
9FDE	2A	FB		BPL	AR1	Si Positif on
						Si negatif on
						* continue
9FE0	F7	E7FE		STB	DONN	Envoie de l'octet
						* demande
9FE3	30	1F		LEAX	-1,X	Decompte le nombre
						* d'octets
9FE5	26	F2		BNE	RET	Si Pas zero on
						* continue
9FE7	35	D7	FIN	PULS	D,U,X,CC,PC	Restitue le
						* contexte
		0000			END	

### PARL (version MO5)

7FB3				ORG	\$7FB3	Origine
		A7FE	DONN	EQU	\$A7FE	Registre de donnees
		A7FF	RCOM	EQU	\$A7FF	Registre de commande
		8000	DTAB	EQU	\$8000	Debut de table
7FB3	01		VAL	FCB	1	Rang dans la table
7FB4	86	03	DEBUT	LDA	#3	Troisieme mot
7FB6	B7	7FB3		STA	VAL	dans VAL
7FB9	BD	7FBF		JSR	PARL	Sous/Prog PARL
7FBC	BD	B000		STOP		
7FBF	34	57	PARL	PSHS	D,U,X,CC	Sauve le contexte
7FC1	B6	7FB3		LDA	VAL	Compare le rang
						* dans la table avec le nombre de mots
7FC4	81	15		CMPS	#21	21 mots dans la
						* table
7FC6	22	21		BHI	FIN	Si Plus grand FIN
7FC8	48			ASLA		2 octets Par adresse
7FC9	CE	8000		LDU	#DTAB	Debut de la table
7FCC	EC	C6		LDU	A,U	Adresse relative
7FCE	33	CB		LEAU	D,U	Adresse absolue
7FD0	C6	1A		LDB	#1A	Positionne le reg-
						* gistre de commande en modes'
						* ARRRET LENT et REQ inactif
7FD2	F7	A7FF		STB	RCOM	
7FD5	AE	C1		LDX	,U++	Longueur du mot
7FD7	33	41		LEAU	1,U	Saute le 3em octet
7FD9	30	1D		LEAX	-3,X	Deja 3 octets de
						* Passes
7FDB	E6	C0		RET	LDB	,U+
						Hauteur initiale
						* ou donnees
7FDD	7D	A7FF	AR1	TST	RCOM	Test le bit 7 du
						* registre de commande
7FE0	2A	FB		BPL	AR1	Si Positif on
						Si negatif on
						* continue

7FE2	F7	A7FE		STB	DONN	Envoie de l'octet
			* demande			
7FE5	30	1F		LEAX	-1,X	Decompte le nombre
			* d'octets			
7FE7	26	F2		BNE	RET	Si Pas zero .on
			* continue			
7FE9	35	D7	FIN	PULS	D,U,X,CC,PC	Restitue le
			* contexte			
		0000		END		

Pour obtenir une locution continue, vous pouvez dans DEBUT incrémenter VAL et boucler jusqu'au dernier mot stocké en mémoire.

### Sur TO7, TO7/70, TO9

9FB4	06	00	DEBUT	LDA	#0	Mot zero
9FB6	B7	9FB3	AR2	STA	VAL	dans VAL
9FB9	BD	9FC2		JSR	PARL	Sous/Prog PARL
9FBC	4C			INCA		Le mot suivant
9FBD	81	15		CMPA	#21T	Est la fin ?
9FBF	26	F5		BNE	AR2	Non:Alors on bou-
			* cle			
9FC1	3F			SWI		Oui

### Sur MO5

7FB4	86	00	DEBUT	LDA	#0	Mot zero
7FB6	B7	7FB3	AR2	STA	VAL	dans VAL
7FB9	BD	7FC4		JSR	PARL	Sous/Prog PARL
7FBC	4C			INCA		Le mot suivant
7FBD	81	15		CMPA	#21T	Est la fin ?
7FBF	26	F5		BNE	AR2	Non:Alors on bou-
			* cle			
7FC1	BD	B000		STOP		Oui

## Utilisation de l'IRQ

En utilisant le MEA 8000, comme le montrent les exemples précédents, le microprocesseur est constamment occupé et ne peut être utilisé pour d'autres tâches. L'occupation permanente du système peut être gênante quand les phrases prononcées sont un peu longues. D'autant plus que le 6809 passe une grande partie du temps à boucler en attendant que le synthétiseur lui demande des informations.

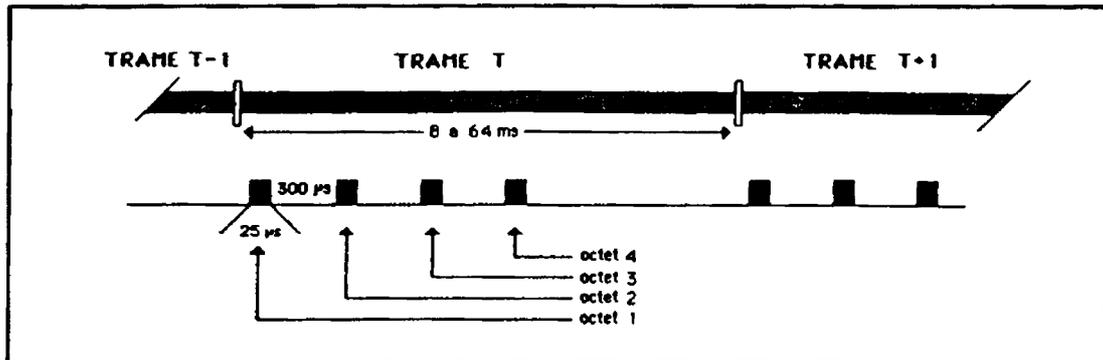
**Faisons un petit calcul:** Prenons une trame de durée minimum 8 ms. Il faut, dans le programme précédent, environ 25 cycles d'horloge par tour de boucle pour charger un octet. Comme on doit charger 4 octets par trame, il faudra 100 cycles d'horloge de une microseconde, soit 100 microsecondes pour charger une trame, soit un taux d'utilisation de :

$100/8000 = 0.0125$       1.25% pour une trame de 8 ms

0.625% pour une trame de 16 ms

et                              0.16% pour une trame de 64 ms

... quel gâchis!!!



L'utilisation de la demande d'interruption IRQ du 6809 va permettre de diminuer considérablement le temps d'occupation du microprocesseur pour servir la synthèse vocale. Dans ce mode de fonctionnement, le MEA 8000 n'utilise le système que le temps nécessaire au chargement des codes vocaux, soit environ 2 % du temps machine car, en plus de la boucle que nous avons vue dans le programme précédent, il faut tenir compte du temps de sauvegarde et de restitution des contextes du programme principal et du programme d'interruption. L'application en mode interruption ou mode périphérique que nous proposons est constituée de 3 modules et du programme principal permettant l'appel des ces routines.

**DEBUT:** C'est le programme principal qui appelle les deux sous-programmes PARINT et LANC.

**PARINT:** Permet d'initialiser les registres LONG et PTEUR qui représentent respectivement la longueur des octets restants à charger et la position du prochain octet.

**LANC:** Positionne le MEA 8000 en mode d'interruption ( $\overline{REQ}$  actif) et lance la prononciation du mot ou de l'expression.

**IRQ:** Est la routine d'interruption, que le 6809 viendra exécuter quand il recevra un niveau bas sur son entrée  $\overline{IRQ}$  connectée à la sortie  $\overline{REQ}$  du MEA 8000.

# LISTING T07-T07/70-T09

```

9F00          6021   IRQ   EQU   $9F00
                * de l'IRQ
                E7FE   DONN  EQU   $E7FE
                E7FF   RCOM  EQU   $E7FF
                A000   DTAB  EQU   $A000   Debut de la table
9F00 01        VAL   FCB   1
9F01          LONG  RMB   2           Longueur restante
9F03          PTEUR RMB   2           Pointeur courant

9F05 8E       9F2A   DEBUT LDX   #PIRQ   Charge l'adresse
                * du sous-programme d'interruption dans
                * le registre d'aiguillage de l'IRQ
9F08 BF       6021   STX   IRQ
9F0B 86       03     LDA   #03T   Rang du mot a Pro-
                * noncer
9F0D 8D       9F52   ENCOR JSR   PARINT  Initialisation des
                * registres et Pointeur necessaires au
                * sous Programme d'interruption
9F10 8D       9F49   JSR   LANC   Lance la Parole
9F13 8E       FFFF   LDX   #$FFFF
9F16 30       1F     AR5   LEAX  -1,X
9F18 26       FC     BNE   AR5
9F1A 3F       SWI
9F1B 8E       FFFF   LDX   #$FFFF   3 boucles Pour
                * eviter que l'execution du Prog Prin-
                * cipale soit terminee avant la fin de
                * la Parole
9F1E 30       1F     ARX   LEAX  -1,X
9F20 26       FC     BNE   ARX
9F22 8E       FFFF   LDX   #$FFFF
9F25 30       1F     ARY   LEAX  -1,X
9F27 26       FC     BNE   ARY
9F29 3F       SWI

                *S/P D'INTERRUPTION:
9F2A 34       54     PIRQ  PSHS  U,X,B   Sauvegarde les
                * registres utilises dans le S/P PIRQ
9F2C FE       9F03   LDU   PTEUR  Charge le Pointeur
9F2F BE       9F01   LDX   LONG   Charge la longueur
9F32 E6       C0     LDB   ,U+    Charge l'octet a
                * transferer ....
9F34 F7       E7FE   STB   DONN   ... dans le MEA
9F37 30       1F     LEAX  -1,X   Decompte un octet
9F39 26       05     BNE   ARE   Teste la fin du mot
                * Non. Alors on saute en ARE
9F3B 86       1A     ARR   LDA   #$1A   Oui. On desactive
                * la sortie REQ du MEA Pour eviter qu'il
                * envoie constamment des IRQ
9F3D B7       E7FF   STA   RCOM
9F40 FF       9F03   ARE   STU   PTEUR  Sauve le Pointeur
9F43 BF       9F01   STX   LONG   sauve la longueur

```

```

9F46 35 54 PULS U,X,B Restitue l'etat
* des registres utilises
9F48 38 RTI Retour d'interruption

```

\* S/P DE LANCEMENT:

```

9F49 34 44 LANC PSHS B,U Sauve
9F4B C6 1F LDB ##1F Active la sortie
* REQ et mode CONTINU (facultatif)
9F4D F7 E7FF STB RCOM
9F50 35 C4 PULS B,U,PC Restitue et retour

```

\* S/P D'INITIALISATION:

```

9F52 34 57 PARINT PSHS D,U,X,CC Sauve
9F54 B6 9F00 LDA VAL Charge VAL
9F57 81 15 CMPA ##21 Compare
9F59 22 14 BHI FIN
9F5B 48 ASLA 2 octets
9F5C CE A000 LDU #DTAB Debut de la table
9F5F EC C6 LDD A,U Adresse relative
9F61 33 CB LEAU D,U Adresse absolue
9F63 AE C1 LDX ,U++ Longueur du mot
9F65 33 41 LEAU 1,U Saute le 3em octet
9F67 FF 9F03 STU PTEUR Initialise le
* Pointeur
9F6A 30 1D LEAX -3,X Deja 3 octets de
* Passes
9F6C BF 9F01 STX LONG Charge le registre
* de la longueur courante
9F6F 35 D7 FIN PULS D,U,X,CC,PC Restitue
0000 END

```

# LISTING M05

```

7F00          ORG      $7F00
              2064    IRQ    EQU    $2064      Aiguillage de
              * de l'IRQ
              A7FE    DONN   EQU    $A7FE
              A7FF    RCOM   EQU    $A7FF
              8000    DTAB   EQU    $8000      Debut de la table
7F00 01      VAL    FCB    1
7F01          LONG   RMB    2      Longueur restante
7F03          PTEUR  RMB    2      Pointeur courant

7F05 8E      7F2B    DEBUT  LDX    #PIRQ      Charge l'adresse
              * du sous-programme d'interruption dans
              * le registre d'aiguillage de l'IRQ
7F08 8F      2064    STX    IRQ
7F0B 86      03      LDA    #03T      Rang du mot a Pro-
              * noncer
7F0D BD      7F53    ENCOR  JSR    PARINT  Initialisation des
              * registres et Pointeur necessaires au
              * sous Programme d'interruption
7F10 BD      7F4A    JSR    LANC      Lance la Parole
7F13 8E      FFFF    LDX    #$FFFF
7F16 30      1F      AR5    LEAX   -1,X
7F18 26      FC      BNE    AR5
7F1A 8E      FFFF    LDX    #$FFFF      3 boucles Pour
              * eviter que l'execution du Prog Prin-
              * cipale soit terminee avant la fin de
              * la Parole
7F1D 30      1F      ARX    LEAX   -1,X
7F1F 26      FC      BNE    ARX
7F21 8E      FFFF    LDX    #$FFFF
7F24 30      1F      ARY    LEAX   -1,X
7F26 26      FC      BNE    ARY
7F28 BD      B000    STOP

              *S/P D'INTERRUPTION:

7F2B 34      56      PIRQ   PSHS   U,X,D      Sauvegarde les
              * registres utilises dans le S/P PIRQ
7F2D FE      7F03    LDU    PTEUR  Charge le Pointeur
7F30 BE      7F01    LDX    LONG   Charge la longueur
7F33 E6      C0      LDB    ,U+    Charge l'octet a
              * transferer ....
7F35 F7      A7FE    STB    DONN   ... dans le MEA
7F38 30      1F      LEAX   -1,X   Decompte un octet
7F3A 26      05      BNE    ARE    Teste la fin du mot
              * Non. Alors on saute en ARE
7F3C 86      1A      ARA    LDA    #$1A   Oui. On desactive
              * la sortie REQ du MEA Pour eviter qu'il
              * envoie constamment des IRQ
7F3E B7      A7FF    STA    RCOM
7F41 FF      7F03    ARE    STU    PTEUR  Sauve le Pointeur
7F44 BF      7F01    STX    LONG   sauve la longueur

```

```

7F47 35 56          PULS  U,X,D  Restitue l'etat
* des registres utilises
7F49 3B          RTI      Retour d'interruption

* S/P DE LANCEMENT:

7F4A 34 44        LANC  PSHS  B,U    Sauve
7F4C C6 1F        LDB   #51F  Active la sortie
* REQ et mode CONTINU (facultatif)
7F4E F7 A7FF     STB   RCOM
7F51 35 C4        PULS  B,U,PC Restitue et retour

* S/P D'INITIALISATION:

7F53 34 57        PARINT PSHS  D,U,X,CC Sauve
7F55 B6 7F00     LDA   VAL   Charge VAL
7F58 81 15        CMPA  #&21  Compare
7F5A 22 14        BHI   FIN
7F5C 48          ASLA          2 octets
7F5D CE 8000     LDU   #DTAB Debut de la table
7F60 EC C6        LDD   A,U   Adresse relative
7F62 33 CB        LEAU  D,U   Adresse absolue
7F64 AE C1        LDX   ,U++  Longueur du mot
7F66 33 41        LEAU  1,U   Saute le 3em octet
7F68 FF 7F03     STU   PTEUR  Initialise le
* Pointeur
7F6B 30 1D        LEAX  -3,X  DeJa 3 octets de
* Passes
7F6D BF 7F01     STX   LONG  Charge le registre
* de la longueur courante
7F70 35 D7        FIN   PULS  D,U,X,CC,PC Restitue
      0000     END

```

**Deux choses à la fois:** Nous venons de voir, dans le dernier exemple, que votre ordinateur était capable de faire deux choses à la fois. Exécuter le programme principal, avec appel éventuel des sous-programmes, et parler. Mais attention! Certaines routines des moniteurs, tant TO7, TO7/70, TO9 que MO5, sont protégées contre les interruptions IRQ. Si vous utilisez ces routines dans votre programme, le MEA 8000 devra attendre la fin de l'exécution de la routine en cours pour être servi par l'interruption qu'il a générée. Résultat: le synthétiseur va avoir un débit de parole réduit et repassera en mode INACTIF s'il est programmé en état ARRET LENT ou répètera la dernière trame s'il est en CONTINU.

Votre ordinateur va se comporter comme s'il avait passé les quelques heures qui précèdent au bar du coin!!!

C'est pour cette raison que l'emploi du NMI était plus judicieux. Bien que, si le constructeur a interdit certaines routines des moniteurs aux interruptions qu'il pouvait masquer, c'est probablement parce qu'il avait une raison logicielle tout aussi valable pour NMI. Alors, que faire ?

La solution est de créer vos propres routines interruptibles en utilisant éventuellement des modules ou routines non protégés du moniteur. Les "Secrets du TO7 et du MO5" (Editions TOTEK) devraient vous être d'un grand secours pour cette tâche. Afin de vous éviter quelques "plantages magistraux" nous donnons la liste des principales routines en indiquant leur protection éventuelle.

T07-T07/70			NOMS	M05		
FIRQ	IRQ	CODES		CODES	IRQ	FIRQ
NI	NI	\$E803	PUTC	\$02	NI	NI
I	I	\$E806	GETC	\$0A	I	I
I	I	\$E809	KTST	\$0C	I	I
I	I	\$E80C	DRAW	\$0E	I	I
I	I	\$E80E	PLOT	\$10	I	I
I	NI	\$E812	RSCOS	\$24	NI	NI
NI	NI	\$E815	K7COS	\$20	NI	NI
NI	NI	\$E818	GETL	\$18	NI	NI
I	I	\$E81B	LPIN	\$16	I	I
NI	NI	\$E81E	NOTE	\$1E	NI	NI
I	I	\$E821	GETP	\$14	I	I
NI	NI	\$E824	GETS	\$1A	NI	NI
I	I	\$E827	JOYS	\$1C	I	I
NI	NI	\$E82A	DKCOS	\$26	NI	NI
I	I	\$E833	CHPL	\$12	I	I

NI: non interruptible (protégé); I: interruptible (non protégé)

Comme application de ce que nous venons de voir, nous donnons une réalisation montrant que votre ordinateur est effectivement capable de parler et d'écrire au même moment. La structure logicielle est un peu modifiée par rapport à l'exemple précédent. Le but de cette réalisation est de montrer comment utiliser une routine non protégée du moniteur et comment synchroniser la parole et l'image.

LISTING TO7-TO7/70-TO9

```

9F00          ORG      $9F00  Origine
          E803  PUTC  EQU  $E803  Routine d'affichage
          * non interruptible
          603B  COLOR EQU  $603B
          E833  CHPL  EQU  $E833  Routine d'affichage
          * interruptible
          6041  CHDRAW EQU  $6041
          6021  IRQ   EQU  $6021  Remuillage de
          * de l'IRQ
          E7FE  DONN  EQU  $E7FE
          E7FF  RCOM  EQU  $E7FF
          A000  DTAB  EQU  $A000  Debut de la table
9F00 01      VAL  FCB  1
9F01          LONG  RMB  2  Longueur restante
9F03          PTEUR RMB  2  Pointeur courant
9F05          DRAP  RMB  1  Drapeau de fin de
          * Parole
9F06 4D 4F 4E 20  TEXT  FCC  /MON TO7 FAIT DEU/
9F0A 54 4F 37 20
9F0E 46 41 49 54
9F12 20 44 45 55
9F16 58 20 43 48          FCC  /X CHOSES A LA FOIS/
9F1A 4F 53 45 53
9F1E 20 41 20 4C
9F22 41 20 46 4F
9F26 49 53
9F28 00          FCB  00
9F29 1F 53 54 1B  TEXT1 FCB  $1F,$53,$54,$1B,$53,$1B
9F2D 53 1B
9F2F 44          FCB  $44
9F30 43 27 45 53          FCC  /C'EST BIEN VRAI !!/
9F34 54 20 42 49
9F38 45 4E 20 56
9F3C 52 41 49 20
9F40 21 21
9F42 1F 56 41          FCB  $1F,$56,$41
9F45 00          FCB  00

          *PROGRAMME PRINCIPAL
9F46 86 00      DEBUT LDA  #0  Pointe sur le Pre-
          * mier a Prononcer
9F48 B7 9F00    ENCOR STA  VAL  Charge le registre
9F4B BD 9FD9    JSR   PARINT Initialisation des
          * registres et Pointeur necessaires au
          * sous Programme d'interruption
9F4E BD 9FC4    JSR   LANC  Lance la Parole
9F51 BD 9F6F    AZE   JSR   ECRI  Ecrit le texte
9F54 7D 9F05    TST   DRAP  Teste si la Pronon-
          * ciation du dernier mot est terminee
9F57 26 F8      BNE   AZE   Non.Alors on repart
          * Pour un nouveau cycle d'écriture
9F59 B6 9F00    LDA   VAL  Oui.On incremente
9F5C 4C          * VAL Pour annoncer le Prochain mot
          INCA

```

9F5D	81	14	CMPA	#20T	Est la fin du mes-
					* sage Parle
9F5F	26	E7	BNE	ENCOR	Non. Alors on conti-
					* nue Pour le mot suivant
9F61	8E	9F29	LDX	#TEXT1	Oui. Alors on envoie
					* un message ecrit Pour verifier que le
					* systeme a bien retrouve toute son
					* integrite. Ce troncon de Programme n'
					* est utile que Pour la demonstration.
9F64	E6	80	ARR	LDB	,X+ Un octet de la chaine
9F66	5D			TSTB	La fin ?
9F67	27	05		BEQ	END Oui
9F69	BD	E803		JSR	PUTC Non. On envoie l'octet
9F6C	20	F6		BRA	ARR On continue
9F6E	3F		END	SWI	

\*S/P ECRITURE (interruPtible)

9F6F	34	77	ECRI	PSHS	X,D,Y,U,CC Sauve
9F71	8E	0001		LDX	#1 Pour Positionner la
					* ligne et la colonne de debut de texte
9F74	108E	000C		LDY	#12T
9F78	CE	9F06		LDU	#TEXT Pointe sur le texte
9F7B	E6	C0	BOU	LDB	,U+ Octet de la chaine
9F7D	5D			TSTB	Dernier ?
9F7E	27	18		BEQ	TERM Oui. Alors termine
9F80	F7	6041		STB	CHDRAW Non. L'octet est ch-
					* arge dans le registre de Passage en
					* mode caractere graphique
9F83	B6	603B		LDA	COLOR Pour modifier les
					* couleurs de fond et de forme a chaque
					* Passage
9F86	88	09		ADDA	##9 Fond et forme
9F88	81	C0		CMPA	##C0 Fin du cycle couleur?
9F8A	22	02		BHI	AVG Non.
9F8C	86	C1		LDA	##C1 Oui on reinitiali-
					* sation du cycle couleur
9F8E	B7	603B	AVG	STA	COLOR Charge COLOR
9F91	BD	E833		JSR	CHPL Ecrit le code ASCII
					* Passe Par CHDRAW
9F94	30	01		LEAX	1,X Plus 1 sur la colonne
9F96	20	E3		BRA	BOU On boucle
9F98	35	F7	TERM	PULS	X,D,Y,U,CC,PC Restitue

\*S/P D'INTERRUPTION:

9F9A	34	77	PIRQ	PSHS	U,X,Y,D,CC Sauvegarde les
					* registres utilises dans le S/P PIRQ
9F9C	FE	9F03		LDU	PTEUR Charge le Pointeur
9F9F	BE	9F01		LDX	LONG Charge la longueur
9FA2	E6	C0		LDB	,U+ Charge l'octet a
					* transferer ....
9FA4	F7	E7FE		STB	DONN ... dans le MER
9FA7	30	1F		LEAX	-1,X Decompte un octet

9FA9	26	10	BNE	ARE	Teste la fin du mot
			*		Non. Alors on saute en ARE
9FAB	86	1A	ARR	LDA	##1A Oui. On desactive
			*		la sortie REQ du MEA Pour eviter qu'il
			*		envoie constamment des IRQ
9FAD	B7	E7FF		STA	RCOM
9FB0	7F	9F05		CLR	DRAP
					Pour indiquer que
			*		le mot est fini
9FB3	108E	E844		LDY	##E844 Charge l'adresse
			*		initiale d'interruption moniteur dans
			*		le registre d'aiguillage de l'IRQ
9FB7	10BF	6021		STY	IRQ
9FBB	FF	9F03	ARE	STU	PTEUR
					Sauve le Pointeur
9FBE	BF	9F01		STX	LONG
					sauve la longueur
9FC1	35	77		PULS	U,X,Y,D,CC Restitue l'etat
			*		des registres utilises
9FC3	3B			RTI	Retour d'interruption

\* S/P DE LANCEMENT:

9FC4	34	54	LANC	PSHS	B,U,X	Sauve
9FC6	8E	9F9A		LDX	#PIRQ	Charge l'adresse
			*			du sous-Programme d'interruption dans
			*			le registre d'aiguillage de l'IRQ
9FC9	BF	6021		STX	IRQ	
9FCC	7F	9F05		CLR	DRAP	Initialise a 1
			*			l'indicateur de fin de mot
9FCF	7C	9F05		INC	DRAP	
9FD2	C6	1F		LDB	##1F	Active la sortie
			*			REQ et mode CONTINU (facultatif)
9FD4	F7	E7FF		STB	RCOM	
9FD7	35	D4		PULS	B,U,X,PC	Restitue et RTS

\* S/P D'INITIALISATION:

9FD9	34	57	PARINT	PSHS	D,U,X,CC	Sauve
9FDB	B6	9F00		LDA	VAL	Charge VAL
9FDE	81	15		CMPA	##21	Compare
9FE0	22	14		BHI	FIN	
9FE2	48			ASLA		2 octets
9FE3	CE	A000		LDU	#DTAB	Debut de la table
9FE6	EC	C6		LDD	A,U	Adresse relative
9FE8	33	CB		LEAU	D,U	Adresse absolue
9FEA	AE	C1		LDX	,U++	Longueur du mot
9FEC	33	41		LEAU	1,U	Saute le 3em octet
9FEE	FF	9F03		STU	PTEUR	Initialise le
			*			Pointeur
9FF1	30	1D		LEAX	-3,X	Deja 3 octets de
			*			Passes
9FF3	BF	9F01		STX	LONG	Charge le registre
			*			de la longueur courante
9FF6	35	D7	FIN	PULS	D,U,X,CC,PC	Restitue
		0000		END		

# LISTING M05

```

7F00          ORG      $7F00  Origine
              0002  PUTC  EQU    $02  Routine d'affichage
              * non interruptible
              202B  COLOR EQU    $202B
              0012  CHPL  EQU    $12   Routine d'affichage
              * interruptible
              2036  CHDRAW EQU   $2036
              2064  IRQ   EQU    $2064   Aiguillage de
              * de l'IRQ
              A7FE  DONN  EQU    $A7FE
              A7FF  RCOM  EQU    $A7FF
              8000  DTAB  EQU    $8000   Debut de la table
7F00 01      VAL  FCB    1
7F01          LONG  RMB    2   Longueur restante
7F03          PTEUR RMB    2   Pointeur courant
7F05          DRAP  RMB    1   Drapeau de fin de
              * Parole
7F06 4D 4F 4E 20 TEXT  FCC    /MON M05 FAIT DEU/
7F0A 4D 4F 35 20
7F0E 46 41 49 54
7F12 20 44 45 55
7F16 58 20 43 48          FCC    /X CHOSES A LA FOIS/
7F1A 4F 53 45 53
7F1E 20 41 20 4C
7F22 41 20 46 4F
7F26 49 53
7F28 00          FCB    00
7F29 1F 53 54 1B TEXT1 FCB    $1F,$53,$54,$1B,$53,$1B
7F2D 53 1B
7F2F 44 1B 7A          FCB    $44,$1B,$7A
7F32 43 27 45 53          FCC
7F36 54 20 42 49          /C'EST BIEN VRAI !!/
7F3A 45 4E 20 56
7F3E 52 41 49 20
7F42 21 21
7F44 1F 56 41          FCB    $1F,$56,$41
7F47 00          FCB    00

              *PROGRAMME PRINCIPAL
7F48 86 00          DEBUT LDA    #0   Pointe sur le Pre-
              * mier a Prononcer
7F4A B7 7F00        ENCOR STA    VAL   Charge le registre
7F4D BD 7FDB        JSR    PARINT Initialisation des
              * registres et Pointeur necessaires au
              * sous Programme d'interruption
7F50 BD 7FC6        JSR    LANC   Lance la Parole
7F53 BD 7F72        AZE    JSR    ECRI  Ecrit le texte
7F56 7D 7F05        TST    DRAP   Teste si la Pronon-
              * ciation du dernier mot est terminee
7F59 26 F8          BNE    AZE    Non.Alors on repart
              * Pour un nouveau cycle d'écriture
7F5B B6 7F00        LDA    VAL   Oui,On incremente
              * VAL Pour annoncer le Prochain mot
7F5E 4C          INCA

```

```

7F5F 81 14          CMPA  #20T  Est la fin du mes-
* sage parle
7F61 26 E7          BNE   ENCOR  Non.Alors on conti-
* nue Pour le mot suivant
7F63 8E 7F29        LDX   #TEXT1 Oui.Alors on envoie
* un message ecrit Pour verifier que le
* systeme a bien retrouve toute son
* integrite.Ce troncon de Programme n'
* est utile que Pour la demonstration.
7F66 E6 80          ARR   LDB   ,X+  Un octet de la chaine
7F68 5D              TSTB                La fin ?
7F69 27 04          BEQ   END    Oui
7F6B 3F 02          CALL  PUTC  Non.On envoie l'octet
7F6D 20 F7          BRA   ARR   On continue
7F6F BD B000        END   STOP

```

**\*S/P ECRITURE (interruPtible)**

```

7F72 34 77          ECRI  PSHS  X,D,Y,U,CC Sauve
7F74 8E 0001        LDX   #1    Pour Positionner la
* ligne et la colonne de debut de texte
7F77 108E 000C      LDY   #12T
7F7B CE 7F06        LDU   #TEXT  Pointe sur le texte
7F7E E6 C0          BOU   LDB   ,U+  Octet de la chaine
7F80 5D              TSTB                Dernier ?
7F81 27 17          BEQ   TERM   Oui.Alors termine
7F83 F7 2036        STB   CHDRAW Non.L'octet est ch-
* arge dans le registre de Passage en
* mode caractere graphique
7F86 B6 202B        LDA   COLOR  Pour modifier les
* couleurs de fond et de forme a chaque
* Passage
7F89 8B 09          ADDA  #$9    Fond et forme
7F8B 81 C0          CMPA  #$C0  Fin du cycle couleur?
7F8D 22 02          BHI   AVG    Non.
7F8F 86 C1          LDA   #$C1  Oui on reinitiali-
* sation du cycle couleur
7F91 B7 202B        AVG   STA   COLOR  Charge COLOR
7F94 3F 12          CALL  CHPL  Ecrit le code ASCII
* Passe Par CHDRAW
7F96 30 01          LEAX  1,X   Plus 1 sur la colonne
7F98 20 E4          BRA   BOU   On boucle
7F9A 35 F7          TERM  PULS  X,D,Y,U,CC,PC Restitue

```

**\*S/P D'INTERRUPTION:**

```

7F9C 34 77          PIRQ  PSHS  U,X,Y,D,CC Sauvegarde les
* registres utilises dans le S/P PIRQ
7F9E FE 7F03        LDU   PTEUR  Charge le Pointeur
7FA1 BE 7F01        LDX   LONG   Charge la longueur
7FA4 E6 C0          LDB   ,U+   Charge l'octet a
* transferer ....
7FA6 F7 A7FE        STB   DONN   ... dans le MEA
7FA9 30 1F          LEAX  -1,X  DecompTe un octet

```

7FAB 26	10	BNE	ARE	Teste la fin du mot
		*		Non. Alors on saute en ARE
7FAD 86	1A	ARA	LDA	##1A Oui. On desactive
		*		la sortie REQ du MEA Pour eviter qu'il
		*		envoie constamment des IRQ
7FAF 87	A7FF	STA	RCOM	
7FB2 7F	7F05	CLR	DRAP	Pour indiquer que
		*		le mot est fini
7FB5 108E	F0AD	LDY	##F0AD	Charge l'adresse
		*		initiale d'interruption moniteur dans
		*		le registre d'aiguillage de l'IRQ
7FB9 10BF	2064	STY	IRQ	
7FBD FF	7F03	ARE	STU	PTEUR Sauve le Pointeur
7FC0 BF	7F01		STX	LONG sauve la longueur
7FC3 35	77		PULS	U,X,Y,D,CC Restitue l'etat
		*		des registres utilises
7FC5 3E			RTI	Retour d'interruption

\* S/P DE LANCEMENT :

7FC6 34	54	LANC	PSHS	B,U,X Sauve
7FC8 8E	7F9C		LDX	##PIRQ Charge l'adresse
		*		du sous-Programme d'interruption dans
		*		le registre d'aiguillage de l'IRQ
7FCB BF	2064		STX	IRQ
7FCE 7F	7F05		CLR	DRAP Initialise a 1
		*		l'indicateur de fin de mot
7FD1 7C	7F05		INC	DRAP
7FD4 C6	1F		LDB	##1F Active la sortie
		*		REQ et mode CONTINU (facultatif)
7FD6 F7	A7FF		STB	RCOM
7FD9 35	D4		PULS	B,U,X,PC Restitue et RTS

\* S/P D'INITIALISATION :

7FDB 34	57	PARINT	PSHS	D,U,X,CC Sauve
7FDD B6	7F00		LDA	VAL Charge VAL
7FE0 81	15		CMPI	##21 Compare
7FE2 22	14		BHI	FIN
7FE4 48			ASLA	2 octets
7FE5 CE	8000		LDU	##DTAB Debut de la table
7FE8 EC	C6		LDD	A,U Adresse relative
7FEA 33	C8		LEAU	D,U Adresse absolue
7FEC AE	C1		LDX	,U++ Longueur du mot
7FEE 33	41		LEAU	1,U Saute le 3em octet
7FF0 FF	7F03		STU	PTEUR Initialise le
		*		Pointeur
7FF3 30	1D		LEAX	-3,X Deja 3 octets de
		*		Passes
7FF5 BF	7F01		STX	LONG Charge le registre
		*		de la longueur courante
7FF8 35	D7	FIN	PULS	D,U,X,CC,PC Restitue
	0000		END	

**VAL:** est chargé avec le rang du premier mot à prononcer (0 dans le listing).

**PARINT:** permet d'initialiser les registres PTEUR et LONG avec les valeurs relatives au mot spécifié par VAL.

**LANC:** en plus des fonctions précédentes, ce sous-programme permet maintenant de charger le registre d'aiguillage de l'IRQ afin de conserver toute l'intégrité du système même après l'exécution de PARINT. De plus, LANC initialise à 1 un drapeau (DRAP) qui permet de vérifier à chaque instant si l'on est dans une phase d'expression afin de détecter la fin du mot.

**ECRI:** est le sous-programme qui permet d'écrire le texte à l'écran en utilisant une routine non protégée du moniteur (CHPL). A la sortie d'ECRI, on teste DRAP afin de savoir si la prononciation est terminée. Si oui, on relance en bouclant sur ENCOR après avoir incrémenté VAL. La boucle s'arrête quand le dernier mot est trouvé. Notez qu'il est toujours possible de connaître le statut de la synthèse.

**DRAP:** représente l'état: 1 = parle, 0 = silence.

**VAL:** représente le mot en cours

**LONG:** la longueur du mot restant à prononcer

En utilisant de manière judicieuse ces paramètres il vous sera facile de synchroniser l'image et la parole, ce que nous avons fait dans le dernier exemple en utilisant DRAP.

## **Les phonèmes sous ASSEMBLEUR**

Nous ne reviendrons pas sur la définition des phonèmes et leur rôle en phonétique. Ces éléments ont déjà été traités au début de cet ouvrage. Nous nous bornerons sous ASSEMBLEUR à faire prononcer à notre ordinateur une chaîne de caractères définie par une directive d'assemblage FCC (Définition d'une constante de chaîne). Nous utiliserons pour cela les codes des phonèmes dits 3.3, répertoriés en annexe. A chaque phonème nous avons attribué une touche du clavier aussi près que possible de la prononciation, afin de se rapprocher le plus possible du texte.

**Tableau d'attribution des phonèmes**

SON	EXEMPLE	REPRESENTATION (touche)
A	pApA,rAt	a
E	IE,	e
I	navIre,Ilmace	i
O,AU	platEAU,O,OH	o
U	lUne,prUne	u
AI,EI,E,ET	gAI, orEille	&
é	Ecole,EclatER	*
EU	hEUrEUx	E
OU	lOUp	w
AN,EN	nathAN	A
IN,UN,AIN,EIN	lapIN,sAINt	I
ON	lONg,SON	O
OI	joIe,dOIgt	W
B	Boule,Bar	b
D	Domino	d
G	GUerre,Gare	g
J	Jaune,GIbier	j
K	Carte,QUinte	k
L	LiLa,Lumière	l
M	MaMan,Miel	m
N	Navire	n
P	PaPa,Pirate	p
R1	Roule,Rat	R
R2	lourd,bonjour	r
S	Salade,Cédic	s
T	Tas,Tomate	t
V	Vélo,Valise	v
Z	Zéro,vaIse	z
CH	CHarme,éCHelle	\$
GN	éparGNE,montaGNard	N
AIL	travAIL	@
EIL	vermEIL	*
EUIL	dEUIL,OEIL	%
IEN	BIEN,rIEN	<
OIN	pOINt,IOIN	>
O	pORt, OR	0 (zéro)
SILENCE 32ms		;
SILENCE 64ms		point
Particularités:	CHIEN se prononce FAMILLE " " ORDINATEUR T07 BASIC	\$il ou\$< famie ou famiee ORDinater t;o;s*t bazik

Pour cette réalisation, nous avons tenu le raisonnement suivant: Puisque chaque phonème a un incrément de pitch constant pour chacune des trames le composant, la hauteur de son (en Hertz) en sortie est égale à la valeur qu'elle avait à l'entrée. Alors pourquoi répéter la hauteur initiale à chaque phonème?

Chaque fois que nous pointerons, dans notre application, sur les quatre premiers octets d'un phonème, nous ne tiendrons compte que de l'octet définissant les poids faibles de la longueur. La hauteur initiale est envoyée au synthétiseur sitôt après son initialisation en mode ARRET-LENT et REQ inactif (code \$1A). Ainsi, les codes vocaux s'enchaîneront les uns à la suite des autres comme s'ils faisaient partie de la même expression.

### LISTING TO7-TO7/70-TO9

```

                E7FE    DONN    EQU    $E7FE
                E7FF    RCOM    EQU    $E7FF
A000                                ORG    $A000    Origine

                                *TABLE DES ADRESSES RELATIVES
                                *   avec les caracteres correspondants
A000 61 02 74 65  TABL    FCB    'a,$02,$74,'e,$02,$88
A004 02 88
A006 69 03 80 6F    FCB    'i,$03,$80,'o,$02,$D8
A00A 02 D8
A00C 75 02 B0 26    FCB    'u,$02,$B0,'&,$02,$C4
A010 02 C4
A012 2A 03 DC 45    FCB    '*,$03,$DC,'E,$03,$08
A016 03 08
A018 77 03 68 41    FCB    'w,$03,$68,'A,$03,$3C
A01C 03 3C
A01E 49 03 54 4F    FCB    'I,$03,$54,'O,$02,$EC
A022 02 EC
A024 57 03 1C 62    FCB    'W,$03,$1C,'b,$03,$AC
A028 03 AC
A02A 64 03 94 66    FCB    'd,$03,$94,'f,$00,$7C
A02E 00 7C
A030 67 00 8C 6A    FCB    'g,$00,$8C,'J,$00,$A4
A034 00 A4
A036 6B 00 BC 6C    FCB    'k,$00,$BC,'l,$00,$D4
A03A 00 D4
A03C 6D 00 E4 6E    FCB    'm,$00,$E4,'n,$00,$F4
A040 00 F4
A042 70 01 08 52    FCB    'P,$01,$08,'R,$01,$24
A046 01 24
A048 72 03 F0 73    FCB    'r,$03,$F0,'s,$01,$3C
A04C 01 3C
A04E 74 03 C4 76    FCB    't,$03,$C4,'v,$01,$48
A052 01 48
A054 7A 01 5C 24    FCB    'z,$01,$5C,'$, $01,$68
A058 01 68

```

```

A05A 4E 01 74 40      FCB      'N,$01,$74,'@,$01,$98
A05E 01 98
A060 23 01 C4 25      FCB      '#,$01,$C4,'%,$01,$E8
A064 01 E8
A066 3C 02 C1 3E      FCB      '<,$02,$C1,'>,$02,$44
A06A 02 44
A06C 30 02 9C 2D      FCB      '0,$02,$9C,'-,$03,$78
A070 03 78
A072 20 02 64 2E      FCB      ',$02,$64,'.,$02,$6C
A076 02 6C
A078                      ORG      TABL+$78

```

\*Ici la table des Phonemes 3.3 a partir  
\* de l'adresse 0078

```

A078 FFFF FF00      FDB      $FFFF,$FF00,$0010,$3C3C
A07C 0010 3C3C

```

\* etc etc.....

```

A400 73 2A 64 69      *Chaine de caracteres a Prononcer
A404 6B 2E 6E 61      CHAI    FCC      /s*dik.natA.Pa.Pwr.fOs*/
A408 74 41 2E 70
A40C 61 2E 70 77
A410 72 2E 66 4F
A414 73 2A

```

\*Traduction: Cedic Nathan Pas Pour foncer

```

A416 108E A400      * Programme appelant PARCH
DEBUT LDY      #CHAI    Pointe sur le Pre-
* octet de la chaine
A41A BD      A41E      JSR      PARCH    Sous-Programme de
* traitement des chaines de "Phonemes"
A41D 3F      SWI
A41E 34      3F      PARCH    FSHS    Y,X,DP,B,A,CC Sauvegarde
A420 86      1A      LDA      #S1A    Initialise le MEA
* en ARRET-LENT et REQ inactif
A422 B7      E7FF      STA      RCOM
A425 86      3C      LDA      #S3C    Envoie la hauteur
* initiale commune a chaque Phoneme
A427 B7      E7FE      STA      DONN
A42A 31      3F      LEAY    -1,Y    Pour faciliter la
* boucle
A42C 31      21      BOU1    LEAY    1,Y
A42E 8E      A000      LDX      #TABL   Pointe sur TABL
A431 6D      A4      BOU0    TST      ,Y     Teste si le caract-
* ere courant est:00
A433 27      2A      BEQ     FIN     Oui.Alors c'etait
* la fin de la chaine
A435 A6      84      LDA      ,X     Non.Alors on com-
* pare avec les caracteres specifiques
* dans la table des adresses relatives

```

```

A437 A1  A4          CMPA  ,Y
A439 27  08          BEQ   TROU  Egalite.Alors trouve
A43B 81  FF          CMPA  #$FF  A-t-on a la fin de
* la table ?
A43D 27  ED          BEQ   BOU1  Oui.Alors le carac-
* tere de la chaine n'est Pas specifie
* dans la table et on repart lire le
* Prochain caractere de la chaine
A43F 30  03          LEAX  3,X  Non.Alors on incre-
* mente le Pointeur de 3 Pour faire une
* nouvelle comparaison avec le carac-
* tere suivant de la table
A441 20  EE          BRA   BOU0
A443 EC  01          TROU  LDD   1,X  Adresse relative
A445 C3  A001        ADDD  #TABL+1 Additionne TABL+1
*Pour Pointer sur l'octet de Poids faible
* representant la longueur du Phoneme
A448 1F  01          TFR   D,X  Pointeur dans X
A44A A6  84          LDA   ,X  Longueur du Phoneme
A44C 30  03          LEAX  3,X  Pointe sur l'octet
* de la Premiere trame
A44E 80  04          SUBA  #4   DeJa 4 octets
A450 E6  80          BOU3  LDB  ,X+  Octet des trames
A452 7D  E7FF        BOU2  TST  RCOM  Le MEAS000 reclame-t
*-il...des octets ....des octets.....
A455 2A  FB          BPL   BOU2  Non.Alors on attend
* qu'il reclame
A457 F7  E7FE        STB   DONN  Oui.Alors on lui
* donne un octet
A45A 4A           DECA   On decompte les octets
A45B 26  F3          BNE   BOU3  Fin du Phoneme.Non.
* Alors on continue Pour l'octet suivant
A45D 20  CD          BRA   BOU1  Oui,C'est la fin
A45F 35  BF          FIN   PULS  Y,X,DP,B,A,CC,PC Restitue
A461 3F           SWI

```

# LISTING MO5

```

          A7FE   DONN   EQU   $A7FE
          A7FF   RCOM   EQU   $A7FF
7F00     ORG     $7F00   Origine
  
```

\*TABLE DES ADRESSES RELATIVES  
 \* avec les caracteres correspondants

```

7F00 61 02 74 65  TABL  FCB   'a,$02,$74,'e,$02,$88
7F04 02 88
7F06 69 03 80 6F      FCB   'i,$03,$80,'o,$02,$D8
7F0A 02 D8
7F0C 75 02 B0 26      FCB   'u,$02,$B0,'&,$02,$C4
7F10 02 C4
7F12 2A 03 DC 45      FCB   '*,$03,$DC,'E,$03,$08
7F16 03 08
7F18 77 03 68 41      FCB   'w,$03,$68,'A,$03,$3C
7F1C 03 3C
7F1E 49 03 54 4F      FCB   'I,$03,$54,'O,$02,$EC
7F22 02 EC
7F24 57 03 1C 62      FCB   'W,$03,$1C,'b,$03,$AC
7F28 03 AC
7F2A 64 03 94 66      FCB   'd,$03,$94,'f,$00,$7C
7F2E 00 7C
7F30 67 00 8C 6A      FCB   'g,$00,$8C,'J,$00,$A4
7F34 00 A4
7F36 6B 00 BC 6C      FCB   'k,$00,$BC,'l,$00,$D4
7F3A 00 D4
7F3C 6D 00 E4 6E      FCB   'm,$00,$E4,'n,$00,$F4
7F40 00 F4
7F42 70 01 08 52      FCB   'p,$01,$08,'R,$01,$24
7F46 01 24
7F48 72 03 F0 73      FCB   'r,$03,$F0,'s,$01,$3C
7F4C 01 3C
7F4E 74 03 C4 76      FCB   't,$03,$C4,'v,$01,$48
7F52 01 48
7F54 7A 01 5C 24      FCB   'z,$01,$5C,'$, $01,$68
7F58 01 68
7F5A 4E 01 74 40      FCB   'N,$01,$74,'@,$01,$98
7F5E 01 98
7F60 23 01 C4 25      FCB   '#,$01,$C4,'%,$01,$E8
7F64 01 E8
7F66 3C 02 C1 3E      FCB   '<,$02,$1C,'>,$02,$44
7F6A 02 44
7F6C 30 02 9C 2D      FCB   '0,$02,$9C,'-,$03,$78
7F70 03 78
7F72 20 02 64 2E      FCB   ',$02,$64,'.,$02,$6C
7F76 02 6C
7F78      ORG     TABL+$78
  
```

```

* Ici la table des Phonemes 3.3 a Partir
* de l'adresse 0078
7F78 FFFF FF00      FDB      $FFFF,$FF00,$0010,$3C3C
7F7C 0010 3C3C

* etc etc.....

```

```

*Chaine de caracteres a Prononcer
8300 73 2A 64 69    CHAI   FCC      /s*dik.natA.Pa.Pwr.f0s*/
8304 6B 2E 6E 61
8308 74 41 2E 70
830C 61 2E 70 77
8310 72 2E 66 4F
8314 73 2A

```

\*Traduction: Cedic Nathan Pas Pour foncer

```

* Programme appelant PARCH
8316 108E 8300    DEBUT LDY   #CHAI   Pointe sur le Pre-
* octet de la chaine
831A BD      8320    JSR   PARCH   Sous-Programme de
* traitement des chaines de "Phonemes"
831D BD      B000    STOP

8320 34      3F      PARCH PSHS   Y,X,DP,B,A,CC Sauvegarde
8322 86      1A      LDA   #$1A   Initialise le MEA
* en ARRET-LENT et REQ inactif
8324 B7      A7FF    STA   RCOM
8327 86      3C      LDA   #$3C   Envoie la hauteur
* initiale commune a chaque Phoneme
8329 B7      A7FE    STA   DONN
832C 31      3F      LEAY -1,Y   Pour faciliter la
* boucle
832E 31      21      BOU1  LEAY  1,Y
8330 8E      7F00    LDX   #TABL  Pointe sur TABL
8333 6D      A4      BOU0  TST   ,Y   Teste si le caract-
* ere courant est:00

8335 27      2A      BEQ   FIN    Oui.Alors c'etait
* la fin de la chaine
8337 A6      84      LDA   ,X     Non.Alors on com-
* pare avec les caracteres specifies
* dans la table des adresses relatives
8339 A1      A4      CMPA  ,Y
833B 27      08      BEQ   TROU   Egalite.Alors trouve
833D 81      FF      CMPA  #$FF   A-t-on a la fin de
* la table ?
833F 27      ED      BEQ   BOU1   Oui.Alors le caract-
* ere de la chaine n'est Pas specifie
* dans la table et on repart lire le
* Prochain caractere de la chaine
8341 30      03      LEAX  3,X   Non.Alors on incre-
* mente le Pointeur de 3 Pour faire une
* nouvelle comparaison avec le caract-
* ere suivant de la table

```

8343	20	EE		BRA	BOU0	
8345	EC	01	TROU	LDD	1,X	Adresse relative
8347	C3	7F01		RDDD	#TABL+1	Additionne TABL+1
						*Pour Pointer sur l'octet de Poids faible
						* representant la longueur du Phoneme
834A	1F	01		TFR	D,X	Pointeur dans X
834C	A6	84		LDA	,X	Longueur du Phoneme
834E	30	03		LEAX	3,X	Pointe sur l'octet
						* de la Premiere trame
8350	80	04		SUBA	#4	Deja 4 octets
8352	E6	80	BOU3	LDB	,X+	Octet des trames
8354	7D	A7FF	BOU2	TST	RCDM	Le MEA8000 reclame-t
						*-il...des octets ....des octets.....
8357	2A	FB		BPL	BOU2	Non.Alors on attend
						* qu'il reclame
8359	F7	A7FE		STB	DONN	Oui.Alors on lui
						* donne un octet
835C	4A			DECA		On decompte les octets
835D	26	F3		BNE	BOU3	Fin du Phoneme.Non.
						* Alors on continue pour l'octet suivant
835F	20	CD		BRA	BOU1	Oui,C'est la fin
8361	35	BF	FIN	PULS	Y,X,DP,B,A,CC,PC	Restitue

# Utilisation sous BASIC

---

Le BASIC est le plus répandu des langages informatiques. On se devait, dans cet ouvrage, de donner quelques exemples d'emploi de la synthèse vocale sous contrôle de ce langage. Nous vous proposons 3 programmes utilisant les mêmes codes vocaux que les applications "tournant" sous ASSEMBLEUR et qui peuvent être classés en fonction de l'utilisation qu'ils font ou non des sous-programmes binaires :

1. Tout BASIC,
2. Minimum de binaire et
3. Totalité binaire.

## Le tout BASIC

Nous prévenons tout de suite le lecteur que le programme "tout BASIC" est une réalisation qui n'a d'intérêt que de mettre en évidence la lenteur du BASIC et de justifier son inadaptation pour l'application projetée. Nous vous proposons en quelque sorte "d'écouter la lenteur du BASIC". Les codes vocaux sont traités par chaînes de caractères, solution qui n'améliore pas la vitesse d'exécution.

### LISTING TO7-TO7/70-TO9

```
10 CLEAR,&HA000
20 A=1
30 POKE&HE7FF,&H1E
40 P$=HEX$(PEEK(&HA000+(A*2)))
50 Q$=HEX$(PEEK(&HA001+(A*2)))
60 H$=P$+Q$
70 GOSUB180
80 P=D+( &HA000 )
90 P$=HEX$(PEEK(P))
100 Q$=HEX$(PEEK(P+1))
110 H$=P$+Q$:GOSUB180:L=D-4:P=P+4
120 POKE&HE7FE,(PEEK(P-1))
130 WAIT&HE7FF,&H80
140 FOR J=0 TO 3:POKE&HE7FE,(PEEK(P+J))
150 NEXT J
160 P=P+J:L=L-J:IF L=0 THEN170 ELSE 130
170 STOP
180 '*** sous programme hex->deci ***
190 D=0:E=0
200 FORX=1TOLEN(H$)
210 A$=MID$(H$,LEN(H$)-X+1,1)
220 IFASC(A$)<48 OR ASC(A$)>70THENE=-1
230 IFASC(A$)<65 ANDASC(A$)>57THENE=-1
```

```

240 IFA$="A" THEN A%=10:GOTO310
250 IFA$="B" THEN A%=11:GOTO310
260 IFA$="C" THEN A%=12:GOTO310
270 IFA$="D" THEN A%=13:GOTO310
280 IFA$="E" THEN A%=14:GOTO310
290 IFA$="F" THEN A%=15:GOTO310
300 A%=VAL(A$)
310 D=D+A%*CINT(16^(X-1))
320 NEXT X:RETURN
330 END

```

## LISTING MO5

```

10 CLEAR,&H8000
20 A=1
30 POKE&HA7FF,&H1E
40 P$=HEX$(PEEK(&H8000+(A*2)))
50 Q$=HEX$(PEEK(&HA001+(A*2)))
60 H$=P$+Q$
70 GOSUB180
80 P=D+( &H8000)
90 P$=HEX$(PEEK(P))
100 Q$=HEX$(PEEK(P+1))
110 H$=P$+Q$:GOSUB180:L=D-4:P=P+4
120 POKE&HA7FE,(PEEK(P-1))
130 WAIT&HA7FF,&H80
140 FOR J=0 TO 3:POKE&HA7FE,(PEEK(P+J))
150 NEXT J
160 P=P+J:L=L-J:IF L=0 THEN170 ELSE 130
170 STOP

```

**Notez :** les lignes de 180 à 320 sont identiques pour TO7 et MO5.

— En ligne 20, A est la variable qui, initialisée à 1, représente le rang du mot à prononcer.

— En ligne 30, on force le registre de commande du MEA à \$1B. Valeur correspondant au mode CONTINU et à la sortie REQ inactive. Compte tenu de la faible vitesse d'exécution du BASIC, le mode CONTINU est obligatoire car il y a un risque que le MEA repasse dans l'état INACTIF si les codes vocaux de la prochaine trame ne lui sont pas présentés avant la fin de la trame en cours. L'octet suivant serait alors interprété comme la hauteur initiale d'un nouveau mot alors qu'il est le premier octet d'une trame. Bonjour cafouillages et bruits infâmes !!

— Lignes 40, 50 et 60, on crée une chaîne de caractères avec les deux octets pointés par le début de la table à laquelle on ajoute le produit par deux du rang du mot pointé. Nous avons vu qu'il fallait deux octets pour spécifier une adresse relative dans la table.

— Par GOSUB 180, on appelle le sous-programme de conversion d'une chaîne de caractères en nombre décimal. Au retour de GOSUB, la variable D représente la valeur décimale de l'adresse relative. Le calcul de l'adresse absolue se fait en ajoutant (ligne 80) le début de la table, à la valeur de l'adresse relative. La variable P est affectée au pointeur.

— Aux lignes 90, 100 et 110 avec le même processus que nous avons vu plus haut, on vient acquérir dans D la valeur des deux premiers octets pointés, spécifiant la longueur de l'expression. Puisque l'on retranche quatre octets à la longueur pour tenir compte des octets du début de l'expression, il faut ajouter quatre octets au pointeur pour être "en phase".

Le POKE &HE7FE, (PEEK(P - 1)) en ligne 120 ou POKE &HA7FF, (PEEK(P - 1)) pour MO5 permet de charger la hauteur initiale dans le registre de données du MEA 8000. L'instruction WAIT en ligne 130 fait un ET entre le registre de contrôle du MEA et la valeur \$80, ce qui a pour conséquence de suspendre l'exécution du programme tant que le résultat de l'opération booléenne n'est pas égal à 1. La suite du programme ne sera donc exécutée qu'après l'accord du MEA 8000 qui marque son autorisation en mettant le bit 7 du registre de contrôle à 1.

L'instruction WAIT est insérée dans une boucle de 4 passages (J = 0 TO 3) qui permet d'envoyer les codes vocaux au MEA par salves de quatre octets, en attendant à chaque boucle que le synthétiseur autorise le transfert. Tous les quatre octets, on fait les comptes (ligne 160) et on teste si L, qui représente la longueur restante, est à zéro. Dans l'affirmative, c'est la fin ; dans le cas contraire, on repart pour une séquence de 4 octets.

L'exécution du programme s'obtient par RUN. Alors, qu'en pensez-vous?... Le BASIC, c'est bien lent!!!! N'avez vous pas l'impression que votre ordinateur est retourné, de nouveau, comme avec les routines moniteur non interruptibles, faire un petit séjour au café du coin ? A moins que vous souhaitiez faire la démonstration que votre ordinateur est bien fatigué, pour vous en faire offrir un autre, par exemple, ce programme "tout BASIC" est à proscrire.

## Minimum de binaire

Le BASIC a l'avantage de la simplicité mais cette facilité se "paye" en vitesse d'exécution. L'idée du programme que nous vous proposons maintenant est de conserver la simplicité du BASIC tout en utilisant, là où c'est nécessaire, la rapidité du langage machine (binaire). Pour simplifier, la partie du programme écrite en langage machine est spécifiée en DATA dans le programme BASIC.

## LISTING TO7-TO7/70-TO9

```
10 CLEAR,&H9FE4
20 FOR A=&HE8 TO &HFF
30 READ D:POKE&H9F00+A,D:NEXTA
40 DATA &H34,&H55,&HFE,&H9F,&HE6,&HBE,&H9F,&HE4,&HE6,&HCO,&H
7D,&HE7,&HFF,&H2A,&HFB,&HF7,&HE7,&HFE,&H30,&H1F,&H26,&HF2,&H
35,&HDS
50 A=2 'Rang du mot à prononcer
60 POKE&HE7FF,&H1A
70 P=PEEK(&HA000+(A*2))
80 Q=PEEK(&HA001+(A*2))
90 P=0+(P*256)+(&HA000)
100 L=PEEK (P)
110 L1=PEEK (P+1)
120 L=L1+(L*256)
130 L=L-3:P=P+3
140 POKE&H9FE6,INT (P/256)
150 POKE&H9FE7,P-INT (P/256)*256
160 POKE&H9FE4,INT (L/256)
170 POKE&H9FE5,L-INT (L/256)*256
180 EXEC &H9FE8
190 STOP
200 END
```

## LISTING MO5

```
10 CLEAR,&H7FE4
20 FOR A=&HE8 TO &HFF
30 READ D:POKE&H7F00+A,D:NEXTA
40 DATA &H34,&H55,&HFE,&H7F,&HE6,&HBE,&H7F,&HE4,&HE6,&HCO,&H
7D,&HA7,&HFF,&H2A,&HFB,&HF7,&HA7,&HFE,&H30,&H1F,&H26,&HF2,&H
35,&HDS
50 A=2 'Rang du mot à prononcer
60 POKE&HA7FF,&H1A
70 P=PEEK(&H8000+(A*2))
80 Q=PEEK(&H8001+(A*2))
90 P=0+(P*256)+(&H8000)
100 L=PEEK (P)
110 L1=PEEK (P+1)
120 L=L1+(L*256)
130 L=L-3:P=P+3
140 POKE&H7FE6,INT (P/256)
150 POKE&H7FE7,P-INT (P/256)*256
160 POKE&H7FE4,INT (L/256)
170 POKE&H7FE5,L-INT (L/256)*256
180 EXEC &H7FE8
190 STOP
200 END
```

Contrairement à l'application précédente, les codes vocaux sont traités comme des variables, ce qui "allège" un peu la structure du programme. La ligne 10 est un CLEAR pour éviter que le BASIC vienne "écraser" le binaire de l'application et les codes vocaux implantés à partir de l'adresse \$A000 pour TO7 ou \$8000 pour MO5.

Les lignes 20, 30 et 40 permettent d'écrire en mémoire les codes binaires spécifiés en DATA dans le programme. Le binaire du programme est implanté à partir de l'adresse \$9FE8 pour TO7 et \$7FE8 pour MO5. A est la variable qui donne le rang du mot à prononcer.

A la ligne 60, le registre de commande est initialisé à \$1A correspondant au mode ARRET-LENT et REQ inactif. Cette fois, comme on est sûr que le MEA ne risque pas de repasser en mode INACTIF au cours d'un mot, le mode CONTINU ne s'impose pas.

Le pointeur est calculé aux lignes 70, 80 et 90 tandis que la longueur de l'expression est acquise aux lignes 100, 110 et 120. Remarquez qu'il convient d'affecter d'un coefficient 256 la valeur de l'octet représentant les poids forts de la longueur et du pointeur.

La ligne 130 permet d'ajuster la longueur et le pointeur avant de les charger dans les registres 16 bits utilisés par le binaire. Le pointeur est chargé en \$9FE6 pour TO7 et \$7FE6 pour MO5. La longueur est chargée en \$9FF4 pour TO7 et \$7FE4 pour MO5. Les poids forts de ces paramètres sont obtenus en divisant par 256 la valeur puis en chargeant la partie entière du quotient dans l'octet correspondant. Les poids faibles s'obtiennent par soustraction de la partie entière calculée précédemment et affectée d'un coefficient 256, à la valeur d'origine.

La ligne 180 permet de lancer l'exécution de programme binaire. L'ajout des lignes 50, 182, 185 et 188 permet de prononcer l'ensemble du texte.

```
50 FOR A=0 TO 20 'nombre de mots dans la table
182 FOR T=0 TO 500
185 NEXT T
188 NEXT A
```

La boucle en 182 introduit une temporisation entre chaque expression. Elle peut, bien entendu, être modifiée suivant le souhait de l'utilisateur. De même, la boucle en ligne 50 peut être également modifiée. Nous vous suggérons, pour vous amuser un peu, de faire prononcer les mots par ordre décroissant ou bien suivant une loi fonction de votre inspiration.

## Totalité binaire

Ce troisième programme est une variante du précédent. Il reprend la même structure. La partie écrite précédemment en BASIC est maintenant réalisée en binaire. Le BASIC n'a plus qu'un paramètre d'entrée à passer au binaire (le rang du mot) tandis qu'il en fallait deux (pointeur et longueur) dans la version précédente.

## LISTING TO7-TO7/70-TO9

```
10 CLEAR,&H9FD3
20 RESTORE50
30 FOR A=&HD4 TO &HFF
40 READ D:POKE&H9F00+A,D:NEXTA
50 DATA &H34,&H57,&HB6,&H9F,&HD3,&HB1,&H15,&H22,&H21,&H4B,&H
CE,&HA0,&H00,&HEC,&HC6,&H33,&HCB,&HC6,&H1A,&HF7,&HE7,&HFF,&H
AE,&HC1,&H33,&H41,&H30,&H1D,&HE6,&HC0,&H7D,&HE7,&HFF,&H2A,&H
FB,&HF7,&HE7,&HFE,&H30,&H1F,&H26,&HF2,&H35,&HD7
60 FOR M=0 TO 20
70 POKE&H9FD3,M
80 EXEC&H9FD4
90 FOR T=0 TO 500
100 NEXT T
110 NEXT M
120 STOP
```

## LISTING MO5

```
10 CLEAR,&H7FD3
20 RESTORE50
30 FOR A=&HD4 TO &HFF
40 READ D:POKE&H7F00+A,D:NEXTA
50 DATA &H34,&H57,&HB6,&H7F,&HD3,&HB1,&H15,&H22,&H21,&H4B,&H
CE,&H80,&H00,&HEC,&HC6,&H33,&HCB,&HC6,&H1A,&HF7,&HA7,&HFF,&H
AE,&HC1,&H33,&H41,&H30,&H1D,&HE6,&HC0,&H7D,&HA7,&HFF,&H2A,&H
FB,&HF7,&HA7,&HFE,&H30,&H1F,&H26,&HF2,&H35,&HD7
60 FOR M=0 TO 20
70 POKE&H7FD3,M
80 EXEC&H7FD4
90 FOR T=0 TO 500
100 NEXT T
110 NEXT M
120 STOP
```

Le rang du mot à prononcer (M) est chargé par le BASIC à l'adresse \$9FD3 sur TO7 et \$7FD3 sur MO5.

## Les phonèmes sous BASIC

Nous avons vu, sous ASSEMBLEUR, qu'il était assez facile de traiter les phonèmes sous forme de chaîne de caractères. Sous BASIC, cette facilité est encore accrue car le BASIC met à notre disposition des instructions spécifiques aux chaînes de caractères. Nous pourrions donc concaténer des expressions phonétiques entre elles, en fabriquer de nouvelles à partir d'autres, etc.

De plus, le BASIC dispose d'instructions que nous apprécierons beaucoup (DEFUSR et USR), qui vont nous permettre de passer les chaînes de caractères élaborées sous BASIC comme paramètres d'entrée du sous-programme binaire.

Les deux programmes que nous vous proposons vont permettre de composer des mots, à partir du clavier, immédiatement prononçables par votre ordinateur. La touche **ESPACE** commande la répétition du dernier mot prononcé. Les codes des phonèmes utilisés sont ceux répertoriés 3.3 en annexe, et que nous avons déjà utilisés sous ASSEMBLEUR. L'affectation des phonèmes aux touches du clavier est conforme au tableau page 91 .

## VERSION T07-T07/70-T09

```

10 CLS
20 SCREEN2,0,0
30 CLEAR,&HA000
40 RESTORE70
50 FOR A=&H02 TO &H5B
60 READ D:POKE&HA400+A,D:NEXTA
70 DATA &H34,&H7F,&HB1,&H03,&H26,&H51,&HA6,&H80,&H27,&H4D,&H
B7,&HA4,&H00,&H10,&HAE,&HB4,&HB6,&H1A,&HB7,&HE7,&HFF,&HB6,&H
3C,&HB7,&HE7,&HFE,&HA6,&HAC,&HB7,&HA4,&H01,&HBE,&HA0,&H00,&H
A6,&HB4,&HB1,&HFF,&H27,&H2E,&HB1,&HA4,&H01,&H27,&H04,&H30,&H
03,&H20,&HF1
80 DATA &HEC,&H01,&HC3,&H40,&H00,&H1F,&H01,&H36,&H20,&H10,&H
AE,&HB4,&H30,&H04,&H31,&H3C,&HA6,&H80,&H7D,&HE7,&HFF,&H2A,&H
FB,&HB7,&HE7,&HFE,&H31,&H3F,&H26,&HF2,&H37,&H20,&H7A,&HA4,&H
00,&H26,&HC4,&H12,&H35,&H7F,&H39
90 LOADM"PHONEME"
100 DEFUSR0=&HA402
110 LOCATE 0,10
120 INPUT"mot ", A$ :IF A$<>" " THEN CLS ELSE A$=D$
130 D$=A$:LOCATE 0,0:PRINTD$
140 B$=USR0(A$)
150 GOTO 110

```

Le programme binaire de traitement des codes vocaux est donné en DATA afin d'en faciliter l'analyse et la modification éventuelle. Ce programme est assez voisin de celui que nous avons étudié sous ASSEMBLEUR, sous réserve qu'il reçoive des paramètres de type 3 (chaînes) du BASIC. Nous rappelons les règles de passage de paramètres entre le BASIC et les sous-programmes binaires. Le registre A contient le type du paramètre passé (3 dans notre cas). Le registre 16 bits X pointe sur le premier octet de la zone d'échange (descripteur) dont la longueur est égale au type. Dans notre cas type: 3; zone d'échange: 3 octets. Le premier octet donne la longueur de la chaîne passée et les deux autres octets l'adresse du premier octet de la chaîne.

L'instruction LOADM"PHONEME" permet de charger en mémoire les codes vocaux et la table des adresses relatives. DEFUSR0 = &HA402

définit l'adresse d'exécution du sous-programme binaire implanté de l'adresse \$A400 à l'adresse \$A45B. Chaque caractère spécifiant un phonème est rentré au clavier et affecté, dans l'ordre d'entrée, à la chaîne A\$. Un test permet de vérifier si cette chaîne est vide. Après l'égalité de A\$ et D\$, le contenu de la chaîne est affiché en haut de l'écran.

En ligne 140, le sous-programme binaire est exécuté avec A\$ comme paramètre d'entrée. S'il y avait un paramètre de même type en retour du sous-programme, il serait affecté à la chaîne B\$.

Le GO TO permet de recommencer la frappe puis la prononciation d'une nouvelle phrase phonétique. Comme au passage des paramètres la longueur de la chaîne est définie sur un octet, elle ne peut dépasser 255 caractères soit plus de 5 lignes d'écran.

## VERSION MO5

```

10 CLS
20 SCREEN2,0,0
30 CLEAR,&HB000
40 RESTORE70
50 FOR A=&H00 TO &H46
60 READ D:POKE&HB500+A,D:NEXTA
70 DATA &H34,&H3F,&H10,&HBE,&HA4,&H00,&HB6,&H1A,&HB7,&HA7,&H
FF,&HB6,&H3C,&HB7,&HA7,&HFE,&H31,&H3F,&H31,&H21,&HBE,&H80,&H
00,&H6D,&HA4,&H27,&H2A,&HA6,&HB4,&HA1,&HA4,&H27,&H0B,&HB1,&H
FF,&H27,&HED,&H30,&H03,&H20,&HCE,&HEC,&H01,&HC3,&H80,&H01,&H
1F,&H01
80 DATA &HA6,&HB4,&H30,&H03,&H80,&H04,&HE6,&H80,&H7D,&HA7,&H
FF,&H2A,&HFB,&HF7,&HA7,&HFE,&H4A,&H26,&HF3,&H20,&HCD,&H35,&H
BF
90 LOADM"PHONEME"
95 POKE &HB400,00
110 LOCATE 0,10 :ADD=&HB400
115 B$=" ":PRINT"Mot: ";
120 A$=INPUT$(1):IF ASC(A$)=13 THEN 140
121 IF ASC(A$)=32 THEN IF ADD=&HB400 THEN 145
122 PRINTA$; :POKE ADD,ASC(A$): ADD=ADD+1:B$=B$+A$
124 GOTO 120
140 C!S:POKE ADD,00:LOCATE 0,2:PRINT B$
145 EXEC &HB500
150 GOTO 110

```

Contrairement au BASIC du TO7, celui du MO5 ne dispose des instructions DEFUSR et USR qu'avec le DOS. Nous sommes donc obligés de bâtir un autre programme en utilisant l'autre instruction qui permet d'appeler un sous-programme : EXEC. Mais EXEC ne permet pas le passage de paramètres, nous sommes donc dans l'obligation de réserver une zone mémoire que nous utiliserons pour le transfert des chaînes de caractères. Dans notre application sur MO5, cette zone est située entre la fin de la table des codes vocaux (\$8400) et le début du sous-programme binaire (\$8500) soit 255 octets.

Le sous-programme binaire est identique à celui étudié en ASSEMBLEUR ; si ce n'est qu'il est translaté de l'adresse \$8320 à l'adresse \$8500. Tout comme pour TO7, les codes vocaux sont chargés par LOADM "PHONEME" pour l'utilisateur qui ne dispose que d'un périphérique en ligne. Dans le cas contraire, il convient de préciser sur quel périphérique on souhaite faire la lecture (0 : disquette, CASS : cassette).

Dans l'application MO5, la fin de la chaîne est spécifiée au sous-programme binaire par un octet à zéro. A l'initialisation, le premier octet de la chaîne est positionné à zéro, c'est le rôle de la ligne 95. Une variable ADD est affectée à l'adresse qui pointe sur l'octet à charger.

Les caractères de type chaîne sont entrés un à un, puis "filtrés" par les deux tests IF ASC(A\$) = 13 et IF ASC(A\$) = 32 qui identifient les caractères pour réaliser les branchements conditionnels ; le premier sur la touche **ENTREE** qui signifie "fin de la chaîne", le second sur la touche **ESPACE** qui signifie "répète la dernière phrase entrée" si le pointeur d'adresse est au début de la chaîne.

A la ligne 122, le POKE ADD,ASC(A\$) permet d'écrire l'octet dans la chaîne. La ligne 140 précède l'appel du sous-programme binaire et est exécutée après que le premier test ait détecté la touche **ENTREE**. Il convient donc de marquer la fin de chaîne en mémoire en chargeant zéro dans l'octet pointé par ADD. Le sous-programme est appelé en ligne 145 puis un GOTO recommence un cycle.

## **Votre premier jeu parlant**

Pour vous permettre d'apprécier la loquacité de votre ordinateur, nous vous proposons un petit jeu dit du "*Trop petit trop grand*". Ce jeu reprend la structure des exemples que nous avons vu sur TO7, utilise les mêmes phonèmes et emprunte le sous-programme binaire. Les paramètres "chaîne" sont passés par l'instruction USR. Afin que puissiez suivre sur le listing les discours de votre "grand bavard", les commentaires sont traduits en remarques dans le programme.

L'avantage d'un logiciel parlant, c'est qu'il dicte lui même ses règles. Dans notre cas le programme, ... non l'ordinateur ! vous demandera après les présentations d'usage de jouer avec lui. Si vous lui répondez par la négative, il vous insultera et coupera la conversation. Si votre réponse est positive, il vous demandera un nombre, afin de trouver celui qu'il a réservé pour vous. Après votre réponse, il vous indiquera si votre nombre est trop petit ou trop grand pour que vous puissiez ajuster votre tir. Un conseil, soyez logique !, sinon gare à la pluie d'injures... Amusez-vous bien !

```

10 CLS
15 SCREEN2,0,0
20 CLEAR,&HA000
25 RESTORE40
30 FOR A=&H02 TO &H5B
35 READ D:POKE&HA400+A,D:NEXTA
40 DATA &H34,&H7F,&HB1,&H03,&H26,&H51,&HA6,&HB0,&H27,&H4D,&H
B7,&HA4,&H00,&H10,&HAE,&HB4,&HB6,&H1A,&HB7,&HE7,&HFF,&HB6,&H
3C,&HB7,&HE7,&HFE,&HA6,&HA0,&HB7,&HA4,&H01,&HBE,&HA0,&H00,&H
A6,&HB4,&HB1,&HFF,&H27,&H2E,&HB1,&HA4,&H01,&H27,&H04,&H30,&H
03,&H20,&HF1
45 DATA &HEC,&H01,&HC3,&HA0,&H00,&H1F,&H01,&H36,&H20,&H10,&H
AE,&HB4,&H30,&H04,&H31,&H3C,&HA6,&HB0,&H7D,&HE7,&HFF,&H2A,&H
FB,&HB7,&HE7,&HFE,&H31,&H3F,&H26,&HF2,&H37,&H20,&H7A,&HA4,&H
00,&H26,&HC4,&H12,&H35,&H7F,&H39
50 LOADM"PHONEME"
55 DEFUSR=&HA402
60 C1 S:SCREEN2,0,0
65 LOCATE12,4:ATTRB1,1
70 PRINT"BONJOUR":P$="bojwr":P$=USR(P$)
75 FORX=0 TO 300:NEXT
80 LOCATE6,8:PRINT"COMMENT ALLEZ"
85 LOCATE14,12:PRINT"VOUS ?"
90 P$="komA.tal*.vw":P$=USR(P$)
95 FOR X=0 TO 3000:NEXT
100 CLS
105 LOCATE0,1:PRINT"OUI !!?":P$="wi"
110 P$=USR(P$):FORX=0 TO 300:NEXT
115 PRINT"C'est moi...":P$="s&.mW..."
120 P$=USR(P$):PRINT"l'ordinateur qui"
125 PRINT"vous parle":P$="l'odiner.ki.vw.parle"
130 P$=USR(P$)
135 FORX=0 TO 4000:NEXT:CLS
140 ATTRB0,0:PRINT"On m'a appris à parler."
145 P$="O.ma.apri.a.parl*":P$=USR(P$)
150 PRINT"c'est pas encore super!..mais"
155 PRINT"j'espère que vous me comprenez bien "
160 P$="s&.pa.AkO.sup&r..m& j&sp&r.ke.vw.me.konpren*.b<"
165 P$=USR(P$)
170 FORX=0 TO 4000:NEXT
175 PRINT:PRINT:PRINT"VOUS ME COMPRENEZ ?";
180 P$="vw.me.konpren*... *pond*":P$=USR(P$)
185 INPUT" "A$;
190 A$=LEFT$(A$,1)
195 IF A$="O" OR A$="o" THEN 245;
200 PRINT"Ce n'est pas de ma faute si vous"
205 PRINT"êtes bouché"
210 P$="se.nf.pa.de.ma.fote.si.vu.zft.bw*...":P$=USR(P$)
215 PRINT"Les oreilles c'est comme le reste ça":PRINT"se lave,"
"
220 P$="..l*.zor#.e&.kome.le.r&st.sa.se.lave..":P$=USR(P$)
225 PRINT"je n'ai rien à faire avec vous:SALUT!!"
230 P$="..je.nf.r<.na.f&re.av&k.vw..salu"
235 P$=USR(P$)
240 END
245 P$="tr&.b<...je.ne.v&.plu.*krire."
250 'trés bien.. je ne vais plus écrire
255 P$=USR(P$)
260 P$="...je.v&.sElmA.parl*":P$=USR(P$)
265 'je vais seulement parler
270 GOSUB 485
275 CLS:P$="..O.va.jw*,a.l.jE.f.sAble..":P$=USR(P$)
280 'on va jouer à un jeu ensemble

```

```

285 GOSUB 485
290 P$="je.v&.$wsir.I.nombre.Atr.z*ro...$.sA...vw..devr$.le r
etrwv$"
295 'je vais choisir un nombre entre zero et cent vous deve
z le retrouver.
300 P$=USR(P$)
305 GOSUB 485
310 P$="vw.fer$.une.propozisiO...mW.je.vw.dir$.si.&l.l&.p
lu.grAde.w.plu.petit."
315 'vous me ferez une proposition.Moi je vous dirais si ell
e est plus grande ou plus petite.
320 P$=USR(P$);P$="..ke.le.nombre.ke.j*.$wazi":P$=USR(P$)
325 '..que le nombre que j'ai choisi
330 GOSUB 485
335 P$="b<sur.vw.dev*trwv$.A.I.minimome.de.kw":P$=USR(P$)
340 '..bien sur vous devrez trouver en un minimum de coups..
345 GOSUB 485;GOSUB485
350 P$="bO....je.$r*e.le.nombre":P$=USR(P$);GOSUB 485
355 'je cherche le nombre
360 P$="nO.pas.selui.la.il.*.tro.sIple.....a...":P$=USR(P$)
365 'non pas celui-la il est trop simple
370 P$="...a.vwla.j*.trwv*.....sa.va.*tre.dur":P
$=USR(P$)
375 '..a voila j'ai trouve .....
380 N=INT(RND*100+1)
385 GOSUB485;P$="alor.ke.me.propoz$.vw"
390 'alors que me proposez vous.....
395 P$=USR(P$);GOSUB485;GOSUB485
400 GOSUB485;P$="a.wi.j*.wbli*.ke.vw.ne.savi*.pa.AkO.parl*.O
rdinater...vw":P$=USR(P$)
405 'ah! oui j'ai oublie que vous ne savez pas encore parler
:ordinateur.. VOUS
410 GOSUB485;P$="atAd*.vw.zal*.pwvWr.le.rAtr*.a.l&d.du.klavi
*":P$=USR(P$)
415 'attendez vous allez pouvoir le rentrer au clavier
420 GOSUB485
425 P$="votre.propozisiO":P$=USR(P$)
430 CLS;INPUT"Votre proposition ";A
435 IF A<0 OR A>100 THEN 490
440 IF A=N THEN 510
445 R=INT(FND*4+1)
450 ON R GOSUB 550,560,570,580,590
455 P$="le.nombre$.le.vw.propoz*.$&.tro...":P$=USR(P$)
460 'le nombre que vous proposez est trop....
465 IF A>N THEN P$="grA" '..grand
470 IF A<N THEN P$="peti" '..petit
475 P$=USR(P$);GOSUB485;GOSUB485
480 GOTO425
485 FORX=0 TO 1000:NEXT:RETURN
490 P$="il.mesAble.ke.jav&.di.Atre.z*ro $.sA":P$=USR(P$);GOS
UB 485
495 'il me semble que j'avais dit... entre zero et cent
500 P$="vw.ne.sav*.n*m.pe.kont*":P$=USR(P$);GOSUB485;GOTO425
505 'vous ne savez meme pas compter
510 P$="bravo...bravo..vw.zav*.trwv*":P$=USR(P$);GOSUB 485
515 'bravo ..bravo vous avez trouve
520 P$="m& s*t& sIple....O f& une otre parti":P$=USR(P$)
525 'mais c'etait simple...On fait une autre partie
530 PRINT:PRINT:PRINT:PRINT:INPUT"Autre partie ";R$
535 R$=LEFT$(R$,1):IF R$="O" OR R$="o" THEN GOTO 275
540 P$="a.v..zav*.pEr.de.p*rdre.s&t.fW":P$=USR(P$);END
545 'ah!vous avez peur de perdre cette fois
550 P$="aa..ii..aaaa...s&.lwp*..domaje":GOTO595

```

```

555 'ah!ah!ih!ih!....C'est loupe..damage
560 P$="ii..vw.z&t.le.plu.nul.de.sE.ki O.jw*.av&k.mW.":GOTO5
95
565 '...Vous etes le plus nul de ceux qui ont joue avec moi
570 P$="pa.de.$Ase.":GOTO595
575 'pas de chance
580 P$="swi*.I.pE.1ojik":GOTO595
585 'soyez un peu logique
590 P$="il.fodr&.prAdre.d*.kwr.de mat."
595 P$=USR(P$):GOSUR485:RETURN

```

## Y'a une pie dans le poirier... J'entends mon TO7 chanter...

Pour conclure avec le BASIC, nous vous suggérons de faire chanter votre ordinateur. La structure du programme est des plus simples et est inspirée de la troisième variante du programme BASIC donnée précédemment.

En annexe (page 305), vous disposez de deux "partitions" (*Au Clair de lune* et *Du bon tabac*) qu'il vous faudra implanter à partir de l'adresse \$A000 pour TO7, TO7/70 ou TO9 et \$8000 pour MO5.

La "partition", ou codes vocaux d'une chanson, est considérée comme une et une seule expression. La table des adresses relatives est donc inutile, de même que la séquence de calcul de l'adresse absolue dans le programme. L'adresse de début de la partition est passée directement en DATA (&HCE,&HA0,&H00) pour TO7 et (&HCE,&H80,&H00) pour MO5.

Les deux premiers octets donnent la longueur de la "partition" et la hauteur initiale est spécifiée par le quatrième octet.

### VERSION TO7-TO7/70-TO9

```

10 CLEAR,&H9FD3
20 RESTORE50
30 FOR A=&HD4 TO &HF3
40 READ D:POKE&H9F00+A,D:NEXTA
50 DATA &H34,&H57,&HCE,&HA0,&H00,&HC6,&H1A,&HF7,&HE7,&HFF,&H
AE,&HC1,&H33,&H41,&H30,&H1D,&HE6,&HCO,&H7D,&HE7,&HFF,&H2A,&H
FB,&HF7,&HE7,&HFE,&H30,&H1F,&H26,&HF2,&H35,&HD7
80 EXEC&H9FD4
120 STOP

```

### VERSION MO5

```

10 CLEAR,&H7FD3
20 RESTORE50
30 FOR A=&HD4 TO &HF3
40 READ D:POKE&H7F00+A,D:NEXTA
50 DATA &H34,&H57,&HCE,&H80,&H00,&HC6,&H1A,&HF7,&HA7,&HFF,&H
AE,&HC1,&H33,&H41,&H30,&H1D,&HE6,&HCO,&H7D,&HA7,&HFF,&H2A,&H
FB,&HF7,&HA7,&HFE,&H30,&H1F,&H26,&HF2,&H35,&HD7
80 EXEC&H7FD4
120 STOP

```

# Utilisation sous FORTH

---

L'application ci-dessous, réalisée complètement en FORTH, crée dans le dictionnaire du langage FORTH des "mots FORTH" dont l'action sera de prononcer les mots ou expressions correspondants. Le mot "PARL" a pour tâche d'actionner le circuit de synthèse vocale MEA 8000. On dit aussi réaliser la synthèse. Chaque "mot FORTH" devra donc positionner la variable RANG avec le rang dans la table du mot souhaité avant de "demander" à PARL d'agir. Ainsi le mot BONJOUR qui est le premier mot dans la table des codes vocaux utilisés s'écrira :

```
:BONJOUR
  0 RANG!
  PARL
;
```

Le FORTH est un langage merveilleux pour la synthèse de parole puisque un fois que le mot est défini en FORTH, il suffit de taper le texte du mot au clavier pour qu'il soit prononcé.

## VERSION TO7-TO7/70-TO9

```
HEX
E7FE CONSTANT DONN ( Registre de donnée )
E7FF CONSTANT RCOM ( Registre de commande )
A000 CONSTANT DTAB ( Début de table )
      VARIABLE RANG ( Rang dans la table )
```

```
: PARL ( Realise la synthèse )
  RANG @ ( Rang )
  15 ( Nombre de mots en hexa dans la table )
  > NOT
  IF ( Si rang <= nombre de mots )
    2* ( Deux octets par adresse )
    DTAB + @ ( Adresse relative )
    DTAB + ( Adresse absolue )
    1A RCOM C! ( Positionne registre de commande )
    DUP 3 + >R ( Adresse debut de synthèse )
    @ 3 - ( Longueur du mot )
```

```

BEGIN          ( On envoie les octets )
  R> DUP 1+ >R
  C@          ( Hauteur initiale ou donnée )
  BEGIN      ( Test du registre de commande )
    RCOM C@
    <0
  UNTIL
  DONN CI    ( Envoi de l'octet )
  1-        ( Décompte du nombre d'octets )
  0=        ( Et on continue jusqu'à 0 )
  UNTIL
  R> DROP    ( Supprime sauvegarde )
;

: DEBUT
  3 RANG !   ( Troisième mot )
  PARL      ( Appel synthèse )
;

```

---

*Note: programme écrit et communiqué par SEFI. Avec les remerciements des auteurs.*

# Utilisation sous LOGO

---

Le LOGO c'est, bien entendu, le monde de "Madame Tortue". Mais ne croyez-vous pas qu'il serait intéressant de lui donner la parole à notre LOGO de tortue ou tortue de LOGO ? On ne sait plus très bien ! Pour cela, nous allons lui apprendre à parler. Apprendre à parler ou chanter en LOGO, c'est définir les procédures correspondantes. Il nous faut définir deux procédures :

— La première devra charger en mémoire les programmes binaires déjà utilisés en ASSEMBLEUR (TABL.BIN et PARL.BIN, voir page 75). Nous appellerons cette procédure "BINAIR".

— La seconde qui sera spécifique à chaque mot ou expression, nous lui donnerons le nom de l'expression à prononcer. Ainsi la procédure "BONJOUR" devra prononcer le mot "bonjour", etc.

## Procédure BINAIR

POUR BINAIR

FLECTEUR0 (FLECTEUR7 pour cassette)

.CHB TABLO

.CHB PARLO

FIN

## Procédure BONJOUR

POUR BONJOUR

.DEP 40883 0 (32691 pour MO5)

.ROUT 40893 (32701 pour MO5)

FIN

Pour faire prononcer "bonjour" à votre tortue :

? BINAIR

... les fichiers binaires sont chargés

? BONJOUR

... la tortue vous dit "bonjour"

Maintenant, à vous de définir d'autres procédures "parlantes" !

# Utilisation sous LSE

---

Le LSE (Langage Symbolique d'Enseignement) est un langage destiné à l'enseignement dans le système éducatif français. Comme il est implanté sur les machines TO7-TO7/70 et MO5, sous le nom de LSEG-EDL, certains lecteurs peuvent être tentés par la rédaction de logiciels parlants fonctionnant sous ce langage.

La rédaction du logiciel se résume à l'élaboration d'une procédure binaire qui sera appelée sous LSE et à laquelle nous passerons deux paramètres: un de type numérique (le rang dans la table) et l'autre de type chaîne (début de la table). Cette procédure détermine, à partir d'un drapeau, le micro-ordinateur utilisé (TO7 ou MO5). Ce qui conduit à avoir une procédure unique pour les deux machines. La procédure binaire interne (INIPB), élaborée par le CNDP, met à jour l'octet MICRO et les adresses de branchements de la dernière table.

Soit:

```
10 PROCEDURE BINAIRE PARL
20 CHARGER TABL,1,'COVOC'
50 PARL (3,TABL)
70 TERMINER
```

ERREUR 201 si paramètres invalides

Procédure binaire: (source) "communiquée par le CNDP"

ERROR	EQU	\$0124	
POPCH	EQU	\$0106	
POP16	EQU	\$0100	
DEBUT	BRA	START	
	FCB	1	N° de version
	FDB	FIN1-DEBUT	Longueur
	FCB	\$82	1er modalité
	FCB	\$00	
MICRO	FCB	\$08	pour TO7
GARAD	RMB	2	début de table
RCOM	RMB	2	registre de commande
DONN	RMB	2	registre de données
PARL	PSHS	D,U,X,CC	Sauve le contexte
* Mise à jour de RCOM et DONN en fonction de MICRO			
	LDA	*\$20	MO5?
	ANDA	MICRO,PCP	
	BNE	MO5	oui

```

* Pour T07 ou T07/70 (T09 futur)
    LDD    *$E7FF
    STD    RCOM,PCR
    LDD    *$E7FE
    STD    DONN,PCR
    BRA    SUITE
M05      LDD    *$A7FF
    STD    RCOM,PCR
    LDD    *$A7FE
    STD    DONN,PCR
*Dépile les variables de la pile
    BSR    RPOPCH      Récupère la table
    STX    GARAD,PCR
    BSR    RPOP16      Rong de la table
*Valide le rong
    TSTA
    BNE    ERR201
    TSTB
    BMI    ERR201
    CMPB   *20
    BGT    ERR201
    TFR    B,A
    ASLA
    LDU    GARAD,PCR   2 octets par adresse
    LDD    A,U         Début de la table
    LEAU   D,U         Adresse relative
    LDB    *1A        Arrêt lent et REQ inactif
    STB    [RCOM,PCR]
    LDX    ,U++       Longueur du mot
    LEAU   1,U        Saut le 3em octet
    LEAX   -3,X       Déjà 3 octets de passés
RET      LDB    ,U+   Hauteur initiale/données
AR1     TST    [RCOM,PCR]
        BPL    AR1
        STB    [DONN,PCR2]  Envoie l'octet
        LEAX   -1,X       Décompte
        BNE    RET
        PULS   D,U,X,CC,PC

ERR201  LDA    *201
        BRA    RERROR

DTREL
RERROR  JMP    ERROR
RPOCH  JMP    POPCH
RPOP16 JMP    POP16
FTREL
        FCB    (FTREL-DTREL)/3

FIN1

```

Le programme source ci-dessus s'assemble sur GOUPIL 3-6809 sous l'assembleur FLEX ou sur T07 avec l'assembleur A6809 diffusé par le Centre National de Documentation Pédagogique.

# Applications sur d'autres machines

---

Pour chacune des machines étudiées ci-après, un programme permettant la synthèse d'un texte quelconque à partir de phonèmes est fourni, ainsi que les données (schéma et nomenclature) permettant la construction du module synthétiseur. Ce logiciel de composition phonétique ayant pratiquement les mêmes caractéristiques pour toutes les machines, nous expliquerons sa philosophie générale dans ce paragraphe ; les particularités relatives à chaque machine seront indiquées au paragraphe correspondant.

D'autres programmes seront décrits pour certaines machines, que le lecteur pourra aisément adapter à d'autres matériels.

Le logiciel de composition phonétique se compose de 3 parties:

- Un programme BASIC permettant la création d'une expression sous forme phonétique, chaque phonème étant représenté par un caractère disponible au clavier (d'où une différence de jeu de caractères d'une machine à l'autre).
- Un dictionnaire de phonèmes existant en deux versions (3.3 ou 4.2) et dont les codes sont fournis en annexe. La version à utiliser est précisée pour chacun des programmes.
- Une routine en langage-machine activant le synthétiseur vocal.

L'ensemble "phonèmes + routine L.M." est sauvegardé en un seul bloc baptisé en général "PHONESPEECH". Si l'on utilise une cassette comme sauvegarde, il devra être enregistré à la suite du programme BASIC pour permettre son chargement à partir de celui-ci. L'emplacement en mémoire de "PHONESPEECH" est déterminé à partir de la cartographie mémoire du micro-ordinateur considéré, de façon à ce que le BASIC ne puisse l'écraser en aucun cas.

## Fonctionnement

Après chargement du programme BASIC, l'exécution de celui-ci commence par une initialisation et le chargement de l'ensemble "phonèmes + routine L.M.". Le programme affiche ensuite le jeu de caractères utilisés pour représenter chaque phonème, et attend l'entrée de l'expression à vocaliser.

La chaîne de caractères (L\$) doit être terminée par l'un des signes de ponctuation indiqués, correspondant à un silence, pour ne pas tronquer la

**dernière trame. A partir de cette chaîne, le programme lit pour chaque caractère (A\$) le numéro M du phonème correspondant, et calcule l'adresse de son adresse ( $DB + 2 * M$ ) dans la table située au début de la zone mémoire "phonèmes", DB étant l'adresse du début du dictionnaire de phonèmes.**

**Il va alors lire le deuxième octet du phonème qui représente sa longueur K (en effet, le premier est toujours nul, vue la courte durée des phonèmes). Il recopie ensuite tous les octets du cinquième au dernier (octets de trames vocales) dans la zone mémoire où l'expression va être fabriquée, à partir de l'adresse de début choisie pour l'expression, augmentée de 4, pour réserver la place des 4 octets d'en-tête. Il procède de même pour tous les caractères, qu'il place à la suite les uns des autres, en calculant à chaque fois la longueur de l'expression résultante C.**

**Lorsque le dernier caractère a été traité, on écrit la longueur C de l'expression finale (2 premiers octets de l'expression, poids fort en tête), et le "pitch initial" (4ème octet). On a ainsi créé une expression de format conforme à celui expliqué au chapitre "Logiciel de commande". L'expression est alors prononcée par le synthétiseur, par un appel de la routine en langage-machine. Le processus décrit ci-dessus est le même pour toutes les machines et constitue le cœur du programme.**

**Selon les versions, tout ou partie des fonctions supplémentaires ci-dessous est réalisé :**

— **Certaines ponctuations finales particulières permettent de créer une expression chuchotée, ou une courbe d'intonation préfabriquée par modification de l'incrément du pitch de chaque trame.**

— **Certaines versions du logiciel permettent de modifier l'intonation de façon locale par l'insertion de marqueurs appropriés (montée, plat, descente du pitch). Chaque marqueur détermine une allure de courbe entre sa position et celle du précédent (ou le début d'expression si c'est le premier); ceci est également obtenu en modifiant les bits définissant l'incrément du pitch.**

— **On peut également allonger ou raccourcir la durée d'un phonème en le faisant suivre de signes "+" ou "-", ce qui a pour effet de rajouter ou retrancher une trame (l'avant-dernière) au phonème considéré; plusieurs signes consécutifs peuvent être utilisés.**

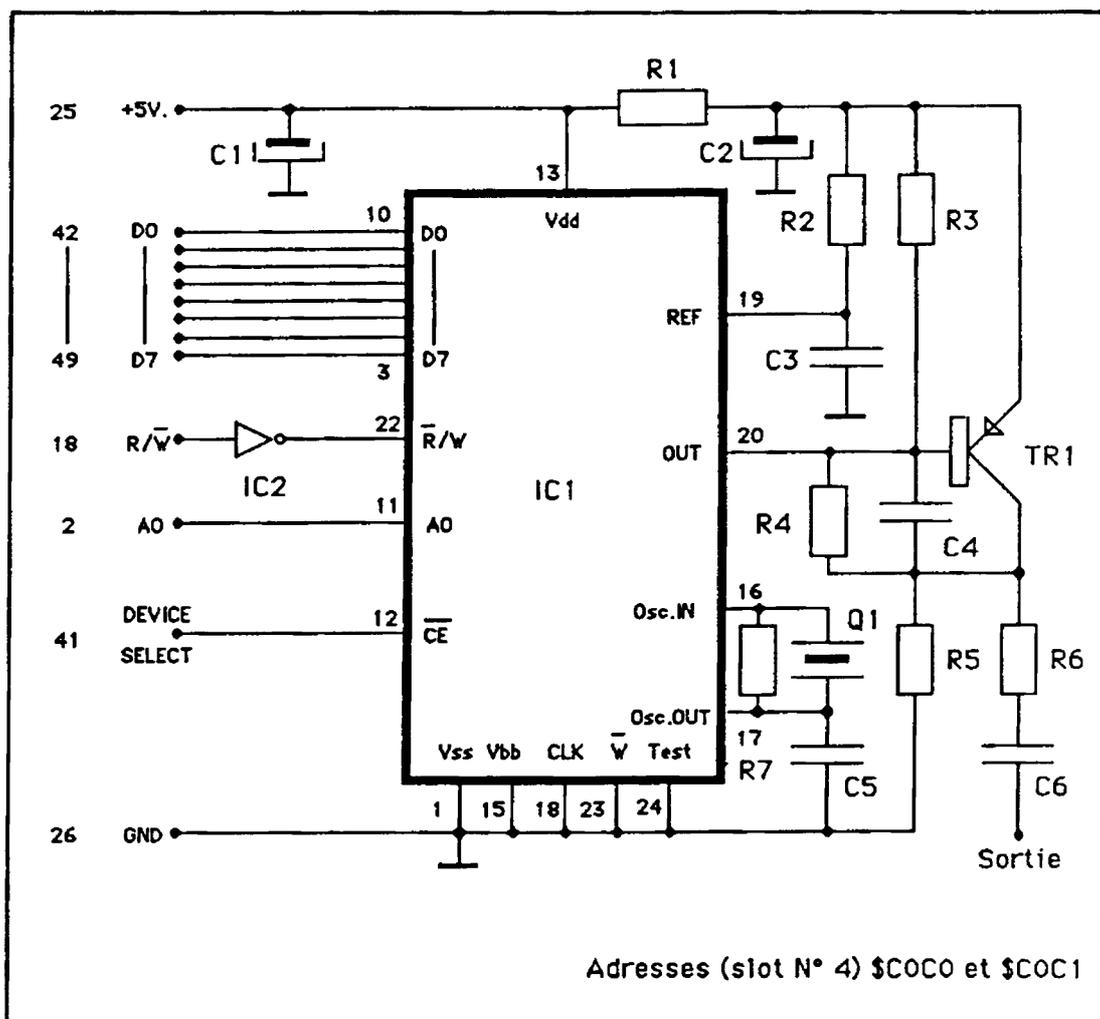
— **Enfin, certains programmes permettent de modifier la vitesse d'élocution en faisant précéder l'expression d'un ou deux symboles spéciaux, ce qui permet d'obtenir 5 vitesses en tout (rapports 0,5 -0,75 -1,0 -1,50 et 2 par rapport à la normale); ceci a pour effet de modifier la durée des trames (augmentation ou réduction de toutes ou d'une sur deux). On peut corriger, réécouter et sauvegarder l'expression ainsi créée pour la réutiliser éventuellement dans un autre programme.**

# Applications sur APPLE II, COMMODORE, ORIC (6502/6510)

Ce chapitre est consacré aux applications sur trois machines équipées du microprocesseur 6502 ou 6510. Il pourra servir de base aux applications sur d'autres machines utilisant ce type de microprocesseur.

## APPLE IIe ou II+

Voici tout d'abord le schéma du synthétiseur et la nomenclature donnant la valeur des composants. Le synthétiseur doit être enfilé dans le connecteur (slot) n° 4, ce qui lui affecte les adresses \$C0C0 (données) et \$C0C1 (commandes). La correspondance entre les signaux et les numéros des bornes est indiquée dans le manuel APPLE.



IC1	1	MEA 8000	Synthétiseur ( version M 4244 )
IC2	1	74 LS 04	Sextuple inverseur
IC3-IC4	2 *	74 LS 30	Porte NAND à 8 entrées
IC5	1 *	74 LS 00	Quadruple NAND à 2 entrées
Q1	1 *	4,000 MHz	Quartz
TR1	1	BC 558	Transistor PNP ( ou similaire )
C1-C2	2	47 $\mu$ F / 10 V	Condensateur électrolytique
C3	1	10 nF	Condensateur polyester
C4	1	22 nF	Condensateur polyester
C5	1 *	22 pF	Céramique
C6	1 *	0,47 $\mu$ F	Condensateur polyester
R1	1	47 $\Omega$	Résistance carbone (1/4W-5%)
R2	1	18 K $\Omega$	Résistance carbone (1/4W-5%)
R3	1	470 $\Omega$	Résistance carbone (1/4W-5%)
R4	1	1,5 K $\Omega$	Résistance carbone (1/4W-5%)
R5	1	330 $\Omega$	Résistance carbone (1/4W-5%)
R6	1	33 $\Omega$	(VG5000:2,2 K $\Omega$ ; AMSTRAD: 10 K $\Omega$ )
R7	1 *	1 M $\Omega$	Résistance carbone (1/4W-5%)
<b>REPERE</b>	<b>Nbr.</b>	<b>Valeur ou Type</b>	<b>Désignation et ( observations )</b>

\* Composant absent dans certaines applications

Sur cette machine, le dictionnaire de phonèmes est à charger à partir de l'adresse \$7000. Les versions 3.3 ou 4.2, dont le listing se trouve en annexe, peuvent être utilisées indifféremment, étant donné l'espace-mémoire disponible.

Le programme en langage-machine (SPEECH) occupe les adresses \$6EF6 à \$6F48; il est réalisé selon le modèle expliqué au chapitre "Logiciel de commande" page 50 , et son listing hexadécimal est fourni ci-dessous. Les adresses en page zéro utilisées pour l'adressage indirect post-indexé sont \$FE et \$FF.

```

$6EF0 00 00 00 00 00 00 A9 00 8D FE 00 A9 60 8D FF 00
$6F00 20 2D 6F 20 35 6F A2 00 A0 03 20 3B 6F 20 2D 6F
$6F10 20 3B 6F 20 3B 6F 20 3B 6F 20 3B 6F CC 01 60 D0
$6F20 EC EC 00 60 30 E7 20 2D 6F 20 35 6F 60 A9 80 CD
$6F30 C1 C0 D0 F9 60 A9 1A 8D C1 C0 60 B1 FE 8D C0 C0
$6F40 C8 C0 00 D0 03 E8 E6 FF 60 FF 00 FF 00 FF 00 FF

```

L'APPLE disposant d'un moniteur intégré (appelé par CALL 151), c'est la manière la plus pratique d'entrer les codes dans la mémoire. Une fois cette opération et toutes vérifications effectuées, sauvegarder l'ensemble phonèmes + routine sur disquette par la commande :

BSAVE PHONESPEECH, A\$6EF6, L\$50A (Phonèmes 3.3)

ou

BSAVE PHONESPEECH, A\$6EF6, L\$6C2 (Phonèmes 4.2)

```

1 PRINT CHR$(4);"BLOAD PHONESPEECH"
2 HOME : PRINT : PRINT "          PHONETIC (C) H. BENOIT 1984"
5 PRINT "          *****"
10 PRINT : PRINT "COMPOSEZ VOTRE PHRASE AVEC LES SYMBOLES
    PHONETIQUES CI-DESSOUS"
15 PRINT : PRINT "A,E,I,O,O(AU),U,*<(EH),&<(AI),W<(OU),Q<(ON)"
20 PRINT : PRINT "1<(IN),2<(EU),3<(OI),4<(AN),<<(IEN),>>(OIN)"
22 PRINT : PRINT "B,D,F,G,J,K,L,M,N,P,R,†,S,T,U,Z,$<(CH)"
25 PRINT : PRINT "€<(AIL),*€(EIL),%<(EUIL),Y<(GN) SILENCE ; ."
28 PRINT : PRINT "INTONATION / = \ DUREE PHONEME + - "
30 PRINT : PRINT "PONCTUATION FINALE OBLIGATOIRE .] "
32 PRINT : PRINT
35 INPUT L$:L = LEN (L$):C = 4:C0 = 4
38 FOR I = 1 TO L
40 A$ = MID$ (L$,I,1)
42 IF A$ = " " THEN A$ = ";"
43 IF A$ = "]" THEN A$ = "-"
45 IF A$ = "-" THEN 600
48 IF A$ = "+" THEN 500
50 IF A$ = "/" THEN 700
51 IF A$ = "\" THEN 700
52 IF A$ = "=" THEN 85
55 READ B$,M: IF B$ < > A$ THEN 55
60 A = 112 + PEEK (28672 + 2 * M):B = PEEK (28673 + 2 * M)
65 K = PEEK (256 * A + B + 1)
70 FOR J = 4 TO K:N = PEEK (256 * A + B + J)
75 POKE 24572 + C + J,N: NEXT J: RESTORE
80 C = C + K - 4
85 NEXT I
90 POKE 24576, INT (C / 256): POKE 24577,C - 256 * INT (C / 256):
    POKE 24579,60
100 IF A$ = " " THEN GOSUB 900
125 CALL 28406
130 PRINT : PRINT "    PRESSER 0 POUR TERMINER"
135 PRINT "    PRESSER 1 POUR REPETER"
140 PRINT "    PRESSER 2 POUR RECOMMENCER"
145 PRINT "    PRESSER 3 POUR SAUVEGARDER"
148 VTAB 19

```

```

150 GET K: IF K = 0 THEN HOME : END
152 IF K = 1 THEN 125
155 IF K = 2 THEN 2
160 IF K = 3 THEN 1000
200 DATA A,0,E,1,I,2,0,3,U,4,&,5,*,6,2,7,W,8,4,9,1,10,Q,11,3,12
210 DATA B,13,D,14,F,15,G,16,J,17,K,18,L,19,M,20,N,21,P,22,R,23,↑,24
220 DATA S,25,T,26,V,27,Z,28,$,29,Y,30,©,31,#,32,%,33,<,34,>,35,0,36
,;,38,.,39,_,39
500 FOR J = 0 TO 3:N = PEEK (24576 + C - J)
510 POKE 24580 + C - J,N
520 N = PEEK (24572 + C - J): POKE 24576 + C - J,N
530 NEXT J:C = C + 4: GOTO 85
600 FOR J = 0 TO 3:N = PEEK (24576 + C - J)
610 POKE 24572 + C - J,N: NEXT J
620 C = C - 4: GOTO 85
700 FOR Q = 24579 + C0 TO 24579 + C STEP 8
710 E = PEEK (Q):G = INT (E / 16)
715 IF G / 2 > INT (G / 2) THEN 745
720 IF A$ = "/" THEN G = 2 * INT (G / 2):E = 16 * G + 1
730 IF A$ = "\ " THEN G = 2 * INT (G / 2) + 1:E = 16 * G + 15
740 POKE Q,E
745 NEXT Q
750 C0 = C: GOTO 85
900 FOR Q = 24583 TO 24575 + C STEP 4
910 E = PEEK (Q):G = INT (E / 16)
920 IF G / 2 > 2 * INT (G / 2) THEN 950
930 G = 2 * INT (G / 2) + 1:E = 16 * G
940 POKE Q,E
950 NEXT Q: RETURN
1000 HOME : INPUT "NOM DE L EXPRESSION?";N$
1005 C$ = STR$ (C)
1010 R$ = "BSAVE " + N$ + ", A$6000, L" + C$
1015 PRINT CHR$ (4);R$

```

Le programme BASIC appelé "PHONETIC", dont le listing est fourni ci-dessus, commence par charger l'ensemble "PHONESPEECH"; il affiche le jeu de caractères et attend l'entrée de l'expression; il offre les possibilités suivantes:

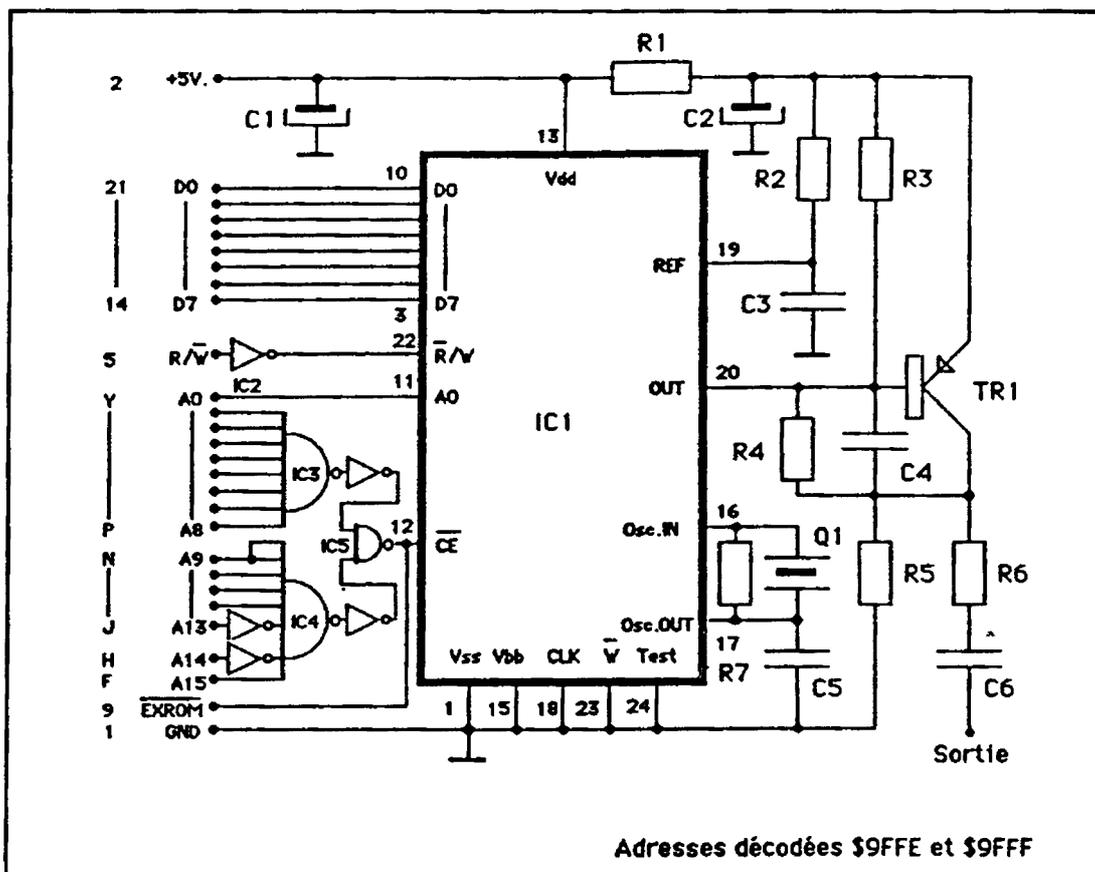
- Chuchotement par le symbole ] en fin d'expression.
- Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou -.
- Création d'une courbe d'intonation par l'utilisation des marqueurs / (intonation montante), = (intonation constante) et \ (intonation descendante). Chaque symbole définit la pente considérée entre le marqueur précédent et lui-même. Quelques indications pour la création d'une courbe d'intonation artificielle sont données page 60 .

L'expression créée se développe à partir de l'adresse \$6000, ce qui autorise une longueur d'expression de 3840 octets (jusqu'à \$6EF5). Une fois l'expression créée (ce qui dure quelques instants), le synthétiseur la

prononce (ligne 125 CALL 28406). Il vous est alors possible de terminer (0), de répéter (1), de recommencer l'entrée (2), ou de sauvegarder votre expression (3). Dans ce dernier cas, le programme vous demande de lui donner un nom et se charge de la commande du lecteur de disquette. Pour réutiliser cette expression ultérieurement, il faudra simplement charger la routine "SPEECH" (\$6EF6 à \$6F48) et l'expression désirée, qui sera prononcée par l'instruction CALL 28406.

## COMMODORE 64

Le synthétiseur doit être enfiché dans le connecteur arrière pour cartouches; il occupe les adresses \$9FFE (données) et \$9FFF (commandes). Son schéma est représenté ci-dessous et la valeur des composants est donnée par la nomenclature page 118. La correspondance entre les signaux et les numéros des bornes est indiquée dans le guide de l'utilisateur COMMODORE. La sortie du synthétiseur peut être appliquée à l'entrée audio externe du COMMODORE (mélange avec la sortie du synthétiseur musical interne); ceci nécessite l'ouverture du canal audio externe (ligne 50 POKE 54296, 15).



Le dictionnaire de phonèmes (version 4.2) est à charger à partir de l'adresse \$C000 (49152). Le programme en langage-machine (SPEECH) occupe les adresses \$C600 à \$C648; il est réalisé selon le modèle expliqué au chapitre "Logiciel de commande" p. 50, et son listing hexadécimal est fourni ci-après. Les adresses en page zéro utilisées pour l'adressage indirect post-indexé sont \$FC (252) et \$FD (253).

```
$C600 20 2D C6 20 35 C6 A2 00 A0 03 20 3B C6 20 2D C6
$C610 20 3B C6 20 3B C6 20 3B C6 20 3B C6 CC 01 C7 D0
$C620 EC EC 00 C7 30 E7 20 2D C6 20 35 C6 60 A9 80 CD
$C630 FF 9F D0 F9 60 A9 1A 8D FF 9F 60 B1 FC 8D FE 9F
$C640 C8 C0 00 D0 03 E8 E6 FD 60 FF 00 FF 00 FF 00 FF
```

Le COMMODORE ne disposant pas de la fonction HEX\$ ni de commande BASIC pour sauvegarder un fichier binaire, ceci peut se faire grâce au programme ci-dessous, qui permet le chargement sous forme hexadécimale avec vérification de la somme tous les 16 octets, ainsi que la sauvegarde sur disquette ou cassette du fichier binaire constitué. Sauvegarder l'ensemble phonèmes + routine (adresses \$C000 à \$C648) sous l'appellation "PH4.2SPEECH".

```
100 REM*COPYRIGHT 1985 STEPHANE BENOIT*
110 INPUT"ADRESSE DE DEBUT":AC:AC=AC
120 INPUT"ADRESSE DE FIN":AF
130 PRINT"MACW":INPUTN#E=0
140 FORI=1TO16:H#=MID$(N#,2*I-1,1):L#=MID$(N#,2*I,1)
150 READY$:H:IFH$<>Y#THEN150
160 RESTORE
170 READX$:L:IFL$<>X#THEN170
180 RESTORE
190 A=16*H+L:POKEAC+I-1,A:B=B+A:NEXT
200 DATA0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,A,10,B,11,C,12,D,13,E,14,F,15
210 INPUT"SOMME",S
220 IFS<>BTHENPRINT"ERREUR":GOTO130
230 AC=AC+16:IFAC>=AFTHEN250
240 GOTO130
250 PRINT"VOULEZ-VOUS SAUVEGARDER LES CO
DES (O/N)":INPUTE#
260 IFE#="N"THENEND
280 INPUT"TITRE":NM#:L=LEN(NM#)
290 INPUT"DISQUE(8) OU CASSETTE(1)":PR
300 POKE780,1:POKE781,PR:POKE782,255
310 SYS65466
320 FORI=1TOL:POKE831+I,ASC(MID$(NM#,I,1)):NEXT
330 POKE780,L:POKE781,832-256+INT(832/256):POKE782,INT(832/256)
340 SYS65469
342 LD=AD-256*INT(AD/256)
344 LF=AF-256*INT(AF/256)+1
350 POKE251,LD:POKE252,AD/256
370 POKE780,251:POKE781,LF:POKE782,AF/256
380 SYS65496
390 END
```



```

370 IFRIGHT$(L$,1)="*"THEN GOSUB 500
380 POKE252,0:POKE253,199:SYS50688
390 PRINT"          PRESSER C POUR CORRIGER"
400 PRINT"          PRESSER S POUR SAUVEGARDER"
410 PRINT"          ESPACE POUR REPETER"
430 GETE$: IFE$="" THEN 430
440 IFE$=" " THEN PRINT"TTTT":GOTO380
450 IFE$="C" THEN PRINT"TTTTT":GOTO110
455 IFE$="S" THEN PRINT"TT":GOTO1250
460 PRINT:RESTORE:GOTO10
470 DATA 0,E,1,I,2,O,3,U,4,&,5,^,6,2,7,W,8,4,9,1,10,Q,11,
3,12,B,13,D,14|
480 DATA J,17,K,18,L,19,M,20,N,21,P,22,R,23,↑,24,S,25,T,26,
V,27,Z,28,$,29,
490 DATA @,31,#,32,%,33,<,34,>,35,0,36,.,38,..,39,G,16,Y,30,
*,39,F,15 |
500 FOR Q=50951 TO 50944+T-1 STEP 4
510 E=PEEK(Q)
520 G=INT(E/16)
530 R=E-16*G
540 IFR=0 AND G>2*INT(G/2) THEN 570
550 G=2*INT(G/2)+1:E=16*G
560 POKE Q,E
570 NEXT Q
580 RETURN
800 FOR J=0 TO 3
810 N=PEEK(50944+C-J)
820 POKE(50944+C-J-4),N
850 NEXT J
860 C=C-4
870 GOTO 290
1000 FOR J=0 TO 3
1010 N=PEEK(50944+C-J)
1020 POKE(50944+C-J+4),N
1030 N=PEEK(50944+C-J-4)
1040 POKE(50944+C-J),N
1050 NEXT J
1060 C=C+4
1070 GOTO 290
1100 FOR J=50944+C0+3 TO 50944+C+3 STEP 8
1110 E=PEEK(J):G=INT(E/16)
1115 IF G/2>INT(G/2) THEN 1200
1120 IFA$="[" THEN G=2*INT(G/2):E=16*G+1
1130 IFA$="]" THEN G=2*INT(G/2)+1:E=16*G+15
1150 POKE J,E
1200 NEXT J
1210 C0=C:GOTO 290
1250 INPUT"NOM FICHER",NM$:LN=LEN(NM$)
1300 POKE780,LN:POKE781,8:POKE782,255
1310 SYS65466
1320 FOR I=1 TO LN:POKE831+I,ASC(MID$(NM$,I,1)):NEXT
1330 POKE780,LN:POKE781,832-256*INT(832/256):POKE782,
INT(832/256)
1340 SYS65469
1350 POKE251,0:POKE252,199
1360 AF=50944+C:LF=AF-256*INT(AF/256)
1370 POKE780,251:POKE781,LF:POKE782,AF/256
1380 SYS65496

```

Le programme affiche le jeu de caractères et attend l'entrée de l'expression sous forme phonétique, avec les possibilités de modifications suivantes :

- Chuchotement par le symbole \* en fin d'expression.
- Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou -.
- Création d'une courbe d'intonation par l'utilisation des marqueurs [ (intonation montante), = (intonation constante) et ] (intonation descendante). Chaque marqueur définit la pente considérée entre le marqueur précédent et lui-même. Quelques indications pour la création d'une courbe d'intonation artificielle sont données page 60.

L'expression à créer se développe à partir de l'adresse \$C700. Une fois l'expression prête (ce qui demande quelques instants), le synthétiseur la prononce (ligne 380 POKE252,0: POKE253,199: SYS50688). Il vous est alors possible de corriger (C), de répéter (ESPACE), de recommencer l'entrée (RETURN) ou de sauvegarder votre expression (S). Dans ce dernier cas, le programme vous demande de lui donner un nom et se charge de la commande du lecteur de disquettes. Il enregistre alors la routine de parole et l'expression à la suite. (Pour un lecteur de cassettes, remplacer à la ligne 1300 POKE 781,8 par POKE 781,1).

Pour réutiliser cette expression ultérieurement, il faudra simplement la recharger et incorporer les instructions suivantes dans le programme appelant :

```
10 POKE 54296,15          (Ouvre l'entrée audio externe)
20 POKE 252,0: POKE 253,199 (Adresse expression dans $FC et $FD)
30 SYS 50688              (Prononce l'expression)
```

## 2) Le programme CHANT 64

Ce programme a pour but la création d'une expression chantée en utilisant les principes expliqués page 65 . Il est dérivé du programme "PHONEM-64" avec lequel il possède des parties communes importantes. Il ne fonctionne correctement qu'avec les phonèmes 4.2 (Trame constante de 16 ms).

```
1 PRINT "C" : PRINT "E"
2 POKE 53280,5 : POKE 53281,7
3 PRINT "AAAAA"
4 IFFLAG=1 THEN G
5 PRINT "          CHANT 64  (C) H. BENOIT 1985"
6 PRINT "          *****"
```

```

7 PRINT"*****" PRINT" CHARGEMENT VOCABULAIRE EN COUPS"
8 FLAG=1 LOAD"PH4.25PEECH",8,1
9 POKE53280,4:POKE53281,2
10 PRINT"0" PRINT"00"
20 PRINT" (CHANT 64 (C) H.BENOIT 1985"
30 PRINT" *****" PRINT"0"
50 POKE54296,15
60 PRINT" A,E,I,O,U,&(AI), (EH),Q(ON),W(OU)" PRINT
70 PRINT" 1(IN),2(EU),3(OI),4(AN),<(IEN),>(OIN)" PRINT
80 PRINT" B,D,F,G,J,K(C,Q),L,M,N,P,R,↑,S,T,V,Z" PRINT
90 PRINT" #<(CH),Y<(GH),@<(AIL),#<(EIL),%<(EUIL)" PRINT
102 PRINT" FA2=0..SI2=3 D03=4..SI3=A D04=B..FA4=F":PRINT
104 PRINT" MULTIP. DUREE 0,5 1 2 4 (0 1 2 3)":PRINT
106 PRINT"0 1E L=TEXTE 2E L=NOTE 3E L=DUREE ":PRINT
"0"
110 INPUTL$:INPUTM$:INPUTF$:L=LEN(L$)
140 C=4 C0=C
150 FOPI=1TOL
160 A#=MID$(L$,1,1):H#=MID$(M$,1,1)
165 FD#=MID$(F$,1,1):FD=VAL(FD$):F0=F
170 IFA#="" THENA#=""
180 READB$,M
181 IFA#<>B$THEN180
182 IFN#="" THENF=88
184 IFN#="1" THENF=98
186 IFN#="2" THENF=110
188 IFN#="3" THENF=124
190 IFN#="4" THENF=130
192 IFN#="5" THENF=146
194 IFN#="6" THENF=164
196 IFN#="7" THENF=174
198 IFN#="8" THENF=196
200 IFN#="9" THENF=220
202 IFN#="A" THENF=246
204 IFN#="B" THENF=262
206 IFN#="C" THENF=294
208 IFN#="D" THENF=330
210 IFN#="E" THENF=350
212 IFN#="F" THENF=392
216 IFI=1THENPOKE50947,F/2:F0=F
220 A=192+PEEK(49152+2*M)
230 B=PEEK(49152+2*M+1)
240 Y=PEEK(256*A+B+1)
250 FORJ=4TOK
252 N=PEEK(256*A+B+J)
254 POKE50944+C+J-4,N
256 NEXTJ
260 DF=F-F0:P1=DF/2:P2=0:P3=0
262 IFF1=0THEN280
264 IFF1>15THENP1=15:P2=(DF/2)-15:IFF2>15THENP2=15:P3=
(DF/2)-30:IFF3>15THENP3=15
266 IFF1<0THENP1=32+P1:IFF1<17THENP1=17 P2=(DF/2)+47:
IFF2<17THENGOSUB465
268 Z=PEEK(50947+C):G=INT(Z/16):IFG/2>INT(G/2)THENF=F0:
GOTO280

```

```

270 Z=Z+P1 POKE50947+C,Z
272 Z=PEEK(50951+C) : G=INT(Z/16) : IF G/2>INT(G/2) THEN F=FO :
POKE50947+C,Z GOTO280
274 Z=Z+P2 POKE50951+C,Z
276 Z=PEEK(50955+C) : G=INT(Z/16)
278 Z=Z+P3 POKE50955+C,Z
280 FOR J=15 TO K STEP 4
282 Z=PEEK(50940+C+J) : G=INT(Z/16) : E=Z-16*G : H=INT(G/8) :
L=G-2+INT(G/2)
284 G=8*H+2+FO+L : IF E<>0 THEN G=8*H+2+L
286 Z=E+16*G : POKE50940+C+J,Z
288 NEXT J
340 RESTORE : C=C+K-4
350 NEXT I
360 POKE50944,INT(C/256)
370 POKE50945,C-256*INT(C/256)
380 POKE252,0 : POKE253,199 : SYS50688
390 PRINT : PRINT "CORRIGER (C) SAUVEGARDER (S) REPETER
(C) "
430 GET E# : IF E#="" THEN 430
435 PRINT "I" : PRINT "
440 IF E#=" " THEN PRINT "III" : GOTO 380
450 IF E#="C" THEN PRINT "IIIIII" : GOTO 110
455 IF E#="S" THEN PRINT "I" : GOTO 1250
460 PRINT : RESTORE : GOTO 10
465 P2=17 : P3=(DF/2)+62 : IF P3<17 THEN P3=17 : RETURN
470 DATA 0,E,1,I,2,O,3,U,4,&,5,/,6,2,7,W,8,4,9,1,10,Q,
11,3,12,B,13,D,14
480 DATA J,17,K,18,L,19,M,20,N,21,P,22,R,23,†,24,S,25,T,
26,V,27,Z,28,*,29
490 DATA @,31,#,32,%,33,€,34,>,35,0,36,;,38,...,39,G,16,Y,
30,*,39,F,15
1250 INPUT "NOM FICHER" : NM# : LN=LEN(NM#)
1300 POKE780,LN : POKE781,8 : POKE782,255
1310 SYS65466
1320 FOR I=1 TO LN : POKE831+I,ASC(MID$(NM#,I,1)) : NEXT
1330 POKE780,LN : POKE781,832-256*INT(832/256) : POKE782,
INT(832/256)
1340 SYS65469
1350 POKE251,0 : POKE252,198
1360 AF=50944+C : LF=AF-256*INT(AF/256)
1370 POKE780,251 : POKE781,LF : POKE782,AF/256
1380 SYS65496

```

Le chant est obtenu en déterminant pour chaque syllabe un pitch de fréquence égale à celle de la note recherchée, et une durée proche de celle indiquée par la partition. Afin de simplifier le programme, ceci a été réalisé en créant 3 chaînes de caractères pour représenter l'expression, chacune ayant un caractère par phonème, et entrées sur 3 lignes d'écran successives :

- L\$ représente la suite de phonèmes composant l'expression (TEXTE)
- M\$ représente la note attribuée à chaque phonème (NOTE)
- F\$ représente la durée relative de chaque phonème (DUREE)

Le programme ignore dièses et bémols, mais rien n'interdirait de les rajouter à la liste des notes ; celle-ci en comporte 16 (de FA<sub>2</sub> = 88 Hz à SOL<sub>4</sub> = 392Hz), soit un peu plus de 2 octaves.

Le pitch initial est égal à la fréquence de la première note (ligne 216). En raison du codage en valeur relative du pitch sur le MEA 8000, chaque note doit être calculée par rapport à la précédente ; c'est ce que font les lignes 260 à 278 du programme, en atteignant cette note le plus rapidement possible (de 0 à 3 trames selon l'écart entre 2 notes successives ; cet écart est limité à 90 Hz dans cet intervalle de temps). En cas de phonème non-voisé, le calcul de la note est reporté au phonème suivant.

La durée est variable dans un rapport 0,5 à 4 par rapport à la durée nominale du phonème, soit un rapport 8 entre extrêmes. Ceci est obtenu en modifiant la durée de toutes les trames du phonème, sauf pour les 3 premières qui sont utilisées pour calculer la note (lignes 280 à 288). Les possibilités de répétition, correction, sauvegarde sont les mêmes que pour le programme "PHONEM-64".

En pratique, pour utiliser ce programme, il faut :

- a) entrer le texte sous forme phonétique.
- b) faire correspondre une note à chaque syllabe, généralement composée d'une ou plusieurs consonnes et voyelles. La même note devra être attribuée à chacun des caractères (phonèmes) composant la syllabe.
- c) faire correspondre une durée à chaque note. La ou les voyelles de la syllabe constituant la partie caractéristique de la note, c'est donc sur elles que la modification de durée doit essentiellement porter. S'il y a plusieurs voyelles consécutives dans la syllabe, répartir la durée entre elles.

Exemple d'utilisation pour le refrain "Au clair de la lune" :

Texte	Au	clair	de	la	lu	-	ne
Notes	do	do	do	ré	mi	ré	
Durées	Nr	Nr	Nr	Nr	Bl	Bl	

(Nr = noire, Bl = blanche)

Il faudra entrer les chaînes de caractères :

Texte	O	K	L	&	R	D	E	L	A	L	U	N	E
Notes	4	4	4	4	4	4	5	5	6	6	5	5	
Durées	1	1	1	1	1	1	1	1	1	2	1	2	

L'expression chantée ainsi créée occupe exactement le même espace mémoire que l'expression phonétique brute ; on peut la sauvegarder et la réutiliser de la même manière qu'avec "PHONEM-64".

### 3°) Programme L.M. de concaténation d'éléments de vocabulaire

Ce programme débute à l'adresse \$C800; il permet la prononciation enchaînée de plusieurs éléments de vocabulaire en mémoire (phonèmes, mots, bruits etc.), sans avoir à créer d'expression à partir de ces éléments. Leur table d'adresse doit débiter à \$C000 et le vocabulaire peut s'étendre jusqu'à \$C7FF; chaque élément doit avoir une longueur inférieure à 256 octets.

Ceci permet une prononciation immédiate de l'expression recherchée à partir de la lecture du numéro d'ordre de chacun des éléments la composant, préalablement chargés à la suite les uns des autres à partir de l'adresse \$CF00; l'expression doit se terminer par le nombre \$2A (42). Ce chargement peut s'effectuer dans un programme BASIC, qui peut également faire une traduction des codes ASCII en numéros de phonèmes si on désire une entrée phonétique.

Aucune modification d'intonation ou d'autres paramètres n'est possible dans ce cas. Le listing commenté de ce programme, présenté ci-dessous, sera aisément adaptable à une autre machine équipée du 6502 en changeant les adresses spécifiques au COMMODORE.

#### Programme langage machine permettant la concaténation de plusieurs éléments.

Adr.	Op.code	Etiqu.	Mnémo.	Commentaire
<b>Programme principal</b>				
C800	20 52 C8	DEBU:	JSR TREQ	Test de REQ
C803	A9 1A		LDA \$1A	Mode arrêt lent
C805	8D FF 9F		STA 9FFF	Envoi commande STOP
C808	20 52 C8		JSR TREQ	Test de REQ
C80B	A9 3C		LDA \$3C	Pitch initial = 120 Hz
C80D	8D FE 9F		STA 9FFE	Envoi pitch initial
C810	A2 00		LDX \$00	Pointage 1er code
C812	BD 00 CF	LECT:	LDA CF00,X	Lecture N° code
C815	C9 2A		CMP \$2A	Code = Hex.2A (décim.42) ?
C817	D0 01		BNE SUIV	Sinon, élément suivant
C819	60		RTS	Si oui, fin de l'expression
C81A	0A	SUIV:	ASL	Accu = code × 2
C81B	A8		TAY	Y = index d'adr. d'adr. élém.
C81C	B9 00 C0		LDA C000,Y	Adr. d'adr. élém. = C000 + Y
C81F	69 C0		ADC \$C0	P. fort adr. = (C000 + Y) + C0
C821	85 FD		STA FD	P. fort adr. dans FD
C823	C8		INY	Pointe poids faible adresse

C824	B9 00 C0	LDA C000,Y	Poids faible dans Accu.
C827	85 FC	STA FC	P. faible adr. dans FC
C829	20 34 C8	JSR SPEECH	Vocalisation élément
C82C	E8	INX	Pointage élément suivant
C82D	4C 12 C8	JMP LECT	Lecture code élém. suivant
C830	EA	NOP	
C831	EA	NOP	
C832	EA	NOP	
C833	EA	NOP	

### Sous-programme de vocalisation (SPEECH)

C834	A0 01	SPEECH:LDY \$01	Index=1 (longueur p. faible)
C836	B1 FC	LDA (FC),Y	Acc = longueur élément
C838	8D FF CF	STA CFFF	CFFF = longueur élément
C83B	A0 04	LDY \$04	Index = 4 (première trame)
C83D	20 52 C8	NEXT: JSR TREQ	Test de REQ
C840	20 5A C8	JSR TRAM	1e octet trame + incr.Y
C843	20 5A C8	JSR TRAM	2e octet trame + incr.Y
C846	20 5A C8	JSR TRAM	3e octet trame + incr.Y
C849	20 5A C8	JSR TRAM	4e octet trame + incr.Y
C84C	CC FF CF	CPY CFFF	Index=Longueur? (Y = L?)
C84F	30 EC	BMI NEXT	Si NON, tr. suivante (NEXT)
C851	60	RTS	Si OUI fin élément

### Sous-programme de test du bit d'état (REQ)

C852	A9 80	TREQ: LDA \$80	Acc = 80H (1000 0000)
C854	CD FF 9F	CMP 9FFF	Mot état = Acc?(REQ = 1?)
C857	D0 F9	BNE TREQ	Si NON, recommencer
C859	60	RTS	

### Sous-programme d'envoi d'un octet de données vocales (TRAM)

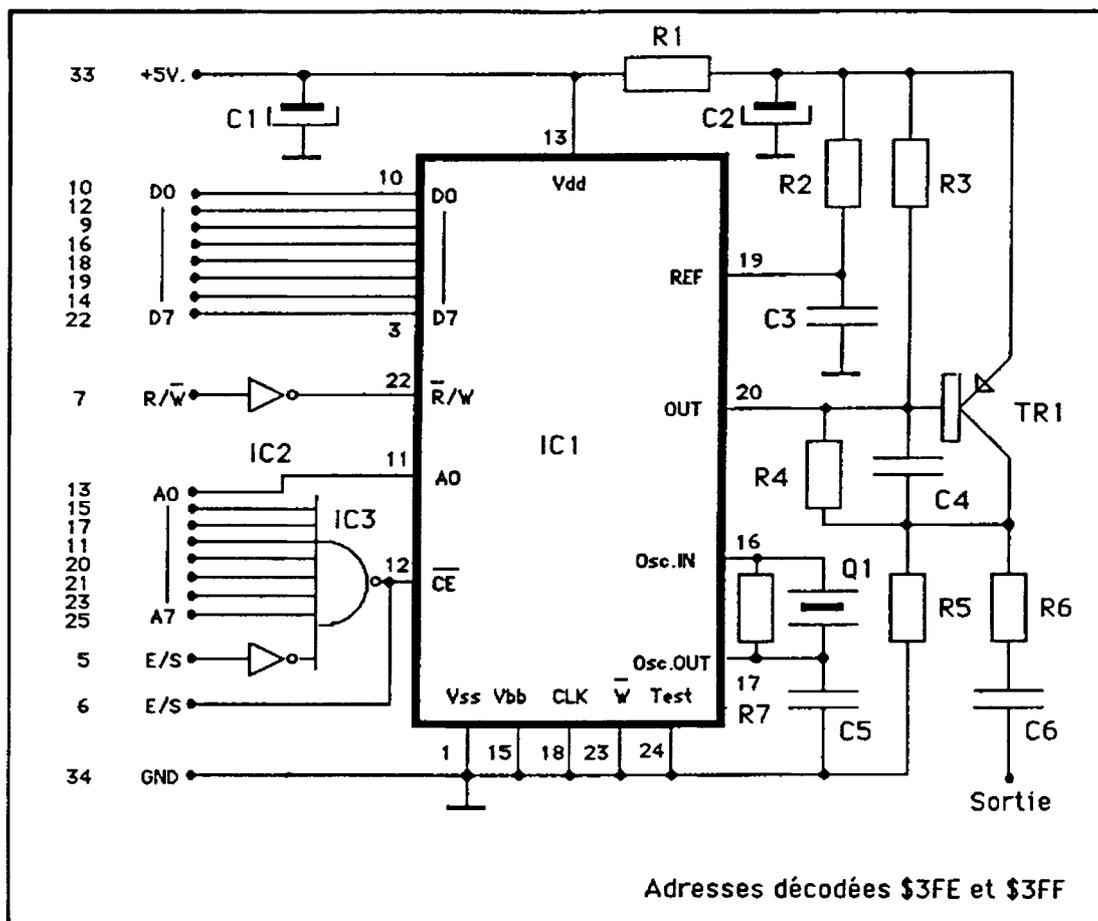
C85A	B1 FC	TRAM: LDA (FC),Y	Acc = octet à envoyer
C85C	8D FE 9F	STA 9FFE	Envoi donnée vocale(A0=0)
C85F	C8	INY	Incrément. index(Y = Y + 1)
C860	60	RTS	

Le programme commence par initialiser le synthétiseur en mode "ARRET LENT", puis envoie un pitch initial de 120 Hz au synthétiseur (cette valeur peut être modifiée par l'octet d'adresse \$C80C qui vaut

PT/2). Le programme prononce ensuite chacun des éléments sans envoyer de commande "STOP" ni de pitch initial entre chacun d'eux pour éviter toute discontinuité (impératif pour concaténer des phonèmes). L'expression s'arrête automatiquement sans envoi de commande "STOP" à la détection du code final (\$2A) grâce au mode "ARRET LENT".

## ORIC 1 et ATMOS

Sur cet ordinateur, le synthétiseur est connecté par l'intermédiaire d'un câble plat au BUS d'expansion (connecteur à 34 voies). Son schéma est représenté ci-dessous et la valeur des composants est donnée par la nomenclature p. 118. Le brochage du connecteur est indiqué dans la notice ORIC. Le synthétiseur occupe deux adresses de la zone réservée aux entrées/sorties; il s'agit des adresses #03FE (données) et #03FF (commandes). La sortie audio peut être envoyée sur le haut-parleur interne (sur prise DIN sortie HiFi) ou sur un amplificateur externe (Péritel par exemple).



Le dictionnaire de phonèmes est à charger à partir de l'adresse #A000. Les versions 3.3 ou 4.2, dont le listing se trouve en annexe, peuvent être utilisées indifféremment, étant donné l'espace-mémoire disponible avant la routine L.M.

Le programme en langage-machine (SPEECH) occupe les adresses #A700 à #A748; il est réalisé selon le modèle expliqué au chapitre "Logiciel de commande" page 50, et son listing hexadécimal est fourni ci-dessous. Les adresses en page zéro utilisées pour l'adressage indirect post-indexé sont #64 et #65.

```
#A700 20 2D A7 20 35 A7 A2 00 A0 03 20 3B A7 20 2D A7 1315
#A710 20 3B A7 20 3B A7 20 3B A7 20 3B A7 CC01 A8 D0 1613
#A720 ECEC00 A8 30 E7 20 2D A7 20 35 A7 60 A9 80 CD 2013
#A730 FF 03 D0 F9 60 A9 1A 8D FF 03 60 B1 64 8D FE 03 2176
#A740 C8 C0 00 D0 03 E8 E6 65 60 55 55 55 55 55 55 1857
```

Les codes des phonèmes et de la routine L.M. pourront être entrés en mémoire au moyen du petit programme ci-dessous, qui assure une vérification de la somme de 16 octets consécutifs.

```
10 CLS:REM***CHARGEUR HEXA***
12 INPUT"ADRESSE DE DEBUT";AD
14 INPUT"ADRESSE DE FIN";AF
20 FORR=AD TO AF STEP 16
30 PRINT RIGHT$(HEX$(R),4);:INPUT T$
40 IF LEN(T$) <> 32 THEN 200
50 INPUT "SOMME ";S:C=0
70 FORT=0 TO 15
80 K=VAL("#" + MID$(T$,2*T+1,2))
90 POKE R+T,K:C=C+K
100 NEXT T
110 IF S <> C THEN 200
120 PRINT CHR$(11)"          EXACT      "
130 NEXT R
200 PRINT:PRINT "          CHR$(27)".. "
210 PRINT "ERREUR":GOTO 30
```

Sauvegarder l'ensemble phonèmes + routine L.M. par l'instruction:

**CSAVE"PHONESPEECH",A#A000,E#A748**

Après avoir chargé l'ensemble "PHONESPEECH" et le programme BASIC appelé "PHONETORIC", dont le listing est fourni ci-après, celui-ci attend dès son lancement l'entrée de l'expression, avec les possibilités suivantes:

— Chuchotement par le symbole ' en fin d'expression.

— Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou -.

— Création d'une courbe d'intonation globale par l'utilisation de la ponctuation finale: / (intonation montante), \ (intonation descendante), ? (intonation descendante, plate, puis montante) et ! (courbe inverse de la précédente).

```
2 REM***** (C) H.BENOIT 1985 *****
4 PRINTCHR$(20)
6 D=#A0:DE=#A000
8 CLS:INK2:PAPER0
9 PRINT:PRINTCHR$(4);CHR$(27);"J                PHONETORIC";
  CHR$(4):PRINT:PRINT
10 PRINT"COMPOSEZ VOTRE EXPRESSION AU MOYEN DES SYMBOLES
  PHONETIQUES":PRINT
11 PRINT"a,e,i,o,u,&(ai),*(eh),E(eu),w(ou)":PRINT
12 PRINT"A(an),I(in),O(on),W(oi),<(ien),>(oin)":PRINT
13 PRINT"b,d,g,j,k,(c),l,m,n,p,r,s,t,v,z":PRINT
14 PRINT"@(a1),#(e1),%(eu1),-(S11),=(S12)":PRINT
15 PRINT"PONCTUATION OBLIGATOIRE . ' ! ? \ /":PRINT:PRINT
16 INPUTL$:L=LEN(L$)
18 POKE#A803,60
19 C=4
20 FORI=1TOL-1
25 A$ =MID$(L$,1+I,1)
26 IFA$="--"THEN700
27 IFA$="+ "THEN600
28 IFA$=" "THENA$=";"
29 IFA$="."THENA$="="
30 READB$,M
40 IFB$<>A$ THEN30
60 A=D+(PEEK(DB+2*M)):REM-HH ADR.SON
70 B=PEEK(DB+2*M+1):REM-LL ADR.SON
72 K=PEEK(256*A+B+1):REM-LONG.SON
84 FORJ=4TOK:REM-TRANSFERT CODE VOCAL
86 N=PEEK(256*A+B+J):REM-SUN ->
87 POKE#A800+C+J-4,N:REM- ->MOT
88 NEXTJ:RESTORE:C=C+K-4
89 NEXTI:PRINT
90 T=C+4:REM-T=LONG.EXPR.
91 POKE#A800,INT(T/256)
92 POKE#A801,T-256*INT(T/256)
93 IFRIGHT$(L$,1)="\"THENGOSUB220
94 IFRIGHT$(L$,1)="/"THENGOSUB270
95 IFRIGHT$(L$,1)="!"THENGOSUB200
96 IFRIGHT$(L$,1)="?"THENGOSUB250
97 IFRIGHT$(L$,1)="' "THENGOSUB300
99 POKE#65,168:POKE#64,0:CALL#A700
100 PRINT"PRESSER 0 POUR TERMINER"
102 PRINT"PRESSER 1 POUR SAUVEGARDER"
105 PRINT"ESPACE FOUR REPETER"
106 PRINT"AUTRES TOUCHES FOUR RECOMMENCER"
110 GETE$:IFE$=" "THEN99
115 IFE$="1"THEN500
120 IFE$="0"THENPRINTCHR$(20):END
```

```

130 PRINT:RESTORE:GOTO8
140 DATAa,0,e,1,i,2,o,3,u,4,&,5,* ,6,E,7,w,8,A,9,I,10,O,11,
W,12,h,13,d,14,f,15
150 DATAg,16,j,17,k,18,l,19,m,20,n,21,p,22,R,23,r,24,s,25,
t,26,v,27,z,28,$,29
160 DATA N,30,@,31,#,32,%,33,<,34,>,35,0,36,;,38,=,39,/ ,39
,\,39,?,39,! ,39
170 END
200 POKE#A803,64:FORQ=#A800+T-1TO#A800+T/2STEP-4*INT(T/48)
201 GOSUB405:NEXTQ
202 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)
203 GOSUB455:NEXTQ:RETURN
220 POKE#A803,64:FORQ=#A800+T-1TO#A800+T/2STEP-4*INT(T/48)
221 GOSUB405:NEXTQ
222 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)
223 GOSUB405:NEXTQ:RETURN
250 POKE#A803,56:FORQ=#A800+T-1TO#A800+T/2STEP-4*INT(T/48)
251 GOSUB455:NEXTQ
252 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)
253 GOSUB405:NEXTQ:RETURN
270 POKE#A803,64:FORQ=#A800+T-1TO#A800+T/2STEP-4*INT(T/48)
271 GOSUB455:NEXTQ
272 FORQ=#A807TO#A800+T/2STEP4*INT(T/24)
273 GOSUB455:NEXTQ:RETURN
300 FORQ=#A807TO#A800+T-1STEP4
305 E$=HEX$(PEEK(Q))
307 G=VAL(LEFT$(E$,2))
310 R=VAL(RIGHT$(E$,1))
315 IFR=0ANDG>2*INT(G/2)THEN330
318 G=2*INT(G/2)+1:E=16*G
325 POKEQ,E
330 NEXTQ
335 RETURN
405 E$=HEX$(PEEK(Q))
407 G=VAL(LEFT$(E$,2))
410 R=VAL(RIGHT$(E$,1))
415 IFR=0ANDG>2*INT(G/2)THEN435
418 G=2*INT(G/2)+1:E=16*G+15
425 POKEQ,E
435 RETURN
455 E$=HEX$(PEEK(Q))
457 G=VAL(LEFT$(E$,2))
460 R=VAL(RIGHT$(E$,1))
465 IFR=0ANDG>2*INT(G/2)THEN485
468 G=2*INT(G/2)+1:E=16*G+1
475 POKEQ,E
485 RETURN
500 CLS:PRINTCHR$(20):PRINT
520 INPUT"NOM DU FICHIER ";N$
530 CSAVEN$,A#A700,E#A800+T
540 END
600 FORJ=0TO3
610 N=PEEK(#A800+C-J)
620 POKE(#A800+C+4-J),N
630 N+PEEK(#A800+C-J-4)
640 POKE(#A800+C-J),N
650 NEXTJ
660 C=C+4

```

```
670 GOTO89
700 FORJ=0TO3
710 N=PEEK(#A800+C-J-4)
720 POKE(#A800+C-J),N
730 NEXTJ
740 C=C-4
750 GOTO89
```

L'expression à créer se développe à partir de l'adresse #A800, et peut aller jusqu'à #B3FF. Une fois l'expression créée (ce qui dure quelques instants), le synthétiseur la prononce (ligne 99 POKE#65,168: POKE#64,0: CALL#A700). Il vous est alors possible de répéter (ESPACE), de terminer (0), de sauvegarder votre expression (1) (dans ce dernier cas, le programme vous demande de lui donner un nom et enregistre sur cassette la routine "SPEECH" et l'expression à la suite l'une de l'autre), ou de recommencer l'entrée (autres touches).

Pour réutiliser cette expression dans un programme BASIC, il suffit de la charger en mémoire et d'incorporer la ligne 99 dans ce programme.

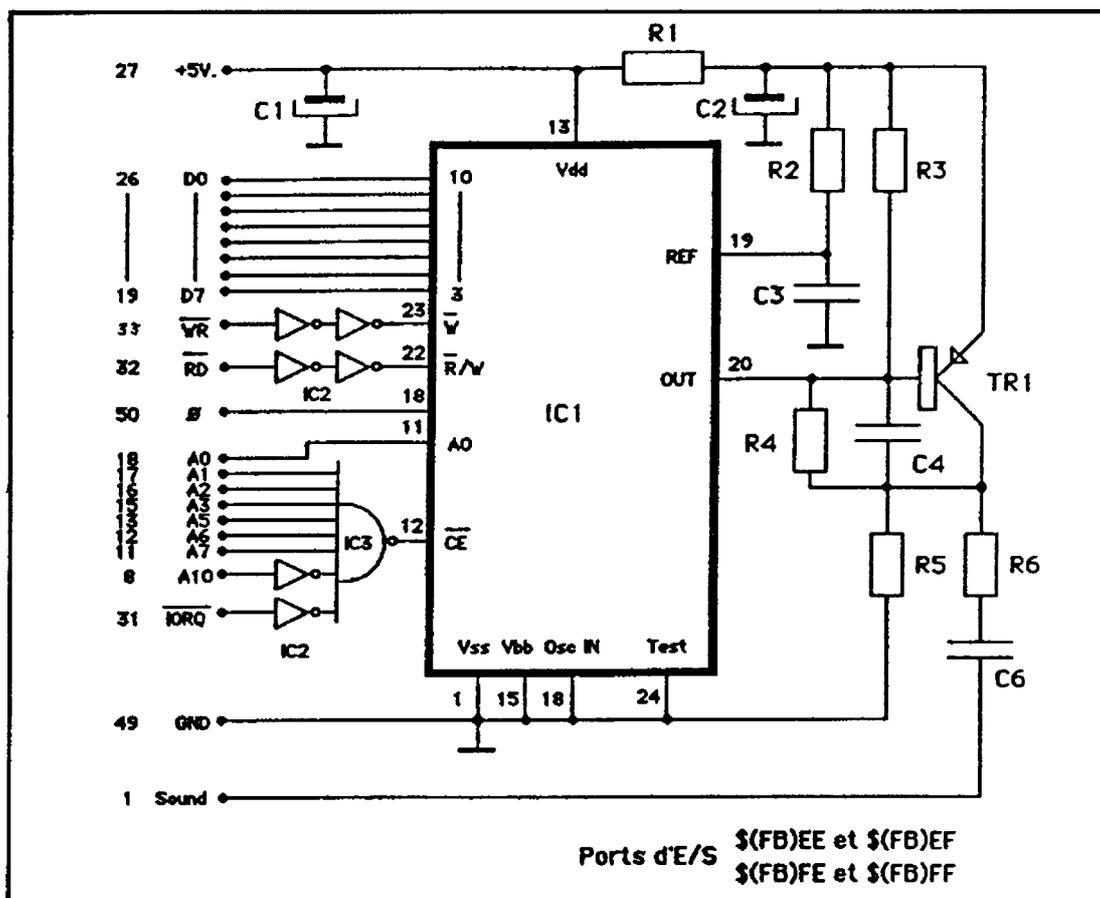
# Applications sur AMSTRAD, MSX, VG 5000, MZ-700 (Z-80A)

Dans cette rubrique, nous décrivons une ou plusieurs applications sur chacune des machines suivantes: AMSTRAD CPC, MSX (toutes marques), VG 5000 (Philips, Radiola, Schneider), SHARP MZ-700.

La transposition à d'autres machines avec Z-80 sera aisée: elle consistera essentiellement à adapter les ports d'entrée-sortie utilisés par le MEA 8000 à une paire non utilisée par la machine, ainsi que l'emplacement du vocabulaire et de la routine de commande du synthétiseur en mémoire.

## AMSTRAD CPC 464/664/6128

Le synthétiseur est relié par un connecteur femelle à 50 contacts au bus d'extension. Il utilise l'horloge 4 MHz disponible sur le bus, ce qui évite un quartz. Son schéma est présenté ci-dessous et sa nomenclature est à la page 118.



La sortie audio du synthétiseur, reliée à l'entrée SOUND, active le haut-parleur interne de l'AMSTRAD. En raison de l'utilisation interne des ports d'entrée-sortie du Z-80, seules les instructions d'entrée-sortie qui placent le registre B sur les adresses A8-A15 sont utilisables et les adresses suivantes sont réservées à l'utilisateur: &F8E0 à &F8FF, &F9E0 à &F9FF, &FAE0 à &FAFF, &FBE0 à &FBFF. Ceci nécessiterait théoriquement un décodage d'adresse sur 16 bits. Un décodage simplifié des 8 bits de poids faible est cependant possible, d'après la notice AMSTRAD, à condition de décoder l'état bas de A10 simultanément (ceci évite un déclenchement intempestif par les commandes d'imprimante). Ne décodant pas le bit A4, le synthétiseur répondra aux adresses de poids faible &EE ou &FE (données) ainsi que &EF ou &FF (commandes).

Le dictionnaire de phonèmes est à charger à partir de l'adresse &4000. La version 4.2, dont le listing se trouve en annexe, est à utiliser pour bénéficier de toutes les fonctionnalités du programme; elle s'étend jusqu'à l'adresse &45B7 incluse. Leurs codes pourront être entrés en mémoire par exemple au moyen d'un petit programme BASIC vérifiant la somme de 16 octets consécutifs (voir "chargeur hexa." proposé pour l'ORIC p.132). Sauvegarder les phonèmes à la suite du programme principal en BASIC dont le listing est ci-dessous, par la commande:

**SAVE"!PHONEMES", B,&4000,&5B8 (! = chargement automatique)**

```

1 CLS:MODE 1:MEMORY &3FFF
5 LOCATE 6,12:PRINT "ASTRAPHON (c) H. BENOIT 1985":LOAD
  "PHONEMES",&4000
168 REM routine lancée machine-----
169 RESTORE 170:FOR I=&45EE TO &45FF:READ byte:POKE I,byte:
  NEXT
170 DATA &2a,&fe,&45,&56,&23,&5e,&23,&23
171 DATA &1b,&1b,&1b,&cd,&eb,&45,&3e,&1a
172 DATA &ed,&79,&cd,&f5,&45,&cd,&eb,&45
173 DATA &cd,&f5,&45,&cd,&f5,&45,&cd,&f5
174 DATA &45,&cd,&f5,&45,&af,&bb,&20,&ed
175 DATA &ba,&2c,&ea,&cd,&eb,&45,&3e,&1a
176 DATA &ed,&79,&c9,&01,&ff,&fb,&ed,&78
177 DATA &cb,&7f,&2e,&f7,&c9,&01,&fe,&fb
178 DATA &7e,&ed,&79,&23,&1b,&c9,&c0,&46
200 LOCATE 1,9:MODE 2
250 CLS:PRINT "  ASTRAPHON (C) H.BENOIT 1985":PRINT:PRINT
260 PRINT "a,e,l(et),j(ai),i,o(eau),o(o de or),u,t(eu),w
  (ou),A(an)":PRINT
270 PRINT "I(in),U(ou),W(oi),b,d,f,g(gue),j,k,l,m":PRINT
280 PRINT "n,p,r,r,s,t,v,z,$(ch),N(en),@(ail),&(eil),%
  (euil)":PRINT
290 PRINT "SIL ./ DURCE +- INTON <=> VIT !#":PRINT
300 PRINT "PONCTUATION FINALE OBLIGATOIRE , *":PRINT:PRINT
310 PRINT "Composez votre phrase avec les symboles
  phonétiques ci-dessus":PRINT
340 INPUT L$:L=LEN(L$)

```

```

350 CLS:LOCATE 1,1:PRINT "Veuillez patienter S.V.P."
370 V$=LEFT$(L$,1):W$=MID$(L$,2,1)
379 REM evaluation phoneme -----
380 C=4:C0=0:FOR I=1 TO L          ' C=decalage initial
390 A$=MID$(L$,I,1)
400 IF A$=" " THEN A$="."
410 IF A$="." THEN C0=C:GOTO 580
420 IF A$="<" THEN 820
430 IF A$=">" THEN 820
440 IF A$="+" THEN 890
450 IF A$="-" THEN 970
455 RESTORE 750
460 READ B$,M
470 IF B$<>A$ THEN 460
479 REM position memoire du phoneme -----
480 D=&40:DB=&4000
490 A=D+PEEK(DB+2*M)  ' poids forts de l'adresse du phoneme
500 B=PEEK(DB+2*M+1)  ' poids faibles -----
510 K=PEEK(256*A+B+1)  ' recherche longueur phoneme
519 REM concatenation phonemes -----
520 FOR J=4 TO K
530 N=PEEK(256*A+B+J)
540 POKE &45FC+C+J,N
550 NEXT J
560 C=C+K-4          ' longueur resultante
580 NEXT I
589 REM longueur expression -----
590 U=C/256:V=INT(U)
600 POKE &4600,V
610 POKE &4601,C-256*V
620 POKE &4603,60
630 GOSUB 1080
640 IF A$="*" THEN GOSUB 780
649 REM appel au sous-programme Prononciation -----
650 CALL &4500
660 LOCATE 1,3
670 PRINT "      e=reécouter      r=recommencer"
680 PRINT "      s=sauvegarder autres=arrêt"
690 E$=INKEY$:IF E$="" THEN 690
700 IF L$="e" THEN 650
710 IF E$="s" THEN 1030
720 IF E$="r" THEN 250
740 END
750 DATA a,u,e,1,1,2,o,3,i,4,j,5,l,6,c,7,w,8,A,9,l,10,0,11,
      W,12,0,36
760 DATA b,13,d,14,f,15,g,16,j,17,k,18,l,19,m,20,n,21,p,22,
      R,23,r,24,s,25
770 DATA t,26,v,27,z,28,$,29,N,30,@,31,&,32,Z,33,..38,4,38,
      !,38,*,38,/,39
779 REM sous-programme chuchotement -----
780 FOR J=&4607 TO &45FF+C STEP 4
790 E=PEEK(J):G=INT(E/16)
800 IF G/2=INT(G/2) THEN G=G+1:E=16*G:POKE J,E
810 NEXT J:RETURN
819 REM sous-programmes intonation -----
820 FOR J=&4603+C0 TO &4603(C STEP 8
830 E=PEEK(J):G=INT(E/16)
840 IF G/2>INT(G/2) THEN 880

```

```

850 IF A$="<" THEN G=2*INT(G/2):E=16*G+1
860 IF A$=">" THEN G=2*INT(G/2)+1:E=16*G+15
870 POKE J,E
880 NEXT J:C=C:GOTO 580
890 FOR J=0 TO 3
900 N=PEEK(&4600+C-J)
910 POKE &4604+C-J,N
920 N=PEEK(&45FC+C-J)
930 POKE &4600+C-J,N
940 NEXT J
950 C=C+4
960 GOTO 580
970 FOR J=0 TO 3
980 N=PEEK(&4600+C-J)
990 POKE &45FC+C-J,N
1000 NEXT J
1010 C=C-4
1020 GOTO 580
1029 REM sauvegarde -----
1030 CLS:PRINT L$
1040 INPUT "NOM DU FICHER":N$
1060 SAVE N$,E,&4500,C+256
1070 GOTO 250
1079 REM sous programme vitesse -----
1080 R=8:IF V$=W$ THEN R=4
1090 FOR J=&4607 TO &45FF+C STEP R
1100 E=PEEK(J)
1110 IF V$="#" THEN E=E-32
1120 IF V$="!" THEN E=E+32
1130 POKE J,E:NEXT J:RETURN

```

Le programme en langage-machine (SPEECH) occupe les adresses &45B8 à &45FD. En raison de l'utilisation interne particulière des ports d'entrée-sortie sur cette machine, il diffère quelque peu du modèle général expliqué au chapitre "Logiciel de commande"; on trouvera son listing assembleur commenté ci-dessous. L'adresse de début de l'expression à prononcer doit être placée aux adresses &45FE (poids faible) et &45FF (poids fort).

### Routine de prononciation d'une expression de longueur quelconque

Adr.	Op.code	Etiqu.	Mnémo.	Commentaire
------	---------	--------	--------	-------------

#### Programme principal

45B8	2A	FE 45DEBU:	LD HL,(45FE)	Adresse expression dans HL
45B9	56		LD D,(HL)	Poids fort longueur dans D
45BC	23		INC HL	HL pointe p. faible longueur
45BD	5E		LD E,(HL)	Poids faible longueur dans E

45BE 23	INC HL	
45BF 23	INC HL	HL pointe pitch initial
45C0 1B	DEC DE	
45C1 1B	DEC DE	
45C2 1B	DEC DE	DE = longueur restante
45C3 CD EB 45	CALL TREQ	Test de REQ
45C6 3E 1A	LD A,1AH	Hex 1A (Arret lent) ds Acc.
45C8 ED 79	OUT (C),A	Initialisation MEA 8000
45CA CD F7 45	CALL TRAM	Pitch initial +inc.HL+dec.DE
45CD CD EB 45	NEXT: CALL TREQ	Test de REQ
45D0 CD F7 45	CALL TRAM	1e octet trame + i.HL + d.DE
45D3 CD F7 45	CALL TRAM	2e octet trame + i.HL + d.DE
45D6 CD F7 45	CALL TRAM	3e octet trame + i.HL + d.DE
45D9 CD F7 45	CALL TRAM	4e octet trame + i.HL + d.DE
45DC AF	XOR A	Acc = 0
45DD BB	CP E	P. faible longueur = 0 ?
45DE 20 ED	JR NZ,NEXT	Si NON, tr. suivante (NEXT)
45E0 BA	CP D	P. fort longueur = 0 ?
45E1 20 EA	JR NZ,NEXT	Si NON, tr. suivante (NEXT)
45E3 CD EB 45	CALL TREQ	Si OUI, test de REQ
45E6 3E 1A	LD A,1AH	Hex.1A dans Accumulateur
45E8 ED 79	OUT (C),A	Commande STOP finale
45EA C9	RET	Fin de l'expression

#### Sous-programme de test du bit d'état (REQ)

45EB 01 FF FB	TREQ: LD BC,FBFFH	Adr. synthé. (cde) dans BC
45EE ED 78	IN A,(C)	Acc = Mot d'état
45F0 CB 7F	BIT 7,A	Test de REQ (D7 = 1 ?)
45F2 28 F7	JR Z,TREQ	Si NON, recommencer
45F4 C9	RET	

#### Sous-programme d'envoi d'un octet de données vocales (TRAM)

45F5 01 FE FB	TRAM: LD BC,FBFEH	Adr. synthé. (don.) dans BC
45F8 7E	LD A,(HL)	Acc = octet à envoyer
45F9 ED 79	OUT (C),A	Envoi donnée vocale (A0=0)
45FB 23	INC HL	Incrément. adresse octet
45FC 1B	DEC DE	Décrem. longueur restante
45FD C9	RET	
45FE 00		Poids faible adr. expression
45FF 46		Poids fort adr. expression

Après avoir chargé le programme BASIC, son exécution commence par le chargement des phonèmes; SPEECH est chargé par les premières lignes de DATA, puis l'ordinateur attend l'entrée de l'expression. Les options suivantes permettent de déterminer la prosodie:

— Variation de la vitesse d'élocution si l'on commence l'expression par un ou deux signes # (plus vite) ou ! (moins vite); ceci a pour effet de diviser ou multiplier la durée (égale à 16 ms avec les phonèmes 4.2) d'une sur deux ou de toutes les trames de l'expression. La vitesse est ainsi variable en 5 paliers dans un rapport de 1 à 4.

— Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou -.

— Création d'une courbe d'intonation par l'insertion de marqueurs qui déterminent les points d'inflexion et la pente de la courbe d'intonation. Le signe < détermine une pente montante, le signe > une pente descendante et le signe = une pente nulle; chaque marqueur agit entre le précédent (ou le début de l'expression s'il est le premier) et lui-même.

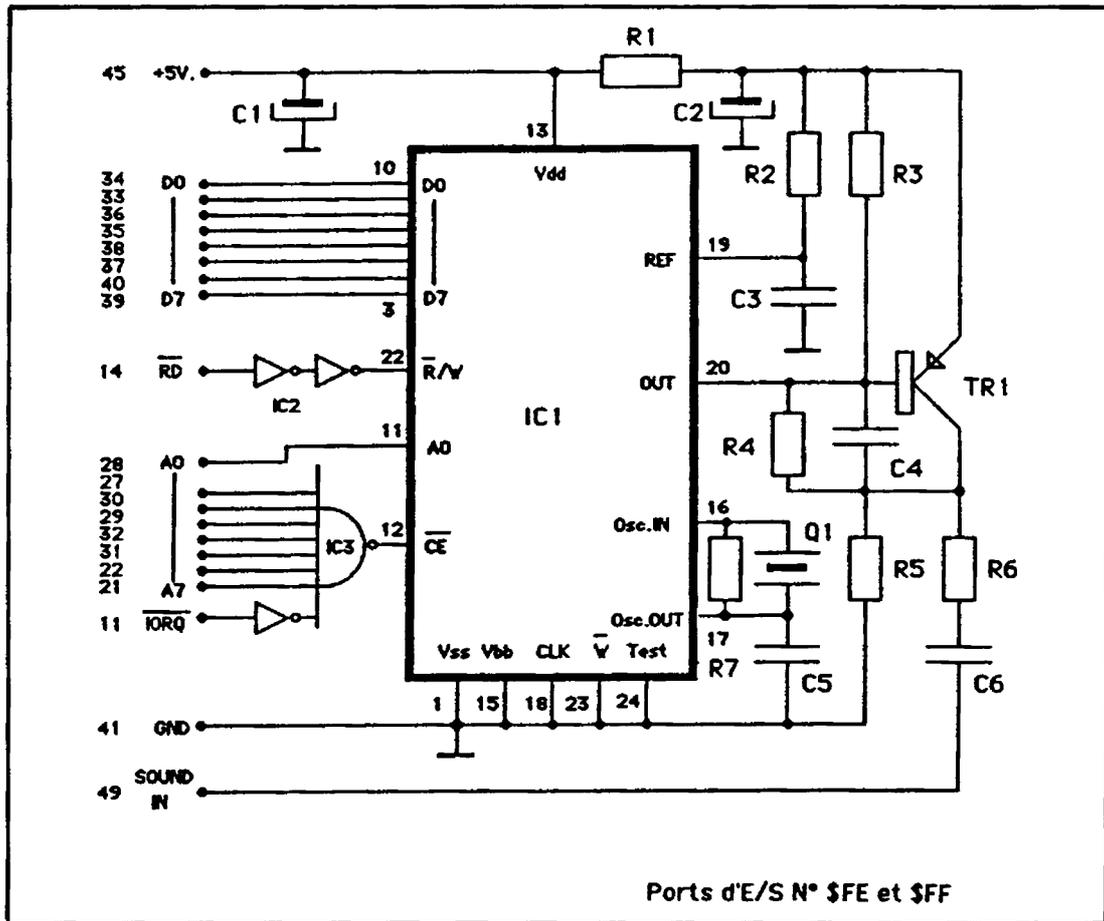
— Chuchotement par le symbole \* en fin d'expression.

L'expression à créer se développe à partir de l'adresse &4600, et pourrait aller théoriquement jusqu'à &ABFF! Une fois l'expression créée (ce qui dure quelques instants), le synthétiseur la prononce (ligne 650 CALL &45B8). Il vous est alors possible de réécouter (E), de recommencer une autre expression (R), de sauvegarder votre expression (S) (dans ce dernier cas, le programme vous demande de lui donner un nom et enregistre sur cassette la routine "SPEECH" et l'expression à la suite l'une de l'autre); l'appui de toute autre touche arrête l'exécution du programme.

Pour réutiliser l'expression sauvegardée dans un programme BASIC, il suffit de la charger et d'incorporer la ligne 650 dans ce programme.

## **Ordinateurs MSX**

Cette application est prévue pour toutes les machines à la norme MSX; elle a été testée sur une machine dotée de 32 K RAM (Philips VG 8010). Le synthétiseur est connecté dans l'un des "slots" destinés aux cartouches d'extension. Son schéma est représenté ci-après et la valeur des composants est donnée par la nomenclature de la page 118. Il occupe les ports d'entrée-sortie &HFE (données) et &HFF (commandes). La sortie audio du synthétiseur est reliée à l'entrée son externe, et ressort sur la même broche que le son du générateur interne.



Le dictionnaire de phonèmes est à charger à partir de l'adresse &HE000. La version 4.2, dont le listing se trouve en annexe, est à utiliser pour bénéficier de toutes les fonctionnalités du programme; elle s'étend jusqu'à l'adresse &HE5B7 incluse.

Le programme en langage-machine (SPEECH) occupe les adresses &HE5B8 à &HE5FF; il est réalisé selon le modèle expliqué au chapitre "Logiciel de commande" p. 52 , et son listing hexadécimal est fourni ci-dessous.

```

&HE5B8 00 2A FE E5 46 23 4E CDEB E5 CDF2 E5 0B 0B 0B
&HE5C8 23 23 CDF7 E5 CDEB E5 CDF7 E5 CDF7 E5 CDF7
&HE5D8 E5 CDF7 E5 AFB9 20 EDB8 20 EACDEB E5 CDF2
&HE5E8 E5 00 C9 DBFF CB 7F 28 FA C9 3E 1A D3 FF C9 7E
&HE5F8 D3 FE 23 0B C9 00 00 E6

```

Les codes des phonèmes et de la routine L.M. pourront être entrés en mémoire par exemple au moyen d'un programme tel que le "chargeur hexa." proposé pour l'ORIC (voir page 132). Sauvegarder l'ensemble

phonèmes + routine L.M. à la suite du programme principal en BASIC  
présenté ci-dessous par la commande:

**BSAVE"CAS: PHSPCH", &HE000,&HE600**

```

10 CLS: CLEAR&H1300,&HE000: LOCATE0,9
20 KEYOFF: PRINT" PHONEM.S.X (C) H.BENOIT 1985": PRINT:
PRINT: PRINT: PRINT:
30 PRINT" VEUILLEZ PATIENTER S.V.P."
40 BLOAD"CAS:"
50 CLS: PRINT" PHONEM.S.X (C) H.BENOIT 1985": PRINT: PRINT
60 PRINT"a,e,é,è,i,o,0,u,E(eu),w(ou),A(an)": PRINT
70 PRINT"I(in),O(on),W(oi),b,d,t,g,j,k,l,m": PRINT
80 PRINT"n,p,R,r,s,t,v,z,$((h),N(en),à,&,Z": PRINT
90 PRINT"SIL ./ DUREE +- INTON <=> VII !# ": PRINT
100 PRINT" PUNCTUATION FINALE OBLIGATOIRE . *": PRINT: PRINT
110 PRINT" Composez";
120 PRINT" votre phrase avec les sym- "
130 PRINT"boles phoniques ci-dessus "": PRINT
140 INPUTL$: L=LEN(L$)
150 LOCATE0,14: PRINT" Veuillez patienter s'il vous plait "
160 PRINT"
170 V$=LEFT$(L$,1): W$=MID$(L$,2,1)
180 C=4: C0=0: FORI=1TOL
190 A$=MID$(L$,I,1)
200 IFA$=" " THEN A$="."
210 IFA$="=" THEN C0=C: GOTO380
220 IFA$="<" THEN 620
230 IFA$=">" THEN 620
240 IFA$="+" THEN 690
250 IFA$="-" THEN 770
260 READB$,M
270 IFB$<>A$ THEN 260
280 D=&HE0: DB=&HE000
290 A=D+I*EEK(DB+2*M)
300 B=PEEK(DB+2*M+1)
310 K=PEEK(256*A+B+1)
320 FORJ=4TOK
330 N=PEEK(256*A+B+J)
340 POKE&HE5FC+C+J,N
350 NEXTJ
360 C=C+K-4
370 RESTORE
380 NEXTI
390 U=C/256: V=INT(U)
400 POKE&HE600,V
410 POKE&HE601,C-256*V
420 POKE&HE603,60
430 GOSUB880
440 IFA$="*" THEN GOSUB580
450 DEF USR=&HE5B8: Z=USR(0)
460 LOCATE0,14
470 PRINT" c=corriger e=ecouter "
480 PRINT" r=recommencer s=sauvegarder "

```

```

490 E$=INKEY$:IFE$=""THEN490
500 IFE$="e"THEN450
510 IFE$="s"THEN830
520 IFE$="r"THENRESTORE:GOTO50
530 IFE$="c"THENLOCATE0,14:PRINT" Corrigez":GOTO120
540 END
550 DATAa,0,e,1,i,2,o,3,u,4,è,5,é,6,E,7,w,8,A,9,I,10,O,
11,W,12,o,36
560 DATAb,13,d,14,f,15,g,16,j,17,k,18,l,19,m,20,n,21,p,
22,R,23,r,24,s,25
570 DATAt,26,v,27,z,28,$,29,N,30,ã,31,8,32,%,33,..,38,*,
38,!,38,x,38,/,39
580 FORJ=&HE607TO&HE5FF+CSTEP4
590 E=PEEK(J):G=INT(E/16)
600 IFG/2=INT(G/2)THENG=G+1:E=16*G:POKEJ,E
610 NEXTJ:RETURN
620 FORJ=&HE603+C0TO&HE603+CSTEP8
630 E=PEEK(J):G=INT(E/16)
640 IFG/2>INT(G/2)THEN680
650 IFA$="<"THENG=2*INT(G/2):E=16*G+1
660 IFA$=">"THENG=2*INT(G/2)+1:E=16*G+15
670 POKEJ,E
680 NEXTJ:C0=C:GOTO380
690 FORJ=0TO3
700 N=PEEK(&HE600+C-J)
710 POKE&HE604+C-J,N
720 N=PEEK(&HE5FC+C-J)
730 POKE&HE600+C-J,N
740 NEXTJ
750 C=C+4
760 GOTO380
770 FORJ=0TO3
780 N=PEEK(&HE600+C-J)
790 POKE&HE5FC+C-J,N
800 NEXTJ
810 C=C-4
820 GOTO380
830 CLS:PRINTL$
840 INPUT"NOM DU FICHER";N$
850 INPUT"MAGNETO PRET (o/n)";K$
860 IFK$="o"THENBSAVE"CAS:N$",&HE5B8,&HE600+C
870 GOTO50
880 R=8:IFV$=W$THENR=4
890 FORJ=&HE607TO&HE5FF+CSTEPR
900 E=PEEK(J)
910 IFV$="*"THENE=E-32
920 IFV$="!"THENE=E+32
930 POKEJ,E:NEXTJ:RETURN

```

Après avoir chargé le programme BASIC, son exécution commence par le chargement de l'ensemble phonèmes + routine L.M. (PHSPCH), puis l'ordinateur attend l'entrée de l'expression.

La prosodie peut être déterminée par les options suivantes :

— Variation de la vitesse d'élocution si l'on commence l'expression par un ou deux signes # (plus vite) ou ! (moins vite); ceci a pour effet de diviser ou multiplier la durée (égale à 16 ms avec les phonèmes 4.2) d'une sur deux ou de toutes les trames de l'expression. La vitesse est ainsi variable en 5 paliers dans un rapport de 1 à 4.

— Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou -.

— Création d'une courbe d'intonation par l'insertion de marqueurs qui déterminent les points d'inflexion et la pente de la courbe d'intonation. Le signe < détermine une pente montante, le signe > une pente descendante et le signe = une pente nulle; chaque marqueur agit entre le précédent (ou le début de l'expression s'il est le premier) et lui-même.

— Chuchotement par le symbole \* en fin d'expression.

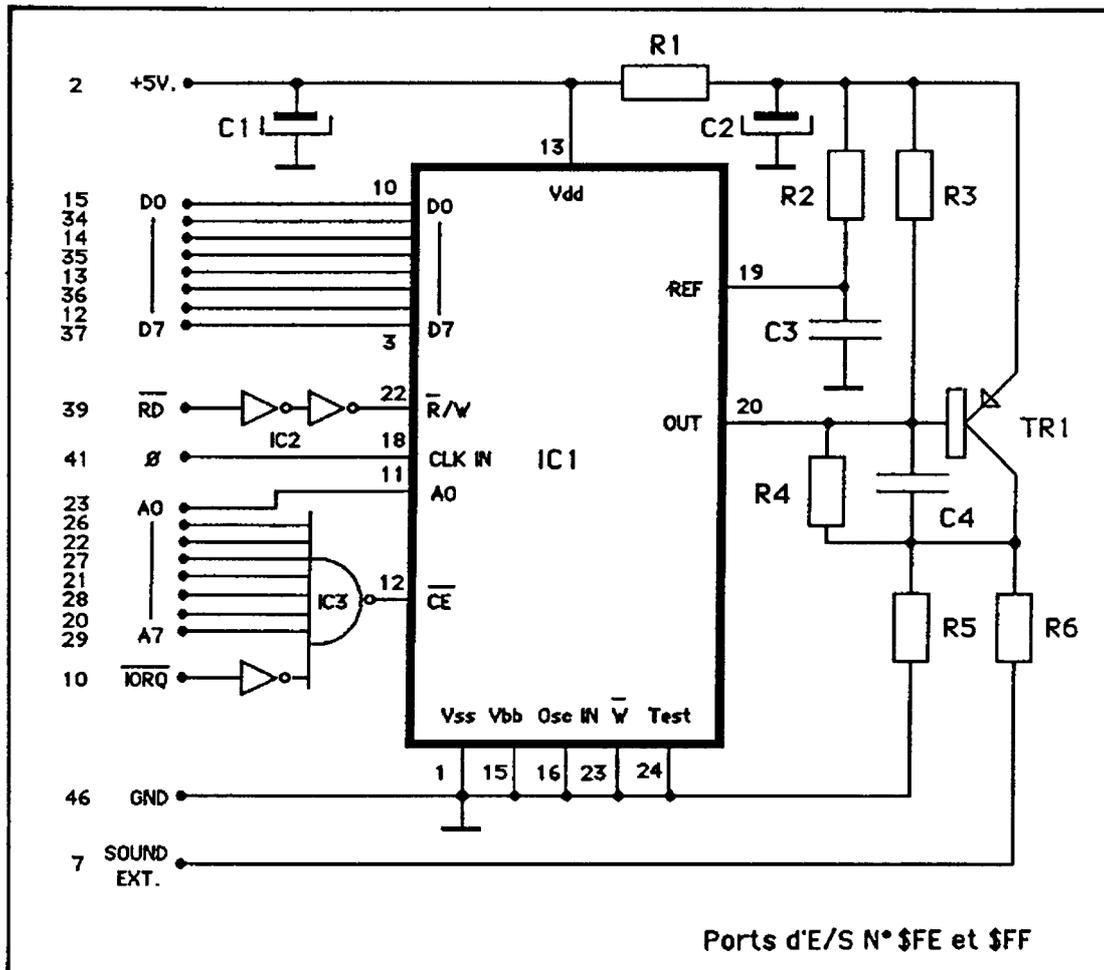
L'expression à créer se développe à partir de l'adresse &HE600, et peut aller jusqu'à &HF37F (zone mémoire de la pile à partir de &HF380). Une fois l'expression créée (ce qui dure quelques instants), le synthétiseur la prononce (ligne 450 DEF USR = &HE5B8: Z = USR(0)).

Il vous est alors possible de corriger (C), de réécouter (E), de recommencer une autre expression (R), de sauvegarder votre expression (S) (dans ce dernier cas, le programme vous demande de lui donner un nom et enregistre sur cassette la routine "SPEECH" et l'expression à la suite l'une de l'autre); l'appui de toute autre touche arrête l'exécution du programme.

Pour réutiliser l'expression sauvegardée dans un programme BASIC, il suffit de la charger et d'incorporer la ligne 450 dans ce programme.

## **PHILIPS VG 5000**

Ce paragraphe décrit l'application sur la version de base (24 K RAM dont 8 pour l'écran vidéo). Le synthétiseur est connecté sur le bus d'extension situé à l'arrière, au moyen d'un connecteur femelle à 50 voies. Il occupe les deux ports d'entrée-sortie "&FE" (données) et "&FF" (commandes). Son schéma est représenté ci-dessous et la valeur des composants est donnée par la nomenclature page 118. Le VG 5000 disposant d'une horloge à 4 MHz sur le BUS, le quartz et les composants associés ont pu être supprimés. La sortie audio du synthétiseur est reliée à l'entrée son externe du connecteur d'extension, et ressort mélangée au son interne sur la prise Péritélévision. Le brochage du connecteur est indiqué dans la documentation fournie avec le VG 5000.



Si l'on dispose de l'extension mémoire, on pourra utiliser le synthétiseur pour MSX (voir chapitre précédent) connecté sur le "slot" de cette extension, et réimplanter éventuellement le vocabulaire et la routine L.M. pour bénéficier de l'extension mémoire.

Le dictionnaire de phonèmes est à charger à partir de l'adresse "&7000". La version 4.2, dont le listing en annexe, est à utiliser pour bénéficier de toutes les fonctionnalités du programme; elle s'étend jusqu'à l'adresse "&757" incluse.

Le programme en langage-machine (SPEECH) occupe les adresses "&75B8" à "&75FF"; il est réalisé selon le modèle expliqué au chapitre "Logiciel de commande" p. 52, et son listing hexadécimal est fourni ci-dessous.

```

75B8  E5 2A FE 75 46 23 4E CDEB 75 CDF2 75 0B 0B 0B
75C8  23 23 CDF7 75 CDEB 75 CDF7 75 CDF7 75 CDF7
75D8  75 CDF7 75 AFB9 20 EDB8 20 EACDEB 75 CDF2
75E8  75 E1 C9 DBFF CB 7F 28 FA C9 3E 1A D3 FF C9 7E
75F8  D3 FE 23 0B C9 00 00 76

```

Sur le VG 5000, l'appel d'une routine L.M. exige de préserver le registre HL (octets "&75B8" = "&E5" PUSH HL et "&75E9" = "&E1" POP HL). Les codes des phonèmes et de la routine L.M. pourront être entrés en mémoire grâce au mini-moniteur (qui permet également d'inspecter la mémoire, voir listing ci-dessous), ou d'un programme de votre cru vérifiant la somme de 16 octets successifs (voire "chargeur hexa." proposé pour l'ORIC p.132).

```

10000 INPUT"MODE";M$
10010 IFM$="D"THEN 10040
10020 IFM$="M"THEN 10240
10030 IFM$="F"THENEND
10040 INPUT"ADRESSE DEBUT";AD:GOTO 10060
10050 AD=AD+128:INPUT"CONT";C$:IFC$<>" "THEN 10000
10060 AF=AD+120
10070 FORA=ADTOAFSTEP8:GOSUB 10280
10080 FORI=0TO7
10090 D(I)=PEEK(A+I)
10100 H=INT(D(I)/16):L=D(I)-16*H
10110 READHH,H$:IFHH<>HTHEN 10110
10120 RESTORE
10130 READLL,L$:IFLL<>LTHEN 10130
10140 RESTORE
10150 PRINT H$;L$:" ";
10160 NEXTI:PRINT" ";
10170 FORI=0TO7
10180 IFD(I)<128ANDD(I)>31THEN 10200
10190 PRINT"";:GOTO 10210
10200 PRINTCHR$(D(I));
10210 NEXTI
10220 NEXTA
10230 GOTO 10050
10240 INPUT"ADRESSE DEBUT";AD:I=0
10250 A=AD+I:GOSUB 10280
10260 INPUTD:IFD<0ORD>255THEN 10000
10270 POKEA,D:I=I+1:GOTO 10250
10280 FD=INT(A/256):FA=A-256*FD
10290 HD=INT(FD/16):LD=FD-16*HD
10300 HA=INT(FA/16):LA=FA-16*HA
10310 READHD,HD$:IFHD<>H0THEN 10310
10320 RESTORE
10330 READLO,LO$:IFLO<>L0THEN 10330
10340 RESTORE
10350 READH1,HA$:IFH1<>H1THEN 10350
10360 RESTORE
10370 READL1,LA$:IFL1<>L1THEN 10370
10380 RESTORE
10390 PRINT HD$;LD$;HA$;LA$:" ";
10400 RETURN
10410 DATA 0,"0",1,"1",2,"2",3,"3",4,"4",5,"5",6,
      "6",7,"7",8,"8",9,"9",10,"A",11,"B",12,"C",
      13,"D",14,"E",15,"F"

```

Sauvegarder l'ensemble phonèmes + routine L.M. par l'instruction:

CSAVEM"PSpch6", "&7000", "&600", à la suite du programme principal en BASIC qui suit.

```
10 INIT4,4:CURSORX 13:CURSOR Y 9
20 TX7,1,0
30 PRINT"PHONETIC 5000":CURSORX 13:PRINT"PHONETIC 5000"
:PRINT
40 TX6,0,0
50 CURSORX 11:PRINT"(C) H. BENDIT 1985":PRINT
60 TX0,0,1
70 CURSORX 11:PRINT"VEUILLEZ PATIENTER"
80 CLOAD"PSpch6"
90 INIT6,2
100 CURSORX 12:CURSOR Y 1:PRINT"PHONETIC-5000"
110 CURSORX 12:CURSOR Y 2:PRINT"*****":PRINT
120 PRINT"a,e,è,e,i,o,ô,u.E(eu),w(ou),A(an),I(in)"
130 PRINT"O(on),W(oi),b,d,f,g,j,k,l,m,n,p,R,r,s,t"
140 PRINT"v,z,(ch),N(gn),@(ail),&(eil),%(euil) "
150 PRINT"VITESSE : _ DUREE + - INTONATION < = >":PRINT
160 PRINT"SILENCES & PONCT.FINALE *(chuchot)":PRINT
170 PRINT"Composez";
180 CURSORX 10:PRINT "votre phrase à l'aide des sym-boles
phonétiques ci-dessus."
190 PRINT "
:PRINT
200 POKE18430,0:CLEAR80,28671
210 INPUTL$:L=LEN(L$)
220 CURSOR Y 14:PRINT " Veuillez patienter s'il vous plait"
"
230 V$=LEFT$(L$,1):W$=MID$(L$,2,1)
240 C=4:CO=0:FORI=1TOL
250 A$=MID$(L$,I,1)
260 IFA$=" "THENA$="."
270 IFA$="="THENCO=C:GOTO 440
280 IFA$="<"THEN 690
290 IFA$=">"THEN 690
300 IFA$="+"THEN 760
310 IFA$="-"THEN 840
320 READB$,M
330 IFB$<>A$THEN 320
340 D=112:DB=28672
350 A=D+PEEK(DB+2*M)
360 B=PEEK(DB+2*M+1)
370 K=PEEK(256*A+B+1)
380 FORJ=4TOK
390 N=PEEK(256*A+B+J)
400 POKE30204+C+J,N
410 NEXTJ
420 C=C+K-4
430 RESTORE
440 NEXTI
450 U=C/256:V=INT(U)
460 POKE30208,V
470 POKE30209,C-256*V
480 IFA$="*"THENGOSUB 650
```

```

490 GOSUB 950
500 POKE30211,60
510 CALL30136
520 CURSOR 14:POKE18430,1
530 PRINT"      APPUYER C POUR CORRIGER      "
540 PRINT"      S POUR SAUVEGARDER"
550 PRINT"      R POUR RECOMMENCER"
560 PRINT"      E POUR ECOUTER"
570 INPUTE$:IFE$="E"THENGOTO 1010
580 IFE$="C"THENCURSOR 14:PRINT"Corrigez ";GOTO 180
590 IFE$="S"THEN 900
600 IFE$="R"THENRESTORE:GOTO 90
610 INIT6,6:END
620 DATAa,0,e,1,i,2,o,3,u,4,e,5,e.e,E,7,w,8,A,9,1,10,
0,11,W,12,o,36
630 DATAb,13,d,14,f,15,g,16,j,17,k,18,l,19,m,20,n,21,
p,22,R,23,r,24
640 DATAc,25,t,26,v,27,z,28,$,29,N,30,@,31,&,32,%,33,
.,38,,39,+,39;!,38,_,38
650 FORJ=30215TO30207+CSTEP4
660 E=PEEK(J):G=INT(E/16)
670 IFG/2=INT(G/2)THENG=G+1:E=16*G:POKEJ,E
680 NEXTJ:RETURN
690 FORJ=30211+C0TO30211+CSTEP8
700 E=PEEK(J):G=INT(E/16)
710 IFG/2=INT(G/2)THEN 750
720 IFA$="<"THENG=2*INT(G/2):E=16*G+1
730 IFA$=">"THENG=2*INT(G/2)+1:E=16*G+15
740 POKEJ,E
750 NEXTJ:C0=C:GOTO 440
760 FORJ=0TO3
770 N=PEEK(30208+C-J)
780 POKE30212+C-J,N
790 N=PEEK(30204+C-J)
800 POKE30208+C-J,N
810 NEXTJ
820 C=C+4
830 GOTO 440
840 FORJ=0TO3
850 N=PEEK(30208+C-J)
860 POKE30204+C-J,N
870 NEXTJ
880 C=C-4
890 GOTO 440
900 INIT3,1:PRINTL$
910 INPUT"NOM DU FICHIER";N$
920 INPUT"MAGNETO PRET (O/N)";K$
925 IFK$="N"THEN 920
930 IFK$="O"THENCSAVEM N$,30136,C+72
940 GOTO 90
950 R=8:IFV$=W$THENR=4
960 FORJ=30215TO30211+CSTEPR
970 E=PEEK(J)
980 IFV$="_"THENE=E+32
990 IFV$="!"THENE=E-32
995 IFE<0THENE=0
1000 POKEJ,E:NEXTJ:RETURN
1010 FORX=1TO100:NEXTX:GOTO 510

```

Après avoir chargé le programme BASIC, son exécution commence par le chargement de l'ensemble phonèmes + routine L.M. (PSPCH6), puis l'ordinateur attend l'entrée de l'expression.

Les possibilités de modification de la prosodie sont les suivantes :

— Variation de la vitesse d'élocution si l'on commence l'expression par un ou deux signes | (plus vite) ou — (moins vite); ceci a pour effet de diviser ou multiplier la durée (égale à 16 ms avec les phonèmes 4.2) d'une trame sur deux ou de toutes les trames de l'expression. La vitesse est ainsi variable en 5 paliers dans un rapport de 1 à 4.

— Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou —.

— Création d'une courbe d'intonation par l'insertion de marqueurs qui déterminent les points d'inflexion et la pente de la courbe d'intonation. Le signe < détermine une pente montante, le signe > une pente descendante et le signe = une pente nulle; chaque marqueur agit entre le précédent (ou le début de l'expression s'il est le premier) et lui-même.

— Chuchotement par le symbole \* en fin d'expression.

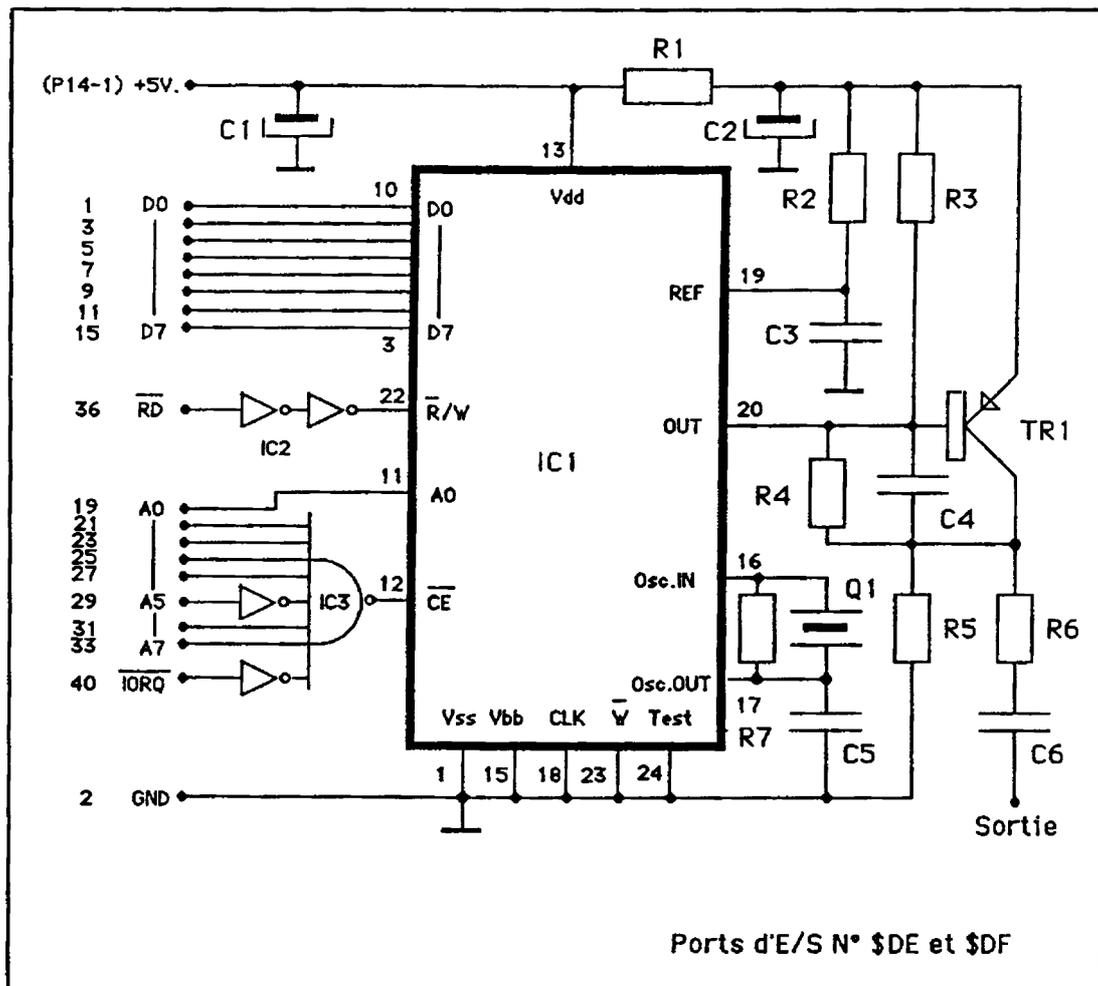
L'expression à créer se développe à partir de l'adresse "&7600"; sur cette machine dont la taille mémoire est restreinte, l'expression créée ne doit pas être trop longue pour ne pas empiéter sur la zone des variables. Une fois l'expression créée (ce qui dure quelques instants), le synthétiseur la prononce (ligne 510 CALL 30136).

Il vous est alors possible de corriger (C), de réécouter (E), de recommencer une autre expression (R), de sauvegarder votre expression (S) (dans ce dernier cas, le programme vous demande de lui donner un nom et enregistre sur cassette la routine "SPEECH" et l'expression à la suite l'une de l'autre); l'appui de toute autre touche arrête l'exécution du programme. Pour réutiliser l'expression sauvegardée dans un programme BASIC, il suffit de la charger et d'incorporer la ligne 510 dans ce programme.

## **SHARP MZ-700**

Cette machine a la particularité d'avoir un BASIC non résidant, dont plusieurs versions existent. Les programmes ci-après ont été écrits avec le S-BASIC, mais les autres versions (K-BASIC, DISK-BASIC, Q.D. BASIC) ne doivent pas poser de problème. Quatre programmes, facilement transposables, vous sont proposés.

Le synthétiseur est connecté sur le bus d'extension situé à l'arrière au moyen d'un connecteur femelle à 50 contacts. Il occupe les deux ports d'entrée-sortie \$DE (données) et \$DF (commandes). L'alimentation + 5V n'étant pas disponible sur le bus, on pourra la prendre sur l'un des connecteurs pour manettes de jeu. La sortie audio devra être envoyée par un fil blindé sur un amplificateur externe ou sur l'entrée son de la prise Péri-télévision en modifiant le câble d'origine qui ne transmet pas le son. Le schéma du synthétiseur est représenté ci-dessous et la valeur des composants est donnée par la nomenclature page 118 ; la notice du MZ-700 donne le brochage des connecteurs utilisés.



### 1°) PHONETIC-700

Le dictionnaire de phonèmes est à charger à partir de l'adresse \$A000. La version 4.2, dont le listing se trouve en annexe, est à utiliser pour bénéficier de toutes les fonctionnalités du programme et occupe jusqu'à l'adresse \$A5B7 incluse.

Le programme en langage-machine (SPEECH) listé ci-dessous, est celui du chapitre "Logiciel de commande" p. 52 ; l'adresse de l'expression à prononcer (ici \$B000) est placée en \$A5FE (poids faible) et \$A5FF (poids fort).

```
$A5B8 00 2A FE A5 46 23 4E CDEB A5 CDF2 A5 0B 0B 0B
$A5C8 23 23 CDF7 A5 CDEB A5 CDF7 A5 CDF7 A5 CDF7
$A5D8 A5 CDF7 A5 AFB9 20 EDB8 20 EACDEB A5 CDF2
$A5E8 A5 00 C9 DBDF CB 7F 28 FA C9 3E 1A D3 DF C9 7E
$A5F8 D3 DE23 0B C9 00 00 B0
```

Les codes des phonèmes et de la routine L.M. pourront être entrés en mémoire au moyen du moniteur intégré. Toujours en mode moniteur, sauvegarder à la suite du programme BASIC (listing ci-dessous) l'ensemble phonèmes + routine L.M. par la commande:

**\*S A000 A5FF 0000:PHONESPEECH-2**

```
10 LIMIT=9FFF
20 COLOR,,5,0:CLS:CURSOR0,0
30 PRINT "*****"
40 PRINT " * * "
50 PRINT " * PHONETIC-700 * "
60 PRINT " * * "
70 PRINT " * (C)H.BENOIT 1984 * "
80 PRINT " * * "
90 PRINT "*****"
100 COLOR,,2,0:PRINT:PRINT " CHARGEMENT PHONEMES EN COURS"
110 LOAD"PHONESPEECH-2"
120 D=$A0
130 DE=$A000
140 COLOR,,4,0:CLS
150 CURSOR12,1:PRINTL3,0"PHONETIC-700"
160 CURSOR12,2:PRINT"-----"
170 PRINT " A,e,I,(oh),O,U,(eh),,(a1),w(o1)"
180 PRINT
190 PRINT " a(an),i(in),o(on),<(ien),>(oin)"
200 PRINT
210 PRINT " t(en),W(ou),@(ail),&(e11),#(eu1)"
220 PRINT
230 PRINT " B,D,I,G,J,K(c/e),L,M,N,P,k,r,S,s"
240 PRINT
250 PRINT " 1,0,2,4(ch),.(en) SILENCES: ; ."
260 PRINT
270 PRINT " DUREE: + - INTONATION: / = \"
280 PRINT
290 PRINT " VITESSE: 1 8 PONCTUATION: . * "
300 PRINT " "
310 PRINT " "
320 PRINT " "
330 CURSOR2,19:PRINTL2,0J"COMPOSEZ votre phrase en utilisant"
```

```

340 PRINT[2,0J]"les symboles phonetiques ci-dessus"
350 CURSOR 0,22
360 INPUTL$:L=LEN(L$):REM*ENTREE EXPR*
370 V$=LEFT$(L$,1):W$=MID$(L$,2,1)
380 CURSOR0,19:PRINT"
390 PRINT[2,0J]"          PATIENTEZ S.V.P.
400 C=4:C0=C
410 FORI=1TOL
420 A$=MID$(L$,I,1)
430 IFA$=" "THENA$=";"
440 IFA$=="="THENC0=C:GOTO610
450 IFA$="\ "THEN930
460 IFA$="/ "THEN930
470 IFA$="+ "THCN950
480 IFA$="- "THEN1030
490 READB$,M
500 ONERRORGOTO1130
510 IFB$<>A$THEN490
520 A=D+PEEK(DB+2*M):REM*CREATION EXPR*
530 B=PEEK(DB+2*M+1)
540 K=PEEK(256*A+B+1)
550 FORJ=4TOK
560 N=PEEK(256*A+B+J)
570 POKE45052+C+J,N
580 NEXTJ
590 C=C+K-4
600 RLSTORE
610 NEXTI
620 U=C/256:V=INT(U):REM**EN TETE EXPR**
630 POKE%B000,V
640 POKE%B001,C-256*V
650 POKE%B003,60
660 GOSUB1220
670 IFA$="*"THINGOSUB830
680 CURSOR8,20:PRINT"
690USR($A5B8):REM**VOCALISATION**
700 CURSOR0,17
710 PRINT[2,0J]"          ESPACE POUR REPETER
720 PRINT[2,0J]"          'S' POUR SAUVEGARDER
730 PRINT[2,0J]"          'C' POUR CORRIGER
740 PRINT[2,0J]"          AUTRES POUR RECOMMENCER
750 GETE$:IFE$=" "THEN750
760 IFE$=" "THEN690
770 IFE$="C"THENCURSOR0,19:PRINT[2,0J]"RE":CURSOR0,17:
GOTO310
780 IFE$="S"THEN1090
790 PRINT:RESTORE:GOTO130
800 DATA0,e,1,I,2,0,3,U,4,,5,',6,E,7,W,8,a,9,1,10,
0,11,u,12,B,13,D,14,F,15,C,
16,J,17,0,36,s,37
810 DATAk,18,L,19,M,20,N,21,P,22,R,23,r,24,S,25,T,26,
V,27,Z,28,$,29,,30,@,31,8,
32,4,33,<,34,>,35
820 DATA;,38,,39,*;39,↑,38,↓,38
830 FORJ=45063TO45056+C-1STEP4:REM**CHUCHOTEMENT**
840 E=PEEK(J):G=INT(E/16)
850 IFG/2=INT(G/2)THENG=G+1

```

```

860 E=16*G:POKEJ,E:NEXTJ
870 RETURN
880 E=PEEK(J):G=INT(E/16):REM**MOD PI**
890 IFG/2>INT(G/2)THENRETURN
900 IFA$="/" THENG=2*INT(G/2):E=16*G+1
910 IFA$="\ " THENG=2*INT(G/2)+1:E=16*G+15
920 POKEJ,E:RETURN
930 FORJ=45056+C0+3TO45056+C+3STEP8:REM**INTONATION**
940 GOSUB800:NEXTJ:C0=C:GOTO610
950 FORJ=0TO3:REM**DUREE + **
960 N=PEEK($B000+C-J)
970 POKE($B000+C+4-J),N
980 N=PEEK($B000+C-J-4)
990 POKE($B000+C-J),N
1000 NEXTJ
1010 C=C+4
1020 GOTO610
1030 FORJ=0TO3:REM**DUREE - **
1040 N=PEEK($B000+C-J)
1050 POKE($B000+C-4-J),N
1060 NEXTJ
1070 C=C-4
1080 GOTO610
1090 CLS:PRINTL$:PRINT:PRINTC2,0J"NOM DU FICHIER? ":INUTN#
1100 PRINT"*S B000 ";HEX$(45056+C);" 0000 ";N#
1110 PRINTC2,0J"RECOPIER LA LIGNE CI-DESSUS PUIS.CR'"
1120 BYE
1130 IFERN=24TIEN1150
1140 END
1150 CURSOR0,19:PRINTC3,0J"VOUS AVEZ UTILISE UN CARACTERE
INCONNU"
1160 PRINT"
1170 H=0:IFI>38THENI=1-40:H=1
1180 COLOR,3,0
1190 CURSOR0,22+H:PRINTTAB(I+J);"? "
1200 COLOR,4,0
1210 RESTORE:GOTO350
1220 R=8:IFW#V$THENR=4:REM**UTILISSE**
1230 FORJ=45063TO45055+OSTEPF
1240 E=PEEK(J)
1250 II V$="↓" THENE=E+32
1260 IF V$="↑" THENE=E-32
1270 POKEJ,E:NEXTJ
1280 RETURN

```

Après avoir chargé le programme BASIC, son exécution commence par le chargement de l'ensemble phonèmes + routine L.M. (PHONESPEECH-2), puis l'ordinateur attend l'entrée de l'expression. Les options prosodiques de ce programme sont les suivantes :

— Variation de la vitesse d'élocution si l'on commence l'expression par un ou deux signes ↑ (plus vite) ou ↓ (moins vite); ceci a pour effet de diviser ou multiplier la durée (égale à 16 ms avec les phonèmes 4.2) d'une trame sur deux ou de toutes les trames de l'expression. La vitesse est ainsi variable en 5 paliers dans un rapport de 1 à 4.

- Augmentation ou réduction de la durée individuelle de chaque phonème en faisant suivre son symbole d'un ou plusieurs signes + ou -.
- Création d'une courbe d'intonation par l'insertion de marqueurs qui déterminent les points d'inflexion et la pente de la courbe d'intonation. Le signe / détermine une pente montante, le signe \ une pente descendante et le signe = une pente nulle ; chaque marqueur agit entre le précédent (ou le début de l'expression s'il est le premier) et lui-même.
- Chuchotement par le symbole \* en fin d'expression.

L'expression à créer se développe à partir de l'adresse \$B000. Une fois l'expression créée (ce qui dure quelques instants), le synthétiseur la prononce (ligne 710 USR(\$A5B8)). Il vous est alors possible de corriger (C), de réécouter (ESPACE), de sauvegarder votre expression (S) (dans ce dernier cas, le programme vous demande de lui donner un nom, et enregistre l'expression seule); l'appui de toute autre touche recommence l'exécution du programme que l'on peut quitter par "SHIFT + BREAK".

Pour réutiliser l'expression dans un autre programme, il faudra tout d'abord charger la routine L.M. "SPEECH" (\$A5B8 à \$A5FF), puis appeler l'expression par l'instruction USR (\$A5B8). Il serait aisé de modifier le programme pour que l'expression débute à l'adresse \$A600 et que la sauvegarde inclue la routine L.M. pour faciliter sa réutilisation.

## 2°) MZ-CHANTEUR

Ce programme a pour but la création d'une expression chantée en utilisant les principes expliqués page 65. Il est dérivé du programme "PHONETIC-700" avec lequel il possède des parties communes importantes. Il ne fonctionne qu'avec les phonèmes 4.2 (trame constante de 16 ms). Le chant est obtenu en déterminant pour chaque syllabe un pitch de fréquence égale à celle de la note recherchée, et une durée proche de celle indiquée par la partition.

```

10 LIMIT=9FFF
20 COLOR,,5,0:CLS:CURSOR0,E
30 PRINT "          *****"
40 PRINT "          *                *"
50 PRINT "          *   MZ   CHANTEUR   *"
60 PRINT "          *                *"
70 PRINT "          * (C)H.BENOIT 1985  *"
80 PRINT "          *                *"
90 PRINT "          *****"
100 COLOR,,2,0:PRINT:PRINT "  CHARGEMENT PHONEMES EN COURS "
110 LOAD "PHONESPEECH-2"
120 D=$A0:DE=$A000
130 COLOR,,4,0:CLS
140 PRINT "  A,e,I,O(oh),U,(eh), (ai),W(oi)"
150 PRINT
160 PRINT "  a(au),i(in),o(on),<(len),>(oin)"

```

```

170 PRINT
180 PRINT " E(eu),W(ou),@(ail),R(eil),I(euil)"
190 PRINT
200 PRINT " B,D,F,G,J,K(c/a),L,M,N,P,R,r,S,s"
210 PRINT
220 PRINT " T,U,Z,(ch), (gn) SILENCES: ; ."
230 PRINT
240 PRINT "0=fa 1=sol 2=la 3=si 4=Do 5=Re 6=Mi 7=Fa"
250 PRINT "8=Sol 9=La A=S; B=DO C=RE D=MI E=FA F=SO"
260 PRINT " Croche=0 Noire=1 Blanche=2 Ronde=3"
270 PRINT
280 PRINT "          Premiere ligne=TEXTE          ."
290 PRINT "          Deuxieme ligne=NOTES           ."
300 PRINT "          Troisieme ligne=DURCE           ."
310 CURSOR2,19:PRINT(2,0)"COMPOSEZ AVEC LES SIGNES CI
DESSUS":CURSOR0,21
320 REM****ENTREE CHANSON****
330 INPUTL$:INPUTM$:INPUTF$:L=LEN(L$)
340 CURSOR0,19:PRINT(2,0)"          PATIENTEZ S.V.P."
350 C=4:C0=C
370 FORI=1TOL
380 A$=MID$(L$,I,1):N$=MID$(M$,I,1):FD$=MID$(F$,I,1)
:FD=VAL(FD$):F0=F
390 IFA$=" "THLNA$=";"
400 READB$,M
410 ONERRORGOTO1120
420 IFB$<>A$THEN400
430 IFN$="0"THENCN=88
440 IFN$="1"THENCN=98
450 IFN$="2"THENCN=110
460 IFN$="3"THENCN=124
470 IFN$="4"THENCN=130
480 IFN$="5"THENCN=146
490 IFN$="6"THENCN=164
500 IFN$="7"THENCN=174
510 IFN$="8"THENCN=196
520 IFN$="9"THENCN=220
530 IFN$="A"THENCN=246
540 IFN$="L"THENCN=262
550 IFN$="C"THENCN=294
560 IFN$="D"THENCN=330
570 IFN$="L"THENCN=350
580 IFN$="F"THENCN=392
590 REM****CREATION EXPRESSION****
600 IFI=1THENM0=C$.D0S;F0=F
610 A=D+PEEK(DB+I*M)
620 B=PEEK(DB+2*M+1)
630 R=PEEK(256*A+B+1)
640 IGRJ=4TOR
650 N=PEEK(256*A+B+J)
660 POKE45052+C+J,N
670 NEXTJ
680 REM****CALCUL DU PITCH****
690 DF=F-F0:P1=DF/2:P2=0:P3=0
700 IFF1=0THEN800
710 IFF1>15THENP1=15:P2=(DF/2)-15:IFF2>15THENP2=15:P3=
(DF/2)-30:IFP3<17THENP3=17
720 IFF1<0THENP1=32+P1:IFF1<17THENP1=17:P2=(DF/2)+47:

```

```

IFP2<17THENP2=17:P3=(D/2)+ 62:IFP3<17THENP3=17
730 Z=PEEK(45059+C):G=INT(Z/16):IFG/2>INT(G/2)THENF=
F0:GOTO800
740 Z=Z+P1:POKE(45059+C),Z
750 Z=PEEK(45063+C):G=INT(Z/16):IFG/2>INT(G/2)THENF=
F0:GOTO800
760 Z=Z+P2:POKE(45063+C),Z
770 Z=PEEK(45067+C):G=INT(Z/16):IFG/2>INT(G/2)THENF=
F0:GOTO800
780 Z=Z+P3:POKE(45067+C),Z
790 REM***MODIFICATION PITCH***
800 FORJ=15TOKSTEP4
810 Z=PEEK(45052+C+J):G=INT(Z/16):E=Z-16*G:H=INT(G/8):
L=G-2*INT(G/2)
820 G=8*H+2*PD+L:IFE<>0THENC=0*H+2+L
830 Z=E+16*G:POKE(45052+C+J),Z
840 NEXTJ
850 C=C+K-4
860 RESTORE
870 NEXTI
880 REM***LONGUEUR EXPRESSION***
890 U=C/256:V=INT(U)
900 POKE$B000,V
910 POKE$B001,C-256*V
920 CURSOR8,19:PRINT"
930 USR($A5B8):REM***VOCALISATION***
940 CURSOR0,16
950 PRINT[2,0]"          ESPACE POUR REPLYER          "
960 PRINT[2,0]"          'S' POUR SAUVEGARDER          "
970 PRINT[2,0]"          'C' POUR CORRIGER          "
980 PRINT[2,0]"
990 GETE$:IFE$=" "THEN990
1000 IFE$=" " THEN930
1010 IFE$="C"THENCURSOR0,19:PRINT[2,0]"RE":CURSOR0,16:
GOTO280
1020 IFE$="S"THENJ080
1030 PRINT:RESTORE:GOTO130
1040 DATAA,0,e,1,I,2,O,3,U,4, ,5, ',6,L,/,W,8,.,9,i,10,
o,11,w,12,B,13,D,14,F,15,G,16,J,17,0,36,s,37
1050 DATAK,18,L,19,M,20,N,21,F,22,R,23,r,24,S,25,l,26,
V,27,Z,28,$,29, ,30,@,31,K ,32,#,33,*,34,>,35
1060 DATA;,38,.,,39
1070 REM***SAUVEGARDE EXPRESSION***
1080 CLS:PRINTL$:PRINT:PRINT[2,0]"NOM DU FICHIER? ":INPUTM
1090 PRINT"*S B000 " ;HEX$(45056+C):" 0000:" ;M
1100 PRINT[2,0]"RECOPIER LA LIGNE CI-DESSUS PUIS 'CR'"
1110 BYE
1120 IFFERN=24THEN1150
1130 END
1140 REM***MESSAGE D'ERREUR***
1150 CURSOR0,19:PRINT[3,0]"VOUS AVEZ UTILISE UN CARACTERE
INCONNU"
1160 PRINT"
1170 COLOR,,3,0
1180 CURSOR0,21:PRINTTAB(1+1);"?"
1190 COLOR,,4,0
1200 RESTORL:GOTO320

```

Afin de simplifier le programme, ceci a été réalisé en créant 3 chaînes de caractères entrées sur 3 lignes d'écran successives, chacune ayant un caractère par phonème, pour représenter l'expression :

- L\$ représente la suite de phonèmes composant l'expression (TEXTE)
- M\$ représente la note attribuée à chaque phonème (NOTE)
- F\$ représente la durée relative de chaque phonème (DUREE)

Le programme ignore dièses et bémols, mais rien n'interdirait de les rajouter à la liste des notes ; celle-ci en comporte 16 (de FA<sub>2</sub> = 88 Hz à SOL<sub>4</sub> = 392 Hz), soit un peu plus de 2 octaves. Le pitch initial est rendu égal à la fréquence de la première note (ligne 600).

En raison du codage en valeur relative du pitch sur le MEA 8000, chaque note doit être calculée par rapport à la précédente ; c'est ce que font les lignes 690 à 780 du programme, en atteignant cette note le plus rapidement possible (de 0 à 3 trames selon l'écart entre 2 notes successives ; cet écart est donc limité à 90 Hz dans cet intervalle de temps). En cas de phonème non-voisé, le calcul de la note est reporté au phonème suivant.

La durée est variable dans un rapport 0,5 à 4 par rapport à la durée nominale du phonème, soit un rapport 8 entre extrêmes. Ceci est obtenu en modifiant la durée de toutes les trames du phonème, sauf celle des 3 premières qui sont utilisées pour modifier la note (lignes 800 à 870). Les possibilités de répétition, correction, sauvegarde sont les mêmes que pour le programme "PHONETIC-700". En pratique, pour utiliser ce programme, il faut :

- a) entrer le texte sous forme phonétique.
- b) faire correspondre une note à chaque syllabe, généralement composée d'une ou plusieurs consonnes et voyelles. La même note devra être attribuée à chacun des caractères (phonèmes) composant la syllabe.
- c) faire correspondre une durée à chaque note ; la ou les voyelles de la syllabe constituant la partie caractéristique de la note, c'est donc sur elles que la modification de durée doit essentiellement porter. S'il y a plusieurs voyelles consécutives dans la syllabe, répartir la durée entre elles.

Exemple d'utilisation pour le refrain "Au clair de la lune" :

Texte	Au	clair	de	la	lu	-	ne
Notes	do	do	do	ré	mi	ré	
Durées	Nr	Nr	Nr	Nr	Bl	Bl	

(Nr = noire, Bl = blanche)

Il faudra entrer les chaînes de caractères :

Texte	O	K	L	`	R	D	e	L	A	L	U	N	e
Notes	4	4	4	4	4	4	4	5	5	6	6	5	5
Durées	1	1	1	1	1	1	1	1	1	1	2	1	2

L'expression chantée ainsi créée occupe exactement le même espace mémoire que l'expression phonétique brute ; on peut la sauvegarder et la réutiliser de la même manière qu'avec "PHONETIC-700".

### 3°) HORLOGE PARLANTE

Ce programme permet à votre ordinateur, qui comporte une horloge intégrée, de vous donner l'heure en français sous forme parlée. Afin d'obtenir une qualité supérieure à celle des phonèmes, nous avons utilisé ici un vocabulaire spécialisé baptisé "VOC'HORLOGE" dont le listing est fourni en annexe. Celui-ci permet d'énoncer par composition tous les chiffres compris entre 0 et 59, ainsi que les mots HEURE, MINUTE, SECONDE et BONJOUR ; il occupe environ 2 kilo-octets de mémoire.

Afin de permettre des liaisons satisfaisantes, les chiffres 6 et 10 ont été amputés de leur terminaison "s", qui a été codée séparément ainsi que la liaison "z" ; en effet il ne faut pas oublier que :

- 6 heures se prononce SIZEURE alors que 6 tout seul se prononce SIS et que 6 minutes se prononce SIMINUTE.
- Il en est de même pour 10.
- Les chiffres 2 et 3 exigent également la liaison "z" lorsqu'ils sont placés devant le mot "heures".

D'autre part, les chiffres des dizaines 20 à 50 ont été codés sans leur terminaison "gt" ou "te", les terminaisons/liaisons "t", "te", "tet" ayant été codées séparément pour pouvoir tenir compte des particularités de notre belle langue :

- 21 heures (qui s'écrit aussi VINGT ET UNE HEURES) sera représenté par les éléments VIN + TET + UNE + HEURE,
- 22 heures sera représenté par VIN + T + DEUX + Z + HEURE.

Le programme en BASIC "HORLOGE PARLANTE", dont le listing se trouve ci-dessous, tient compte de toutes ces particularités. Après avoir entré puis sauvegardé ce programme, entrer, sous "moniteur", le vocabulaire "VOC'HORLOGE" à partir de l'adresse \$A000 et la routine L.M. ci-dessous (construite sur le modèle p. 52 ) à partir de l'adresse \$A800.

```
$A800 00 2A 4E A8 46 23 4E CD33 A8 CD3A A8 0B 0B 0B
$A810 23 23 CD3F A8 CD33 A8 CD3F A8 CD3F A8 CD3F
$A820 A8 CD3F A8 AFB9 20 EDB8 20 EA CD33 A8 CD3A
$A830 A8 00 C9 DBDF CB 7F 28 FA C9 3E 1A D3 DF C9 0B
$A840 7E D3 DE23 C9 00 00 00 00 00 00 00 00 00 00
```

```

10 LIMIT$9FF1
20 COLOR,2,0:CLS:CURSOR0,0
30 PRINT " ***** "
40 PRINT " * * "
50 PRINT " * HORLOGE PARLANTE * "
60 PRINT " * * "
70 PRINT " * (C)H.BENOIT 1984 * "
80 PRINT " * * "
90 PRINT " ***** "
100 COLOR,2,0:PRINT:PRINT"APPUYER 'PLA)' POUR CHARGER
VOCABULAIRE"
110 LOAD"HORLOGESPEECH"
120 CLS:CURSOR0,2
130 PRINT " HORLOGE PARLANTE "
140 PRINT " ***** "
150 D=$A0
160 DE=$A000
170 CURSOR0,5
180 INPUT " ENTREE L'HEURE (HHMMSS) " ;I$
190 TI$=I$
200 CURSOR0,5:PRINT" APPUYER 0 POUR REMETTRE A L'HEURE "
210 PRINT " "
220 H$=LEFT$(I$,2)
230 M$=MID$(I$,3,2)
240 S$=RIGHT$(I$,2)
250 CURSOR15,11
260 PRINTH$;" ":"M$;" ":"S$
270 IFS$="02"THENH$="06"+" "+"27"
280 GETK$:IFK$=" "THENH$="10"+" "+"27"
290 IFK$="0"THENH$="17"+" "+"27"
300 GOTO220
310 IFH$="06"THENH$="06"+" "+"27"
320 IFH$="10"THENH$="10"+" "+"27"
330 IFH$="17"THENH$="10"+" "+"27"+" "+"07"
340 IFH$="18"THENH$="10"+" "+"27"+" "+"08"
350 IFH$="19"THENH$="10"+" "+"27"+" "+"09"
360 IFH$="20"THENH$="17"+" "+"24"
370 IFH$="21"THENH$="17"+" "+"26"+" "+"01"
380 IFH$="22"THENH$="17"+" "+"25"+" "+"02"+" "+"27"
390 IFH$="23"THENH$="17"+" "+"25"+" "+"03"+" "+"27"
400 IFM$="17"THENM$="10"+" "+"27"+" "+"07"
410 IFM$="18"THENM$="10"+" "+"28"+" "+"08"
420 IFM$="19"THENM$="10"+" "+"28"+" "+"09"
430 IFM$="20"THENM$="17"+" "+"24":GOTO460
440 IFM$="21"THENM$="17"+" "+"26"+" ":"01":GOTO450
450 IFLEFT$(M$,1)="2"THENM$="17"+" "+"25"+" "+"0"+RIGHT$(
(M$,1)
460 IFM$="30"THENM$="18"+" "+"25":GOTO490
470 IFM$="31"THENM$="18"+" "+"26"+" "+"01":GOTO490
480 IFLEFT$(M$,1)="3"THENM$="18"+" "+"25"+" "+"0"+RIGHT$(
(M$,1)
490 IFM$="40"THENM$="19"+" "+"25":GOTO520
500 IFM$="41"THENM$="19"+" "+"26"+" "+"01":GOTO520
510 IFLEFT$(M$,1)="4"THENM$="19"+" "+"25"+" "+"0"+RIGHT$(
(M$,1)
520 IFM$="50"THENM$="20"+" "+"25":GOTO550

```

```

530 TI=M$="51" THENM$="20"+" "+"26"+" "+"01" :GOTO550
540 IF LEFT$(M$,1)="5" THENM$="20"+" "+"25"+" "+"0"+RIGHT$(M$,1)
550 L$=H$+" "+"21"+" "+"+M$+" "+"22"
560 L=LEN(L$)
570 FOR I=0 TO (L-2)/3
580 M=VAL(MID$(L$,1+3*I,2))
590 A=D+PEEK(DB+2*M)
600 B=PEEK(DB+2*M+1)
610 POKE $A84E,B:POKE $A84F,A
620 USR($A800)
630 NEXT I
640 RETURN

```

Sauvegarder l'ensemble à la suite du programme BASIC par :

**\*S A000 A844 0000:HORLOSPEECH**

Lors de l'exécution, le programme BASIC commence par charger l'ensemble "HORLOSPEECH". Il vous demande ensuite d'entrer l'heure réelle sous la forme HHMMSS ; cette chaîne T\$ sert à initialiser la variable TI\$ qui représente l'heure de l'horloge interne du MZ-700 ; celle-ci est ensuite actualisée automatiquement à chaque seconde. Le programme affiche en permanence l'heure à l'écran sous la forme :

**HH : MM . SS**

Il vous est possible d'écouter l'heure (heures et minutes seulement) à tout moment en appuyant sur la barre d'espace, et automatiquement à chaque changement de minutes (ceci peut être évité en supprimant la ligne 270). Chaque fois qu'il doit énoncer l'heure, le programme lit la chaîne TI\$ et en tire les chaînes H\$ (heures) et M\$ (minutes), qui sont alors transformées en une suite de numéros d'ordre des éléments de vocabulaire à prononcer (lignes 310 à 540), compte tenu des particularités telles que les liaisons. On forme ensuite la chaîne à prononcer L\$ (ligne 550) en ajoutant les codes des mots "heure" (21) et "minute"(22) à la suite de H\$ et M\$. On en tire un par un le numéro de chaque élément à prononcer (ligne 580), pour calculer son adresse (lignes 590 et 600) ; celle-ci est "pokée" (ligne 610) en \$A84E (poids faible) et \$A84F (poids fort) avant la vocalisation (ligne 620).

#### **4°) Programme L.M. de concaténation d'éléments de vocabulaire :**

Ce programme, dont l'adresse d'exécution est \$A600, est à placer à la suite de la routine SPEECH dont il utilise les sous-programmes STOP, TRAM et TREQ. Il permet la prononciation enchaînée de plusieurs éléments de vocabulaire en mémoire (phonèmes, bruits etc.), sans créer d'expression nouvelle à partir de ces éléments. Leur table d'adresse doit

débuter à \$A000, et le vocabulaire peut s'étendre jusqu'à \$A5B7, ce qui est le cas des phonèmes 4.2; la longueur de chaque élément doit être inférieure à 256 octets.

Ceci permet une prononciation immédiate de l'expression recherchée à partir du numéro d'ordre de chacun des éléments la composant, numéros préalablement chargés à la suite les uns des autres à partir de l'adresse \$A650; la fin de l'expression est provoquée par la lecture d'un numéro supérieur ou égal à \$28 (40 décimal), le programme ayant été initialement prévu pour les phonèmes (39 éléments); ceci peut être modifié en changeant l'octet \$A614.

Le chargement des numéros d'ordre peut s'effectuer par un programme BASIC, qui pourrait également faire une traduction des codes ASCII en numéros de phonèmes si on désirait une entrée phonétique.

Aucune modification d'intonation ou d'autres paramètres n'est possible, car il faudrait modifier les phonèmes eux-mêmes, ce qui interdirait deux utilisations successives dans un contexte différent. Le listing commenté de ce programme sera aisément adaptable à une autre machine équipée du Z-80 en changeant les adresses spécifiques au MZ-700. En pratique, pour les machines étudiées ici, (à l'exception de l'AMSTRAD qui demandera des modifications plus profondes), il suffira de modifier l'octet de poids fort des adresses concernées, puisque toutes les routines SPEECH ont été implantées à une adresse de la forme \$X5B8.

#### **Routine de concaténation d'éléments de vocabulaire de longueur < à 256 octets**

**Ce programme doit être précédé de la routine SPEECH dont il utilise les sous-programmes STOP, TRAM et TREQ.**

<b>Adr.</b>	<b>Op.code</b>	<b>Etiqu.</b>	<b>Mnémo.</b>	<b>Commentaire</b>
<b>Programme principal</b>				
A600	00	DEBU:	NOP	E5(PUSH HL)pour VG 5000
A601	21 00	A0	LD HL, A000H	Adresse début table ds HL
A604	11 50	A6	LD DE, A650H	Adr. début. express. ds DE
A607	CD F2	A5	CALL STOP	Init. MEA (Arrêt lent)
A60A	CD EB	A5	CALL TREQ	Test de REQ
A60D	3E 3C		LD A, 3CH	Hex 3C (60) dans A
A60F	D3 DE		OUT DEH, A	Envoi du Pitch initial
A611	1A	SUIV:	LD A, (DE)	Numéro élément dans A
A612	13		INC DE	DE pointe élément suivant
A613	FE 28		CP 28H	Compar. à Hex. 28 (40)

A615 FA1A A6	JP M, PARL	Si < 40, vocalisation
A618 00	NOP	E1 (POP HL) pour VG 5000
A619 C9	RET	Si > 40, retour (fin d'expr.)

### Sous-programme de vocalisation (PARL)

A61A 87	PARL: ADD A, A	A=p. faible adr. d'adr. élém.
A61B 6F	LD L, A	adresse. d'adr. élém. ds HL
A61C 7E	LD A, (HL)	P. fort adr. rel. élém. ds A
A61D C6 A0	ADD A, A0H	P. fort adr. abs. élém. ds A
A61F 32 4F A6	LD (A64FH), A	Sauvegarde p. fort en A65F
A622 23	INC HL	HL pointe p. faible adr. ele.
A623 7E	LD A, HL	P. faible adr. élém dans Acc
A624 32 4E A6	LD (A64EH), A	Sauveg. p. faible en A65E
A627 E5	PUSH HL	Préserve le registre HL
A628 2A 4E A6	LD HL,(A64EH)	HL pointe élément
A62B 23	INC HL	HL pointe p. faible longueur
A62C 4E	LD C, HL	Longueur élément dans C
A62D 0B	DEC BC	
A62E 0B	DEC BC	
A62F 0B	DEC BC	
A630 0B	DEC BC	BC = longueur restante
A631 23	INC HL	
A632 23	INC HL	
A633 23	INC HL	HL pointe 1er octet trame
A634 CD EB A5NEXT:	CALL TREQ	Test de REQ
A637 CD F7 A5	CALL TRAM	1e octet trame+i.HL+d.BC
A63A CD F7 A5	CALL TRAM	2e octet trame+i.HL+d.BC
A63D CD F7 A5	CALL TRAM	3e octet trame+i.HL+d.BC
A640 CD F7 A5	CALL TRAM	4e octet trame+i.HL+d.BC
A643 AF	XOR A	Acc = 0
A644 B9	CP C	P. faible longueur = 0?
A645 20 ED	JR, NZ NEXT	Si non, tr. suivante (NEXT)
A647 CD EB A5	CALL TREQ	Test de REQ
A64A E1	POP HL	Restitution de HL
A64B C3 11 A6	JP SUIV	Vers élément suivant

Le programme commence par initialiser le synthétiseur en mode "ARRET LENT", puis lui envoie un pitch initial de 120 Hz (cette valeur peut être modifiée par l'octet d'adresse \$A60E qui vaut PT/2). Le programme prononce ensuite chacun des éléments sans envoyer de commande "STOP" ni de pitch initial entre chacun d'eux pour éviter toute discontinuité (impératif pour concaténer des phonèmes). La détection d'un numéro supérieur à 39 termine l'exécution du programme et l'expression s'arrête automatiquement sans envoi de commande "STOP" en raison de l'utilisation du mode "ARRET LENT".

Page blanche

# Chapitre 6

---

## PHONETRAM

### Introduction

---

PHONETRAM est un logiciel de composition et d'édition graphique de codes vocaux utilisables par le circuit de synthèse vocale MEA 8000 de RTC. Il permet en mode interactif de composer, modifier, affiner et sauvegarder, en totalité ou partiellement, des mots ou expressions réutilisables sous ASSEMBLEUR ou sous le contrôle de l'ensemble des langages évolués implantés sur les machines THOMSON (BASIC, LOGO, FORTH, LSE, etc.).

Les codes vocaux générés par PHONETRAM sont, bien entendu, utilisables sur d'autres machines, sous réserve que le langage résidant sur cette machine admette la lecture des supports (disquettes ou cassettes) écrits sous le DOS BASIC utilisé par THOMSON. Dans le cas contraire, il vous faudra passer par une interface de communication RS 232 ou par la programmation de mémoires PROM ou EPROM pour récupérer les codes vocaux.

L'application PHONETRAM décrite ci-après fonctionne sous BASIC et utilise les lecteurs de cassettes ou disquettes pour sauvegarder les codes vocaux générés par le logiciel. Pour réutiliser ces codes vocaux dans une

application particulière, l'utilisateur dispose d'un logiciel intitulé **NOR-MAPHON**, tournant sous **BASIC**, qui assure la normalisation des codes vocaux produits par **PHONETRAM** et génère une table de données vocales normalisée, semblable à celle décrite page 38 .

**PHONETRAM** utilise le plus souvent le crayon optique pour sélectionner les commandes, matérialisées à l'écran par des zones que l'on distingue grâce à l'icône qui symbolise la fonction. La manipulation du crayon optique est grandement facilitée par le pointeur clignotant matérialisant la zone visée quand le crayon optique est à une distance suffisante. Les caractères représentant les phonèmes sont entrés à l'aide du clavier.

## **Principe**

---

**PHONETRAM** réalise deux tâches essentielles qui sont :

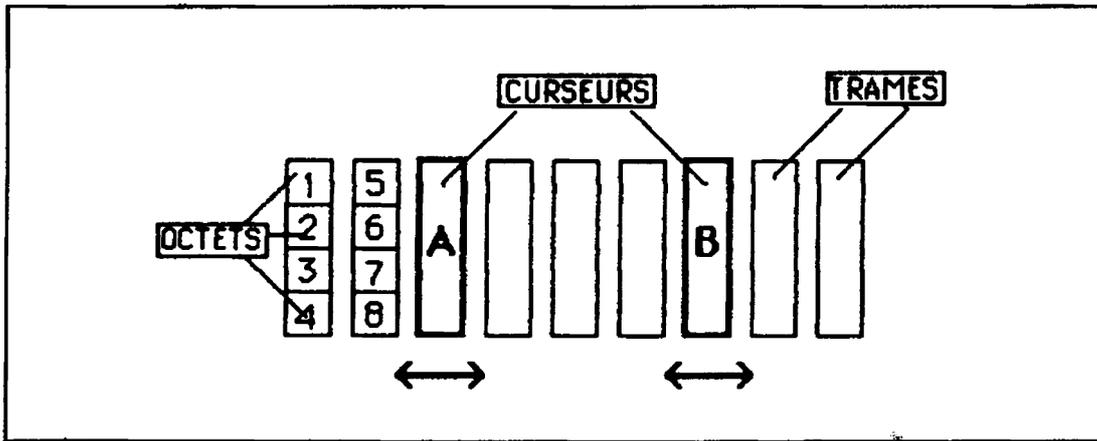
- 1. la concaténation des phonèmes pour composer un mot ou une expression,
- 2. l'affinage phonique du mot ou de l'expression créée, en permettant à l'utilisateur d'appliquer sur son expression "brute" les règles élémentaires de la phonétique (intonation, déclinaison, rythme, etc.).

### **Concaténation des phonèmes**

**PHONETRAM** utilise le tableau des phonèmes dits 3.3, répertoriés en annexe et dont le tableau d'attribution des caractères est donné page 91 . Les caractères, représentant les phonèmes constituant le mot ou l'expression souhaitée par l'utilisateur, sont entrés un à un dans l'ordre de leur prononciation à l'aide du clavier et constituent le mot "brut". Ces données sont stockées en mémoire dans une zone intitulée **MOTX**.

### **Affinage phonique**

Un mot est représenté en mémoire par une suite d'octets qui sont regroupés par paquets de quatre pour construire une "chaîne" de trames. **PHONETRAM** a la possibilité de visualiser deux trames sur ce mot. La visualisation de ces trames est appelée curseur. Le curseur A évolue de la trame un au curseur B et le curseur B évolue du curseur A à la fin du mot ou de l'expression contenu en mémoire.



Chaque curseur “ouvre” la trame recouverte et donne la possibilité à l'utilisateur de modifier l'ensemble des paramètres constituant cette trame. Une zone de mémoire dite “mémoire auxiliaire ou MOTY” permet de sauvegarder temporairement une partie ou la totalité de l'expression. Cette possibilité donne à l'utilisateur le loisir de comparer à chaque instant l'expression en cours d'affinage avec l'expression précédente pour lui permettre d'apprécier le résultat de son travail. Le contenu de la mémoire auxiliaire peut être également ajouté, à partir du curseur B, au mot contenu dans MOTX pour former une autre expression.

PHONETRAM autorise deux modes de représentation graphique de l'expression :

**ECRAN 1:** Cet écran est dit “numérique” parce qu'il représente chaque paramètre non seulement sous forme d'une barre d'élongation proportionnelle à la valeur du paramètre, mais aussi sous forme d'un affichage numérique.

**ECRAN 2:** L'écran 2 donne l'allure générale de la courbe hauteur/temps entre le curseur A et le curseur B. Par opposition au premier, cet écran est dit “analogique” ou “graphique”.

# Exploitation de PHONETRAM

---

Notez: dans tout ce qui suit, nous supposons que vous êtes possesseur de la cassette commercialisée par CEDIC/NATHAN.

— Vérifiez que l'ensemble des périphériques nécessaires est convenablement branché<sup>1</sup>. Avec un TO7, l'extension mémoire est impérative.

— Si vous utilisez un TO7 ou un TO7-70, vérifiez que la cartouche BASIC est bien présente dans la trappe. Sur MO5 et TO9, le BASIC est résidant.

— Mettez sous tension dans l'ordre habituel et préconisé par le constructeur. La page d'en-tête de votre système apparaît à l'écran. A ce niveau, prenez soin de régler votre crayon optique, c'est un élément important dans PHONETRAM.

— Insérez la cassette dans votre LEP, en prenant garde de bien positionner la face correspondant à votre ordinateur.

Face verte → MO5                      Face rouge → TO7, TO7-70, TO9

— Chargez PHONETRAM en sélectionnant l'option 2 (programme enregistré) sur la page d'en-tête. Attendez quelques instants que votre programme soit chargé (3 minutes environ). L'écran 1 de PHONETRAM est visualisé sur votre écran. Le système attend le premier phonème du mot.

L'écran de votre ordinateur est divisé en 3 zones. En partant du haut, on trouve :

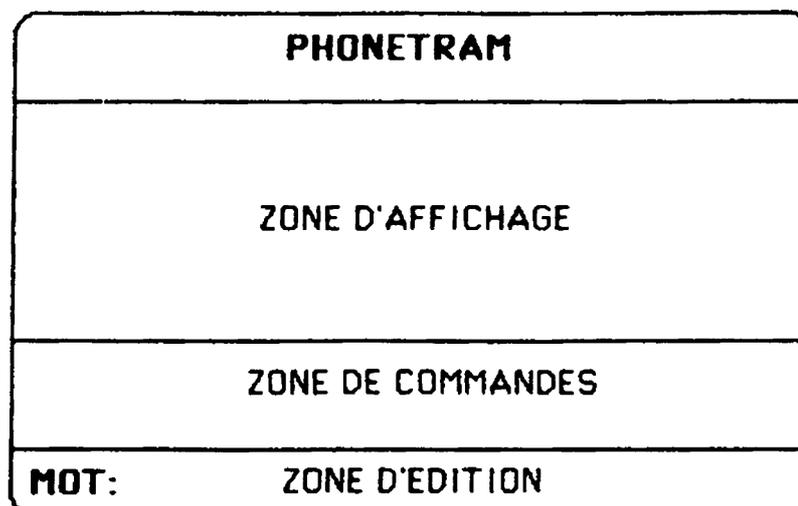
— La zone d'affichage des paramètres. C'est dans cette zone que l'affichage des paramètres pourra être représenté sous deux formes. Une forme dite "numérique" et une forme dite "analogique". L'affichage en cours est l'affichage dit "numérique".

— La zone de commande matérialisée par les pictogrammes contenant les icônes symbolisant les commandes spécifiques.

— La zone d'édition débutant par MOT: destinée à recevoir les caractères représentant les phonèmes.

---

<sup>1</sup> Notez bien: Le modem Télétel ne doit pas être branché sur votre machine car il utilise les mêmes adresses que PAROLE.



## Entrées des phonèmes

Pour commencer, nous vous proposons de “travailler” sur le nom de votre éditeur préféré: CEDIC. Le tableau d’attribution des phonèmes (p. 91 ) donne:

pour C	prononciation s’	le caractère	<b>[s]</b>
pour E	prononciation é	le caractère	<b>[*]</b>
pour D	prononciation d’	le caractère	<b>[d]</b>
pour I	prononciation i’	le caractère	<b>[i]</b>
pour C	prononciation qu’	le caractère	<b>[k]</b>

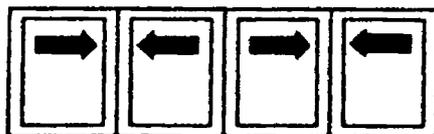
Entrons le premier caractère correspondant au premier phonème **[s]**: PHONETRAM prononce la totalité du mot en cours à chaque entrée de phonème. Pour l’instant, le mot se résume au phonème “s”. Pour le second, **[\*]**, PHONETRAM prononce “sé”. Puis **[d]**, **[i]** et **[k]**.

La barre d’espace permet la répétition du mot en cours stocké dans la zone MOTX. La touche **[←]** autorise le retour en arrière du curseur afin de changer le dernier phonème entré<sup>1</sup>. La touche **[EFF]** efface le mot en cours et repositionne le curseur au début de la zone d’édition. **[RAZ]** est la touche d’abandon de PHONETRAM. Le système retourne sous le contrôle du BASIC.

<sup>1</sup> Notez: Si un des deux curseurs, ou les deux, pointe sur une trame du phonème changé, le logiciel repositionne, à l’issue de cette commande, les deux curseurs sur la première trame de l’expression.

## Les commandes

### Déplacements des curseurs



#### *Définitions :*

Les pictogrammes ci-dessus permettent de déplacer les curseurs sur les trames de l'expression contenue dans la zone mémoire MOTX. Les pictogrammes apparaissant en bleu sur votre écran sont affectés au curseur A, tandis que ceux apparaissant en rouge sont affectés au curseur B. Les flèches indiquent le sens de l'évolution des curseurs sur le mot. La plage d'évolution du curseur A est comprise entre la trame 1 du mot et la trame pointée par le curseur B. Le curseur B peut évoluer de la trame pointée par le curseur A à la fin du mot. Pour chaque mouvement des curseurs, le système calcule, en fonction des paramètres de la trame ouverte et du sens d'évolution du curseur, la "hauteur avant" la trame et la "hauteur après" la trame. Les impossibilités d'évolution des curseurs sont signalées par un bip. Les numéros des trames ouvertes par les curseurs sont affichés sur l'écran.

#### *Application*

En pointant sur les pictogrammes bleus ou rouges flèche gauche, on constate que l'évolution est impossible puisque les deux curseurs se trouvent sur la trame 1 du mot (CEDIC) que nous avons entré précédemment. Seule la flèche rouge droite est active. Le curseur B passe sur la trame 2 puis 3, 4, etc. En arrêtant le curseur B sur une trame quelconque, on vérifie que le curseur A est devenu actif et qu'il peut effectivement évoluer entre la première trame et le curseur B. En positionnant le curseur B sur la dernière trame, on constate que le mot CEDIC comporte 20 trames.

### Prononciation d'une partie de l'expression



#### *Définitions*

Ce pictogramme permet d'envoyer au MEA 8000 les codes vocaux de la partie de l'expression comprise entre le curseur A et le curseur B. Ainsi,

l'utilisateur peut isoler un tronçon de mot, pouvant être au minimum une trame, afin de l'écouter dans le contexte du mot.

### *Application*

En positionnant le curseur A sur la trame 1 et le curseur B sur la trame 20, avant d'activer la commande de prononciation d'une partie de l'expression, on constate que la totalité du mot est prononcée. En modifiant la position des curseurs, on vérifie qu'ils encadrent bien la partie du mot prononcée.

### **Sauvegarde en mémoire auxiliaire**



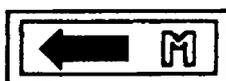
### *Définition*

Ce pictogramme permet de charger la mémoire auxiliaire MOTY avec la partie de l'expression comprise entre les deux curseurs. Dans MOTY, cette partie de l'expression "mère" devient une nouvelle expression. La "hauteur avant" de la trame pointée par le curseur A devient la hauteur initiale de la nouvelle expression. Un bip informe l'utilisateur que la commande a été réalisée.

### *Application*

Nous vous proposons de stocker dans MOTY la totalité du mot (CEDIC) que nous avons entré précédemment. Pour cela, les curseurs A et B doivent encadrer le mot et donc être positionnés respectivement sur les trames 1 et 20. La commande de chargement est alors effectuée en pointant sur le pictogramme adéquat défini ci-dessus. Un bip acquitte la fonction.

### **Restitution de la mémoire auxiliaire**



### *Définition*

Ce pictogramme permet de concaténer la zone MOTX et le contenu de la mémoire auxiliaire MOTY. L'expression contenue dans MOTY est positionnée à partir du curseur B. **La première trame de l'expression prend la place de la trame pointée par le curseur B.** La hauteur initiale de l'expression est chargée comme "hauteur avant" de la trame ouverte par le curseur B. Comme il est peu probable que la hauteur initiale de

l'expression reçue soit égale à la "hauteur avant" de la trame ouverte par le curseur B et pour conserver une similitude entre la hauteur réelle calculée par le MEA 8000 et la hauteur affichée, il convient de repositionner le curseur B sur la trame 1 afin de reporter l'écart éventuel sur la hauteur initiale de l'expression. La concaténation de l'expression contenue dans MOTY est matérialisée, sur la zone d'édition, par un carré rouge et un bip informe l'utilisateur que la commande est terminée.

Si vous souhaitez réaliser des concaténations multiples (écrire plusieurs fois la même expression, à la suite l'une de l'autre), prenez garde de repositionner le curseur B à chaque fois. Dans le cas contraire, la nouvelle expression viendrait "écraser" la précédente.

Notez qu'il n'est pas possible de transférer le contenu de la zone MOTY dans la zone MOTX sans que celle-ci ait été préalablement chargée avec un phonème. C'est le premier chargement dans MOTX qui positionne les pointeurs. Si vous souhaitez que le contenu de MOTY soit en tête dans MOTX, il vous suffit de charger MOTX avec le phonème représenté par le point (.).

### *Application*

Nous vous proposons de concaténer deux mots: CEDIC, sauvegardé en mémoire auxiliaire et un nouveau mot que vous allez entrer, NATHAN, soit en écriture phonétique: natA.

Nous supposons que le mot NATHAN est entré et que le curseur B est positionné sur la dernière trame de NATHAN, soit la trame 18. La commande de restitution est activée. Un carré rouge sur la zone d'édition et un bip indiquent que les mots NATHAN et CEDIC sont concaténés. Pour vérifier, pressez la barre d'espacement de votre micro et écoutez.

Mais là, attention! Un petit piège apparaît car la dernière trame de NATHAN a été écrasée par la première de CEDIC. Alors, comment faire? Et bien, il convient en quelque sorte de prévoir et d'ajouter à la fin du mot un phonème d'une trame qui sera écrasé au moment de la concaténation. Le caractère du phonème qui va bien pour cette tâche, c'est le point (.).

### **Echange entre les deux zones de mémoire MOTX et MOTY**



### *Définition*

Ce pictogramme permet l'échange de l'expression contenue dans la zone MOTX avec celle contenue dans la zone mémoire auxiliaire MOTY. Avec

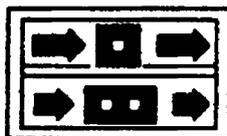
cette commande, les curseurs ne jouent aucun rôle. Deux échanges successifs rétablissent l'état initial. Un bip termine la commande.

### *Application*

Nous avons deux mots à notre disposition, NATHAN et CEDIC, chacun dans une zone mémoire. NATHAN dans MOTX et CEDIC dans MOTY. Nous vous proposons d'échanger ces deux mots. A l'issue de la commande, NATHAN doit être positionné dans MOTY et CEDIC dans MOTX.

La commande est activée, un bip indique que l'échange est réalisé. Une pression sur la barre d'espacement nous le confirme. Une nouvelle commande rétablit l'état initial.

### **Sauvegarde ou restitution d'un mot ou d'une expression dans ou depuis un périphérique**



### *Définition*

Ces pictogrammes autorisent deux fonctions en liaison avec les périphériques.

1. Ils permettent de sauver sur le périphérique assigné le mot ou l'expression contenu dans la zone de mémoire auxiliaire MOTY.
2. Ils autorisent le chargement de la zone de mémoire auxiliaire avec le contenu du fichier binaire spécifié au cours de la commande. Le périphérique spécifié (lecteur de cassettes ou de disquettes) est défini par le pictogramme pointé. Le sens du transfert est visualisé par les flèches contenues dans le pictogramme. Les codes vocaux de l'expression sont sauvegardés sur le support magnétique (disque ou cassette) sous la forme d'un fichier binaire dont le suffixe est forcé à SYT. Ceci afin d'éviter, par erreur, la restitution dans MOTY de fichiers binaires qui ne seraient pas des codes vocaux. Le nom des mots ou expressions sauvegardés est limité à 8 caractères et complété à 8 par des espaces si la longueur du nom est inférieure à 8. A la restitution des codes vocaux dans la mémoire auxiliaire MOTY, seuls sont admis les fichiers binaires dont le suffixe est égal à SYT.

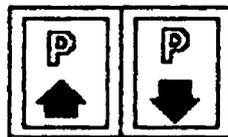
### *Application*

Pour appliquer les commandes que nous venons de définir, nous vous proposons de sauver, sur une cassette ou une disquette, les codes vocaux du mot CEDIC puis de les restituer en mémoire. Nous supposons que la mémoire MOTY est chargée avec les codes vocaux de CEDIC.

Pour cela, il vous faut charger votre lecteur de cassettes ou de disquettes avec le support approprié, puis pointer sur la flèche qui désigne le périphérique souhaité. Le message "NOM A SAUVER" apparaît sur votre écran suivie d'une zone de saisie de 8 caractères. A l'aide du clavier, entrez le nom de votre expression. Par exemple "EDITEUR" pour "s\*dik" ça va bien! La touche **ENTREE** termine la commande et le périphérique pointé est activé. Le message et la zone de saisie sont effacés.

La restitution des codes vocaux dans la mémoire auxiliaire découle d'une procédure similaire, mais le message invitant l'utilisateur à donner le nom du fichier des codes vocaux qu'il souhaite introduire dans MOTY est: "NOM A CHARGER". La touche **ENTREE** termine également la commande. Si vous utilisez un lecteur de cassettes, pensez à repositionner la bande en amont du fichier à lire.

### **Détermination du sens d'évolution des paramètres**



### *Définition*

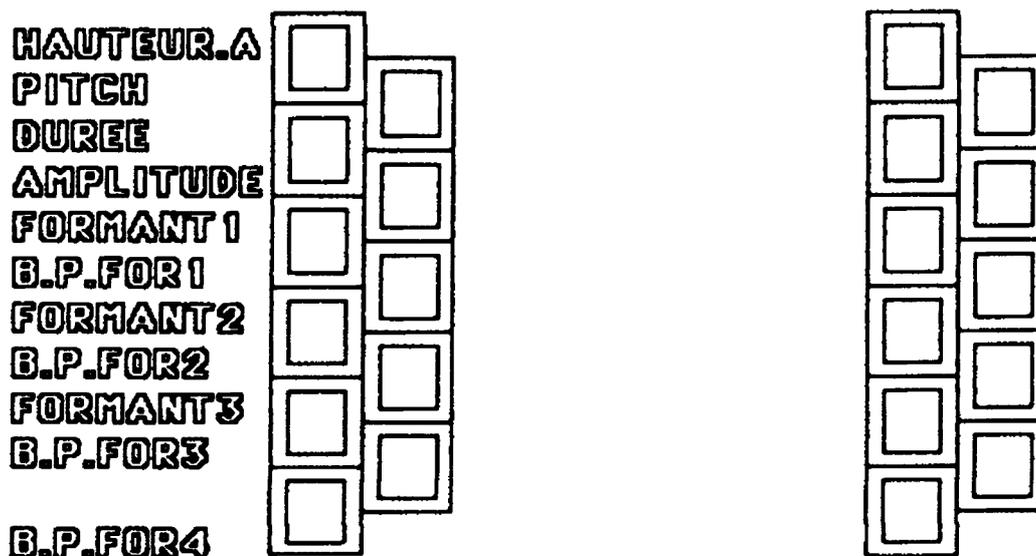
Ces pictogrammes permettent de définir le sens d'évolution des paramètres. La flèche montante indique que les variations des paramètres pointés seront positives (augmentation), tandis que la flèche descendante indique que les variations seront négatives (diminution). La couleur du tour de l'écran indique le sens de l'évolution des paramètres en cours.

Rouge → augmentation. Bleu → diminution.

### *Application*

En pointant le crayon optique sur le pictogramme "flèche montante" vous constatez que le tour de l'écran devient rouge. Ceci vous informe que vous êtes en mesure d'accroître la valeur du paramètre que vous allez pointer. Si vous pointez le pictogramme "flèche descendante" le tour devient bleu, ce qui indique que le prochain paramètre pointé sera diminué.

## Modification des paramètres



Les pictogrammes ci-dessus permettent de modifier le paramètre pointé par le crayon optique. Le sens de variation est défini par la commande d'évolution du sens des paramètres qui a précédé et positionné la couleur du tour de l'écran en conséquence.

Rouge → incrémentation de la valeur binaire du paramètre  
Bleu → décrémentation de la valeur binaire du paramètre

Dans une trame, la "hauteur après" la trame (Hap) est fonction de trois paramètres qui sont :

- La "hauteur avant" la trame, qui peut être soit la "hauteur après" la trame précédente, soit la hauteur initiale quand la trame est la première de l'expression.
- La durée de la trame.
- L'incrément de la variation du pitch (ou incrément de variation de hauteur, en français). L'incrément du pitch sera noté P.I. dans la suite de l'ouvrage.

$$\text{Soit } Hap = Hav + (P.I. * \text{durée})$$

Pour des raisons de commodité d'affinage phonétique, il est pratique de ne pas modifier la prononciation de l'expression située après le curseur pointé. Dans ces conditions, Hap doit rester constante puisque c'est le seul paramètre qui lie les trames entre elles. Une variation du P.I. ou de la durée entraînera une variation en sens inverse d'un des paramètres de la relation ci-dessus, afin de conserver une hauteur en fin de trame constante (Hap). Ainsi, une variation de la durée de la trame entraînera une modification, en sens inverse, du P.I. Une variation du P.I. entraînera un changement de la "hauteur avant" de la trame. Une exception est faite

pour la modification de la “hauteur avant” qui entraîne une modification de la “hauteur après” de la trame.

On distingue donc deux types de paramètres: ceux dont la modification n’a pas de conséquence sur les autres paramètres, et ceux dont la modification entraîne la variation d’un autre paramètre.

Les paramètres suivants (amplitude, formant 1, 2, 3 et bande passante 1, 2, 3, 4) appartiennent à la première catégorie et ne posent pas de problème particulier. La modification s’effectue par incrémentation ou décrémentation, d’une unité, de la valeur binaire spécifiant ce paramètre. La valeur numérique correspondante est affichée à l’écran. Les limites, haute ou basse, du paramètre sont signalées à l’utilisateur par l’envoi d’un bip.

Le P.I., la durée et la “hauteur avant” appartiennent à la seconde catégorie et ont chacun une procédure de modification particulière dans le logiciel.

### **Les variations de l’incrément du pitch (P.I.)**

Le P.I. est exprimé en hertz par 8 millisecondes. Ce qui veut dire que, pour chaque unité de P.I., la hauteur après la trame est incrémentée de un hertz toutes les 8 millisecondes de trame.

**Exemple:** Soit une trame de 16 ms avec un P.I. de 2 et une “hauteur avant” de 120 Hz. La hauteur après la trame est égale à la hauteur avant la trame plus  $(2 * 16)/8$  soit 124 Hz. On souhaite augmenter le P.I. de 1, soit un P.I. de 3. La hauteur après la trame sera donc augmentée de  $(1 * 16)/8 = 2$  Hz soit 126 Hz.

Pour conserver la hauteur après la trame constante, il convient donc de faire évoluer la “hauteur avant” de la trame en sens inverse et avec la même amplitude que la variation attendue à la sortie. Dans notre exemple, la hauteur avant de la trame serait ramenée à 118 Hz ( $120 - 2$ ) soit une hauteur après la trame de  $H_{ap} = 118 + (3 * 2) = 124$  Hz. La hauteur après la trame serait inchangée.

La variation du P.I. entraîne donc, dans PHONETRAM, une modification de la hauteur avant de la trame pointée par le crayon optique. Cette variation de “hauteur avant” est également appliquée sur la trame ouverte par le curseur A si la modification du P.I. a lieu sur le curseur B, ceci afin de conserver une relation de similitude, dans l’intervalle des deux curseurs, entre la hauteur calculée par le MEA 8000 et la hauteur affichée. Si la modification du P.I. a lieu sur le curseur A et que ce curseur ouvre la première trame de l’expression, la variation de hauteur est appliquée sur la hauteur initiale de l’expression.

## Les variations de durée de trame

Nous avons vu précédemment que la durée de trame intervenait dans le calcul de la hauteur en sortie. Une variation de la durée de la trame va donc entraîner une modification de la valeur du P.I. afin de conserver le produit (P.I. \* durée) constant. La durée d'une trame peut être de 8, 16, 32 ou 64 ms correspondant respectivement aux codes binaires 00, 01, 10 et 11.

Une variation de la hauteur binaire de 1 correspond donc à une multiplication par 2 de la durée si la variation est positive et à une division par 2 de la durée si la variation est négative. Dans ces conditions, il convient de diviser ou de multiplier le P.I., à l'opposé de la durée, pour conserver une hauteur en sortie de trame constante.

Un doublement du P.I. peut conduire à un dépassement de la plage autorisée. Dans ce cas, le P.I. est forcé à la valeur maximum et entraîne une variation de la hauteur en sortie de la trame pointée par le curseur.

## La variation de hauteur

La variation de "hauteur avant" de la trame est seulement autorisée sur le curseur A et entraîne une variation de la "hauteur après" de la trame suivant l'équation définie ci-dessus. Cette variation de hauteur est reportée sur le curseur B afin de conserver la similitude entre les hauteurs calculées par le MEA 8000 et les hauteurs affichées à l'écran. Dans le cas où le curseur A "ouvre" la première trame, la "hauteur avant" de la trame est semblable à la hauteur initiale de l'expression. Dans ce cas, les variations de "hauteurs avant" sont également transcrites sur la hauteur initiale de l'expression.

### *Application*

Pour appliquer ce que nous venons de voir sur les modifications des paramètres, nous vous proposons de reprendre le mot "CEDIC" et d'ouvrir, avec les curseurs, la première et la dernière trame du mot. Les valeurs numériques des paramètres sont affichées.

**Premier exemple: modification de l'amplitude.** La valeur de l'amplitude de la trame 1 est .008. En pointant sur le pictogramme "flèche montante", le tour de l'écran devient rouge, vous indiquant que vous êtes dans la phase d'incrémentation des paramètres. Pointez maintenant le crayon optique sur le pictogramme du paramètre que vous souhaitez modifier (l'amplitude sur le curseur A dans notre cas). La valeur du paramètre passe à .125 et la partie de l'expression comprise entre les curseurs A et B est prononcée. En recommençant l'opération, l'amplitude passe à la

valeur .177, etc. Afin de vous familiariser avec les modifications de paramètres, nous vous invitons à essayer ces commandes sur les formants ou les bandes passantes, en changeant éventuellement le sens d'évolution des paramètres et sans trop vous soucier du résultat phonique obtenu.

**Deuxième exemple: *modification du P.I.*** Nous vous proposons de modifier le P.I. de la dernière trame. Le P.I. est positionné à 0 et la durée de la trame est de 16 ms. En pointant sur le pictogramme "flèche descendante", le tour de l'écran devient bleu, vous indiquant que vous êtes dans la phase de décrémentation des paramètres. Pointez maintenant le crayon optique sur le pictogramme du P.I. La valeur du paramètre passe à  $-1$ . et les hauteurs, à l'exception de la "hauteur après" la trame, sont modifiées en fonction de la durée de la trame et de la hauteur précédente, soit  $120 + (-1 * 2) = 118$  Hz. La "hauteur initiale" ou la "hauteur avant" de la trame peuvent être repositionnées à 120 Hz en incrémentant la "hauteur avant" sur le curseur A.

**Troisième exemple: *modification de la durée de la trame.*** Pour ce troisième exemple, nous partons de la situation précédente. Nous vous proposons de réduire à 8 ms la durée de la trame. En pointant sur le pictogramme "flèche descendante", le tour de l'écran devient bleu, vous indiquant que vous êtes dans la phase de décrémentation des paramètres. Pointez votre crayon optique sur le pictogramme de la durée. La valeur de la durée passe à 8 ms et le P.I. à  $-2$ , d'où une hauteur en sortie constante et égale à 118 Hz.

### **Modification de la représentation graphique à l'écran**



#### ***Définition***

Ce pictogramme permet de modifier la présentation graphique sur l'écran. PHONETRAM admet deux pages d'écran (ECRAN 1 et ECRAN 2). Le passage d'un écran à l'autre s'effectue par action sur le pictogramme ci-dessus, quelque soit l'écran affiché. **Ce pictogramme n'est actif que si au moins un phonème est entré sur la ligne d'édition.**

## ***Application***

Après avoir vérifié qu'au moins un phonème est entré sur la ligne d'édition, la commande de modification de représentation graphique à l'écran est active et peut être utilisée en pointant sur le pictogramme avec le crayon optique. L'ECRAN 2 est alors visualisé. Une nouvelle commande permet de revenir sur l'ECRAN 1.

**ECRAN 1:** Cet écran visualise les trames ouvertes par les curseurs et génère les pictogrammes de modification des paramètres. Chaque paramètre des trames ouvertes par les curseurs est présenté par une barre d'élongation proportionnelle à la valeur binaire du paramètre. Un affichage visualise également la valeur numérique des paramètres. Le rang de la trame ouverte, par l'un ou l'autre des curseurs, apparaît à l'écran. Le nom des paramètres, représentés sur la gauche de l'écran et écrits en noir sur fond de couleur identifiant le paramètre spécifié, concerne les deux curseurs. Le curseur A est le plus à gauche (trame bleue) tandis que le curseur B (trame rouge) est positionné sur la droite de l'écran.

**ECRAN 2:** Visualise la courbe de la hauteur du son en fonction du temps entre les deux curseurs. Chaque trame occupe un point caractère de l'écran qui est plus ou moins rempli en fonction de la durée de la trame.

1/8 de point caractère	égal	8 ms
2/8 de point caractère	égal	16 ms
4/8 de point caractère	égal	32 ms
8/8 de point caractère	égal	64 ms

Afin de faciliter la lecture, chaque trame est représentée dans une couleur différente. Les trames générant un son non voisé (bruit) sont représentés en blanc. Chaque début de phonème est représenté sur l'axe des X par le caractère spécifiant ce phonème, ce qui permet à l'utilisateur de pointer précisément le curseur de son choix sur la trame qu'il souhaite ouvrir afin d'en modifier certains paramètres. La "hauteur avant" le curseur A est utilisée comme point de référence et est visualisée sur la gauche de l'écran. Les valeurs supérieures à la référence sont définies par rapport au bas de la fenêtre, tandis que les valeurs inférieures sont définies par rapport au haut de la fenêtre.

# Logiciel

PHONETRAM est un programme presque exclusivement construit en ASSEMBLEUR et dont le programme objet est appelé sous contrôle du BASIC. La partie binaire du programme est constituée de deux blocs: PHONTRAM.BIN et VALDEF.BIN

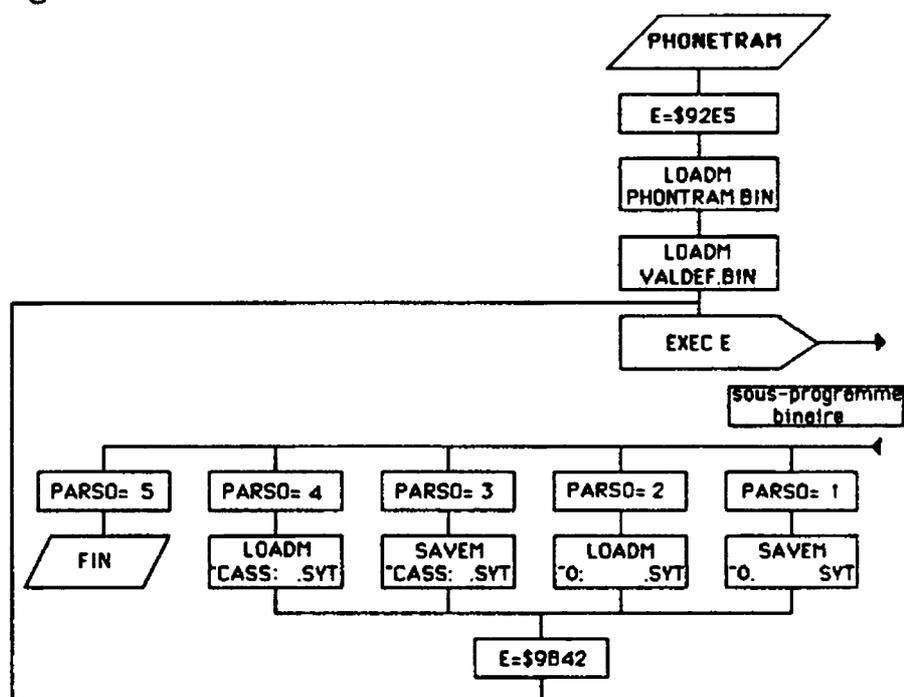
PHONTRAM.BIN est le sous-programme binaire ou objet issu de l'assemblage du programme source décrit dans la suite de cet ouvrage. VALDEF.BIN contient les fichiers binaires nécessaires au fonctionnement de PHONETRAM (table des phonèmes, caractères utilisateur, table des valeurs numériques des paramètres, etc.).

Le logiciel présenté ici est une version simplifiée de PHONETRAM, commercialisé par CEDIC/NATHAN.

## Le programme BASIC

L'adresse d'exécution du sous-programme binaire est définie par la variable E. Au retour du sous-programme, le registre PARSO contient une valeur qui indique au programme BASIC la séquence qu'il convient d'exécuter. Cette séquence terminée, la variable E est chargée avec une autre adresse d'exécution et retourne dans le programme binaire (pour plus de détails, voir les programmes ASM: CASSS-CASSL-DISQL-DISQS pages 226 à 230).

### Organigramme

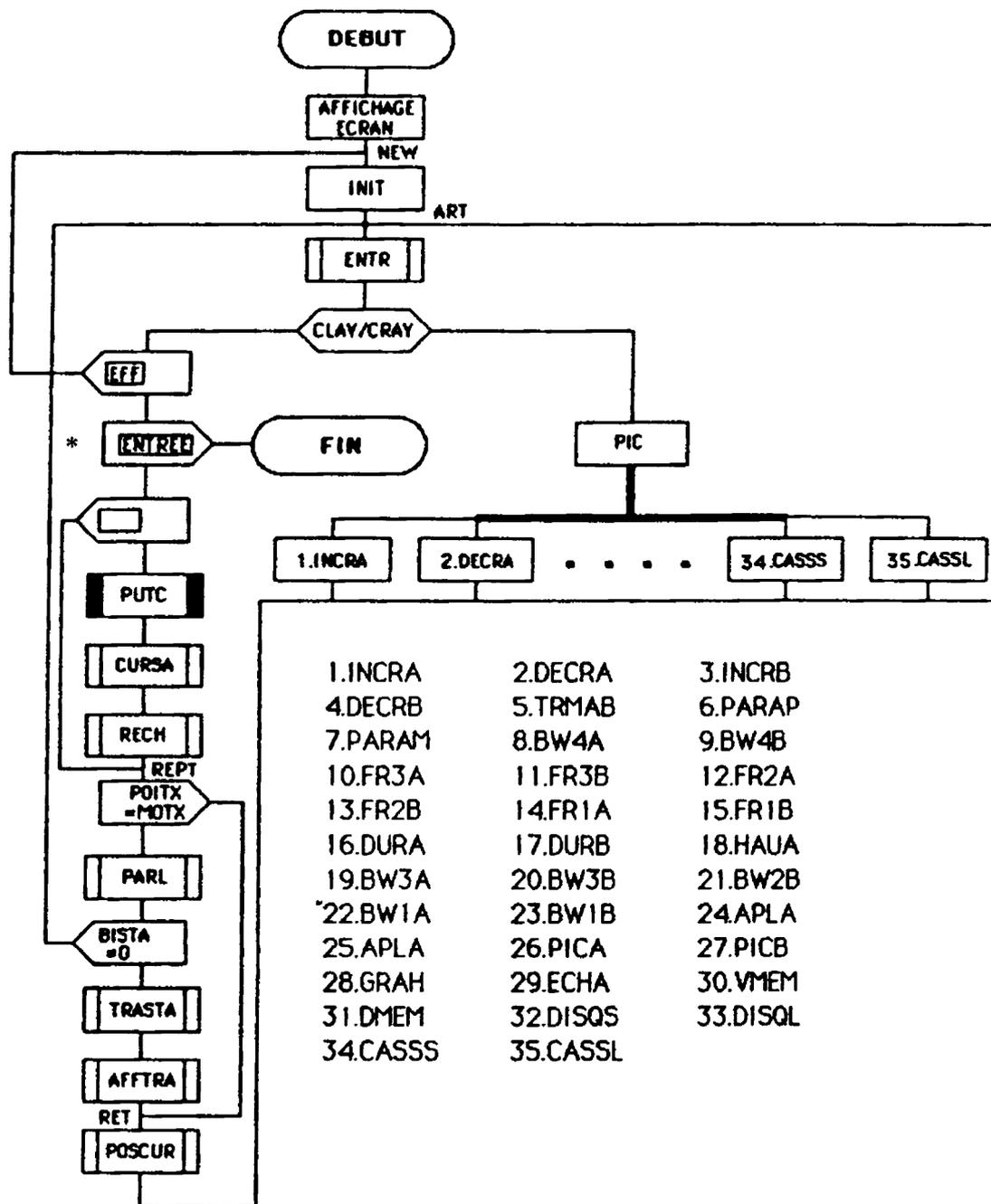


## Listing

```
10 CLEAR,&H9000
12 E=&H92E5 'START
15 SCREEN0,0,0
20 LOADM"0:PHONTRAM.BIN",0000
30 LOADM"0:VALDEF.BIN",0000
40 EXEC E 'START
45 I=PEEK(&H92CB)'PARSO
46 ON I GOSUB50,100,150,200,300
47 E=&H9B42 'EFAC
49 GOTO 40
50 A$="0:"
55 FOR I=0 TO 7
60 A$=A$+CHR$(PEEK(&H92D2+I))
70 NEXT I
75 A$=A$+".SYT"
80 I=256*(PEEK(&H92DA))+(PEEK(&H92DB))'A
DF
90 SAVEM A$,&HB56E,I,0000
95 RETURN
100 A$="0:"
110 FOR I=0 TO 7
120 A$=A$+CHR$(PEEK(&H92D2+I))
130 NEXT I
140 A$=A$+".SYT"
145 LOADM A$,0000
148 RETURN
150 A$="CASS:"
155 FOR I=0 TO 7
160 A$=A$+CHR$(PEEK(&H92D2+I))
170 NEXT I
175 A$=A$+".SYT"
180 I=256*(PEEK(&H92DA))+(PEEK(&H92DB))
190 SAVEM A$,&HB56E,I,0000
195 RETURN
200 A$="CASS:"
210 FOR I=0 TO 7
220 A$=A$+CHR$(PEEK(&H92D2+I))
230 NEXT I
240 A$=A$+".SYT"
245 LOADM A$,0000
248 RETURN
300 STOP
```

# Le programme source ASSEMBLEUR

## Organigramme général



\* RAZ dans la version "CEDIC"

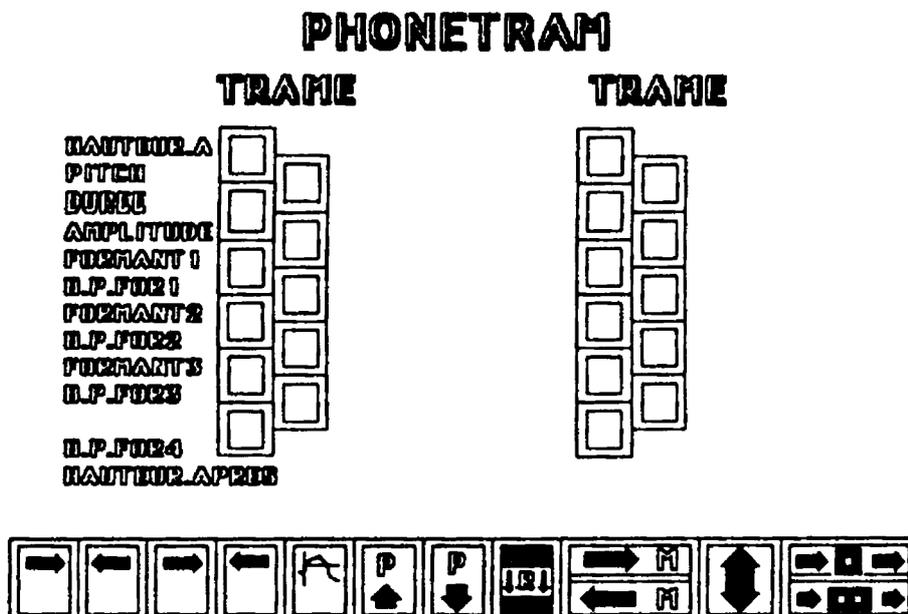
# Affichage d'écran

La section du programme principal intitulé "AFFICHAGE D'ECRAN" génère sur l'écran la zone de commandes (pictogrammes) et la partie fixe de la première des deux fenêtres dite "écran A". Les motifs d'écran sont construits à partir de caractères utilisateurs contenus dans trois tables étiquetées: CARUTI, CARUT1 et CARUT2 auxquelles sont associées les zones d'écran: ECRAN, ECRAN1 et ECR1B. MESSA est la routine d'écriture des caractères pointés par le registre X.

## Registres et drapeaux utilisés ou modifiés:

CARUTI: Début d'une table de caractères utilisateurs  
 CARUT1: Début d'une table de caractères utilisateurs  
 CARUT2: Début d'une table de caractères utilisateurs  
 ECRAN: Zone de l'écran  
 ECRAN1: Zone de l'écran  
 ECR1B: Zone de l'écran  
 USERAF: Pointeur sur le générateur de caractères utilisateur  
 RTNU: Retenue pour le report de la variation du P.I.  
 CHAR: Indique qu'un mot est chargé dans la zone mémoire Y  
 BISTA: Indique l'état de l'affichage

Ecran généré par "affichage d'écran" :



**NOT:**

### Listing source

```
DEBUT  LDU    #SGAR+3  Sauvegarde
        PSHU   DP,S
        LDX   #CARUT1  Debut de table
        STX   USERAF   Pointeur
        LDX   #ECRAN   Zone d'ecran
        JSR   MESSA
        LDX   #CARUT1
        STX   USERAF
        LDX   #ECRAN1
        JSR   MESSA
        LDX   #CARUT1
        STX   USERAF
        LDX   #ECRAN1
        JSR   MESSA
        LDX   #CARUT2
        STX   USERAF
        LDX   #ECR1B
        JSR   MESSA
```

## MESSA:

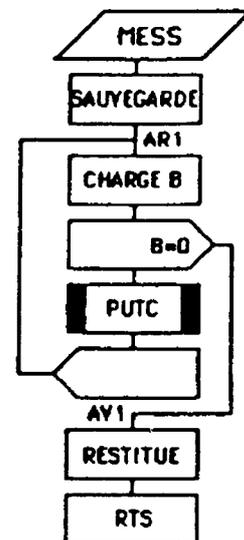
**Registres et drapeaux utilisés ou modifiés:**

PUTC: Routine moniteur d'affichage d'un caractère

### Listing source

```
MESSA  PSHS   B
AR1    LDB   /X+
        BEQ   AV1
        JSR   PUTC
        BRA   AR1
AV1    PULS  B,PC
```

### Organigramme



## INIT:

Les registres et drapeaux qui doivent être initialisés à une valeur particulière sont chargés dans le tronçon de programme principal appelé INIT.

## Registres et drapeaux utilisés ou modifiés:

MOTX: Début de la zone mémoire X  
POITX: Pointeur sur la zone mémoire X  
TABPOS: Début de la zone mémoire intitulée "table de positionnement"  
POITX: Pointeur sur la table de positionnement  
BLOQ: Drapeau de blocage d'entrées invalidées  
PCAR: Drapeau d'indication du premier caractère  
NUTRA: Chaîne de caractères pour l'affichage du numéro de la trame A  
NUTRB: Chaîne de caractères pour l'affichage du numéro de la trame B  
PUTC-

## Listing source

```

          CLR      RTNU      Retenue Picht
          CLR      CHAR      Mot dans My
          CLR      BISTA     Affichage
NEW      LDX      #MOTX     Initialise POITX
          STX      POITX
          LDX      #TABPOS   Force a zero la
* table TABPOS
DF1      CLR      ,X+
          CMPX     #TABPOS+120T
          BNE      DF1
          LDX      #TABPOS   Initialise POITX1
          STX      POITX1
          LDB      #$1B      Tour bleu clair
          JSR      PUTC
          LDB      #$66
          JSR      PUTC
          CLR      BLOQ      Valide les entrees
* Principales
          INC      BLOQ
          CLR      PCAR      Pour indiquer le
* Premier caractere
          CLR      MOTX+1
          LDD      #$3031     Init a 01
          STD      NUTRA+9
          STD      NUTRB+9
          LBRA     PMO
```

## ENTR:

Le sous-programme "ENTR" permet de ramener soit dans A soit dans X et Y les données du clavier ou du crayon optique. Le bit C du registre de condition indique si les paramètres sont valides dans A (C = 1) ou X et Y (C = 0).

## Registres et drapeaux utilisés ou modifiés :

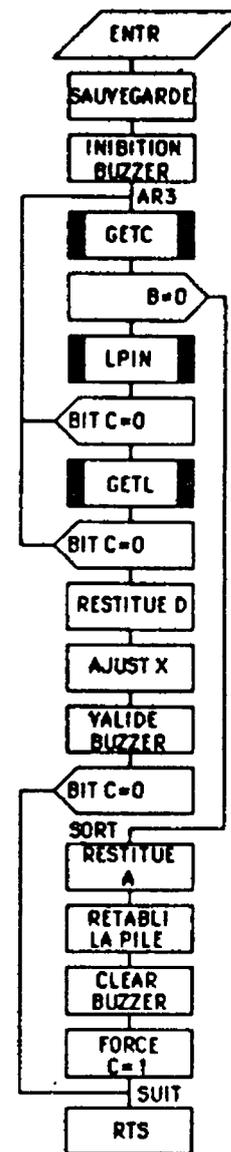
**BUZZ:** Drapeau d'extinction du buzzer  
**GETC:** Routine moniteur de lecteur du clavier  
**LPIN:** Routine moniteur de contrôle du bouton du crayon optique  
**GETL:** Routine moniteur de lecteur du crayon optique

## Listing source

```

ENTR  PSHS  D
      LDA  #1
      STA  BUZZ
AR3   JSR  GETC
      TSTB
      BNE  SORT
      JSR  LPIN
      BCC  AR3
      JSR  GETL
      BCS  AR3
      PULS D
      LEAX 10,X
      CLR  BUZZ
      BRA  SUIT
SORT  PULS  A
      LEAS 1,S
      CLR  BUZZ
      ORCC #01
  
```

## Organigramme



## CURSA :

CURSA charge le registre RECUR avec la position courante du curseur sur la ligne d'édition des phonèmes.

## Registres et drapeaux utilisés ou modifiés:

**RANG:** Donne la dernière position en Y du curseur

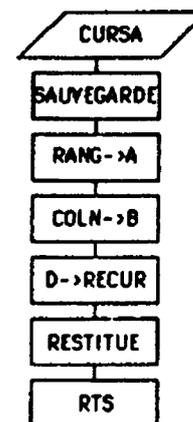
**COLN:** Donne la dernière position en X du curseur

**RECUR:** Registre 16 bits qui conserve les positions du curseur après un affichage sur la ligne d'édition.

## Listing source

```
CURSA PSHS D,X,CC
      LDA RANG Sauve RANG
      LDE COLN Sauve COLN
      STD RECUR Conserve dans RECUR
      PULS D,X,CC,PC
```

## Organigramme



## RECH:

Le sous-programme RECH permet de charger la zone réservée MOTX avec les codes vocaux extraits de la table des données DONE. Le code ASCII du caractère représentant le phonème est passé par le registre B du 6809. La table de positionnement TABPOS est chargée avec le code ASCII du caractère suivi de l'adresse de début du phonème dans la zone MOTX. Ceci afin de positionner le début du phonème, dans la table MOTX, en prévision d'un affichage en mode graphique.

Il y a débordement (branchement à DEBOR) si une tentative de chargement est effectuée alors que MOTX ne dispose plus d'assez de place pour recevoir les codes vocaux attachés au dernier phonème. Le branchement à DEBOR est également réalisé quand le caractère entré au clavier ne correspond pas un caractère affecté à un phonème dans la table des adresses relatives TABL.

DEBOR modifie l'adresse de retour du sous-programme. ART devient l'adresse de retour après passage dans DEBOR. Rien n'est chargé dans MOTX ou TABPOS s'il y a passage par DEBOR.

## Registres et drapeaux utilisés ou modifiés :

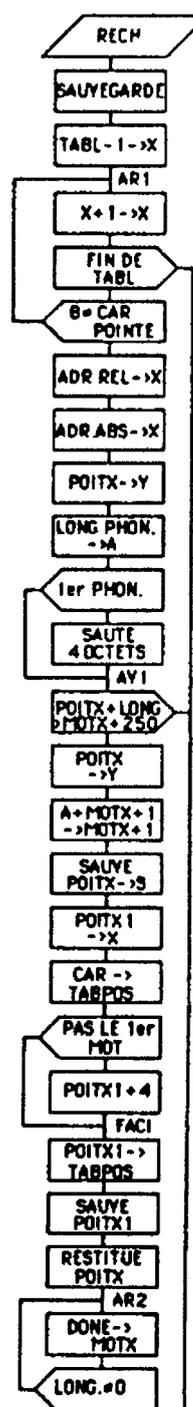
**TABL:** Début de la table des adresses relatives  
**DONE:** Début de la table des données vocales. Fin de TABL  
**ART:** Adresse de retour pour une nouvelle entrée  
**POITX-MOTX-POITX1**

## Listing source

```

RECH  PSHS  X,Y,D,U
      LDX  #TABL-1 Pour la boucle AR1
AR1   LEAX 1,X
      CMPX #DONE  Fin de table?
      BEQ  DEBOR
      CMPB ,X++  Compare B avec le
* caractere pointe dans la table
      BNE  AR1
      LDX  -1,X  Adresse relative
      LEAX TABL,X Adresse absolue
      LDY  POITX Pointeur sur la
* table MOTX
      LDA  ,X+  Longueur dans A
      CMPY #MOTX 1er Phoneme?
      BEQ  AV1
      SUBB #4  Saute 4 octets
      LEAX 4,X
      LEAY B,Y  Pour tester la
* limite de la zone reservee MOTX
      CMPY #MOTX+250T
      BHS  DEBOR Debordement de MOTX
      LDY  POITX Retabli l'etat de Y
      ADDA MOTX+1 Calcul la nouvelle
* longueur de l'expression
      STB  MOTX+1
      FSHS X  Table de Position-
* nement
      LDX  POITX1 Pointeur sur TABPOS
      LDB  3,S  Caractere dans TABPOS
      STB  ,X+
      TFR  Y,U  Pointeur dans U
      CMPY #MOTX 1er Phoneme?
      BNE  FACI  Non.
      LEAU 4,U  Oui. Alors +4 sur
* le Pointeur
FACI  STU  ,X++ Adresse dans MOTX
* sauvee dans TABPOS
      STX  POITX1 Sauve le Pointeur
      PULS X  Restitue le Poin-
* eur dans MOTX
AR2   LDB  ,X+  Transfert les don-
* nees de DONE vers MOTX
      STB  ,Y+
      DECA  Fin?
      BNE  AR2  Non.
  
```

## Organigramme

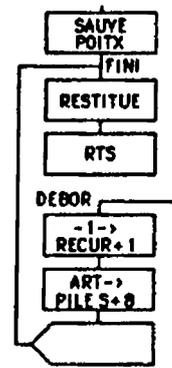


```

FINI   STY   POITX   Oui.Sauve POITX
       PULS  X,Y,D,U,PC

DEBOR  DEC   RECUR+1 Pour effacer le
* dernier caractere
       LDX   #ART   Adresse de retour
       STX   BT,S   dans la Pile systeme
       BRA   FINI

```



## PARL:

PARL est le sous-programme de contrôle du MEA 8000. Le registre X pointe sur le premier octet de la zone MOTX. Avant d'exécuter PARL, le programme principal vérifie si POITX est différent de MOTX. Si la réponse est négative, PARL n'est pas exécuté car aucun phonème n'est entré dans MOTX.

Le registre de commande du MEA 8000 est positionné en mode ARRET LENT et REQ inactif. La dernière boucle (AR3) permet d'attendre la prononciation de la dernière trame pour retourner au programme principal.

### Registres et drapeaux utilisés ou modifiés:

RCOM: Registre de commande du MEA 8000  
 DONE: Registre de données du MEA 8000

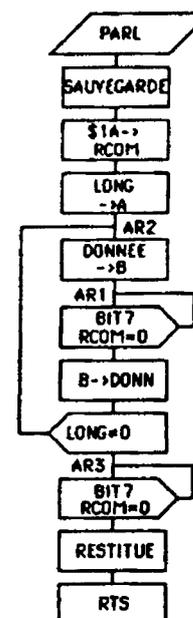
### Listing source

```

PARL   PSHS   D,U
       LDB   #1A   Positionne le reg-
* istre de commande en mode ARRET-LENT
*  et REQ inactif
       STB   RCOM
       LEAX  1,X   Sautte l'octet "00"
       LDA   ,X++  Longueur du mot
       SUBA  #3    Deja 3 octets de
* Passes
AR2    LDB   ,X+   Donnee
AR1    TST   RCOM  Demande?
       BPL   AR1
       STB   DONN  Oui.Donnee dans DONN
       DECA
* sur la longueur
       BNE   AR2   Fin?
AR3    TST   RCOM  Oui.On attend la
* fin de la derniere trame
       BPL   AR3
       PULS  D,U,PC  Vraiment la fin!

```

### Organigramme



## TRASTA:

Le sous-programme TRASTA permet de charger les zones TABT1 et TABT2 qui sont, respectivement, les tables affectées aux "curseurs" A et B. On appelle "curseurs" les trames pointées par PTA et PTB. Les mouvements des curseurs sont obtenus en ajoutant ou retranchant 4 au pointeur de la trame que l'on souhaite modifier.

TRASTA calcule la hauteur à la sortie de la trame en tenant compte de la hauteur en entrant dans la trame, du P.I. et de la durée. Cette valeur est la hauteur d'entrée de la trame suivante.

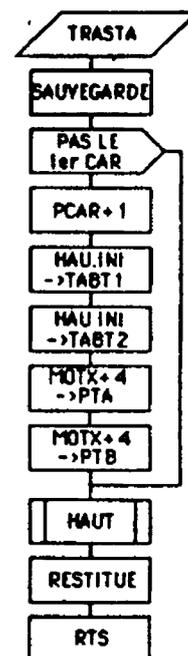
### Registres et drapeaux utilisés ou modifiés:

TABT1: Premier octet de la table affectée à la trame A (bleue)  
TABT2: Premier octet de la table affectée à la trame B (rouge)  
PTA: Pointeur de la trame A sur MOTX  
PTB: Pointeur de la trame B sur MOTX  
PCAR-MOTX

### Listing source

```
TRASTA PSHS X,D
      TST PCAR Premier caractere?
      BNE QSD
      INC PCAR Le 1er caractere
* est Passe
      LDA MOTX+3
      STA TABT1+2 Charge la hauteur
* initiale dans TABT1
      STA TABT2+2 dans TABT2
      LDX #MOTX+4
      STX PTA Pointeur trame A
      STX PTB Pointeur trame B
*Al'initialisation PTA=PTB
QSD JSR HAUT Pour afficher les
* trames Pointees
      PULS X,D,PC
```

### Organigramme



## HAUT:

Le sous-programme HAUT permet le passage des paramètres d'entrée au sous-programme HPD. Ces paramètres sont passés par les registres X et Y. Le registre X pointe sur le premier octet de la trame à écrire dans la table spécifiée par Y.

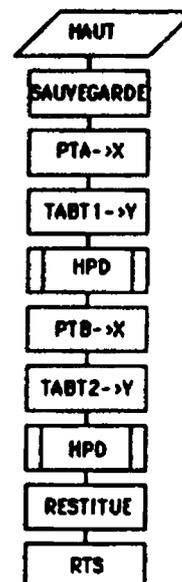
## Registres et drapeaux utilisés ou modifiés

PTA-PTB-TABT1-TABT2

### Listing source

```
HAUT    PSHS    X,Y
        LDX    PTA      Pointeur A ->X
        LDY    #TABT1   TABT1 ->Y
        JSR    HPD
        LDX    PTB      Pointeur B ->X
        LDY    #TABT2   TABT2 ->Y
        JSR    HPD
        PULS   X,Y,PC
```

### Organigramme



## HPD:

HPD est le sous-programme d'affectation des paramètres de la trame pointée par le registre X dans la table spécifiée par Y. Le format des tables TABT1 et TABT2 est le suivant : Chaque paramètre est défini par 4 octets suivis d'une adresse qui désigne le tableau des valeurs numériques autorisées pour ce paramètre.

OCTET 1: Couleur d'affichage      OCTET2: Coefficient d'échelle  
OCTET 3: Valeur binaire            OCTET4: Position en Y sur l'écran  
OCTETS 5 et 6: Adresse des valeurs numériques

L'ordre des paramètres est le suivant :

1-HAUTEUR AVANT	2-P.I.
3-DUREE	4-AMPLITUDE
5-FORMANT 1	6-L.BANDE 1
7-FORMANT 2	8-L.BANDE 2
9-FORMANT 3	10-L.BANDE 3
11-FORMANT 4	12-HAUTEUR APRES

Le drapeau DECINC définit le sens de variation des curseurs ce qui permet de reporter les variations de hauteur d'une trame sur l'autre et cela dans les deux sens.

## Registres et drapeaux utilisés ou modifiés

DECINC: Définit le sens d'évolution du curseur (FF sens négatif (5, 4, 3, etc.)).

### Listing source

```
HPD    PSHS    X,Y,D
        LDA    3,X    Octet 4 de la trame
        TFR    A,B
        ANDA   #$1F   Picth
        STA    8,Y    Dans la table
        ROLB                   Positionne la DUREE
        ROLB                   au debut de l'octet
        ROLB
        ROLB
        ANDB   #$03   Duree dans la table
        STB    14T,Y
        CMPA   #15T   Picth Positif?
        BLS    POSI   Oui.
        CMPA   #16T   Non.Alors bruit ?
        BEQ    EGAL   Oui.
        NEGA                   Non.Alors negatif
        ANDA   #$0F   Masque
        BRA    SUIT1
EGAL    CLRA                   Bruit K=0
POSI    JSR    DELTA   Calcul la valeur
* a ajouter a la hauteur initiale
        LDB    #$FF
        CMPB   DECINC   Sens du report:
* trame precedente ou trame suivante?
        BEQ    AS1     Precedente
        ADDA   2,Y     Suivante
        STA    68T,Y   Hauteur apres
        BRA    OKP
AS1     LDB    68T,Y
        STA    2,Y
        SUBB   2,Y
        STB    2,Y     Hauteur avant
        BRA    OKP
SUIT1   JSR    DELTA   Calcul la valeur
* a retrancher a la hauteur initiale
        LDB    #$FF
        CMPB   DECINC   Sens du report
        BEQ    AS2
        LDB    2,Y
        STA    68,Y
        SUBB   68,Y
        STB    68,Y   Hauteur apres
        BRA    OKP
AS2     ADDA   68,Y
        STA    2,Y     Hauteur avant
```

```

OKP      LDD      2,X      Octet 3 et 4 de la
* trame
LSRA
RORB      Cadrage de l'ampli-
LSRA      tude et FORMANT 1
RORB
LSRA
RORB
LSRB
LSRB
LSRB
LSRB
STA      26T,Y      Formant 1
STB      20T,Y      Amplitude
LDA      1,X      Octet 2 de la trame
ANDA     #51F
STA      38T,Y      Formant 2
LDA      1,X      Octet 2 de la trame
ROLA
ROLA
ROLA
ROLA
ANDA     #07
STA      50T,Y      Formant 3
LDA      ,X      Octet 1 de la tra-
* me
LSLA
ROLB
LSLA
ROLB
ANDB     #03
STB      32T,Y      L. Bande 1
LSLA
ROLB
LSLA
ROLB
ANDB     #03
STB      44T,Y      L. Bande 2
LSLA
ROLB
LSLA
ROLB
ANDB     #03
STB      56T,Y      L. Bande 3
LSLA
ROLB
LSLA
ROLB
ANDB     #03
STB      62T,Y      L. Bande 4
PULS     X,Y,D,PC

```

## DELTA:

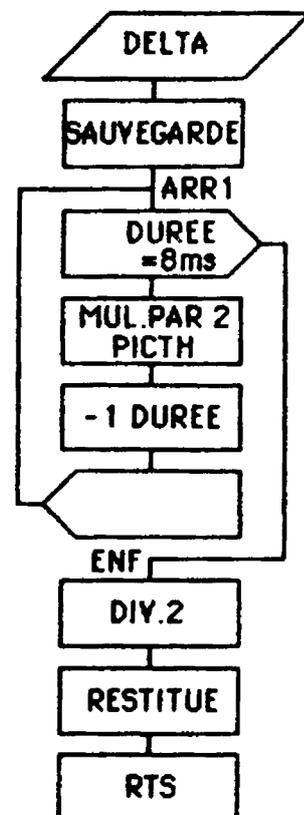
Le sous-programme DELTA calcule la variation de hauteur qu'il convient d'ajouter ou retrancher à la hauteur initiale pour obtenir la hauteur en sortie de la trame. Cette valeur est fonction du P.I. et de la durée de la trame. La valeur du P.I. est passée par le registre A tandis que la durée est passée par le registre B.

Comme le P.I. est exprimé en Hz/8 ms, la variation de hauteur est égale au nombre de fois 8 ms dans la durée. La division par deux est nécessaire pour tenir compte du fait que la hauteur est exprimée avec un pas de deux Hz.

### Listing source

```
DELTA  PSHS  B
ARR1   TSTB
      BEQ   ENF      Code Duree= 0
      LSLA
      DECB
      BRA   ARR1
ENF     LSRA      Div.Par 2 Pour tenir
* compte de l'increment de la hauteur
      PULS  B,PC
```

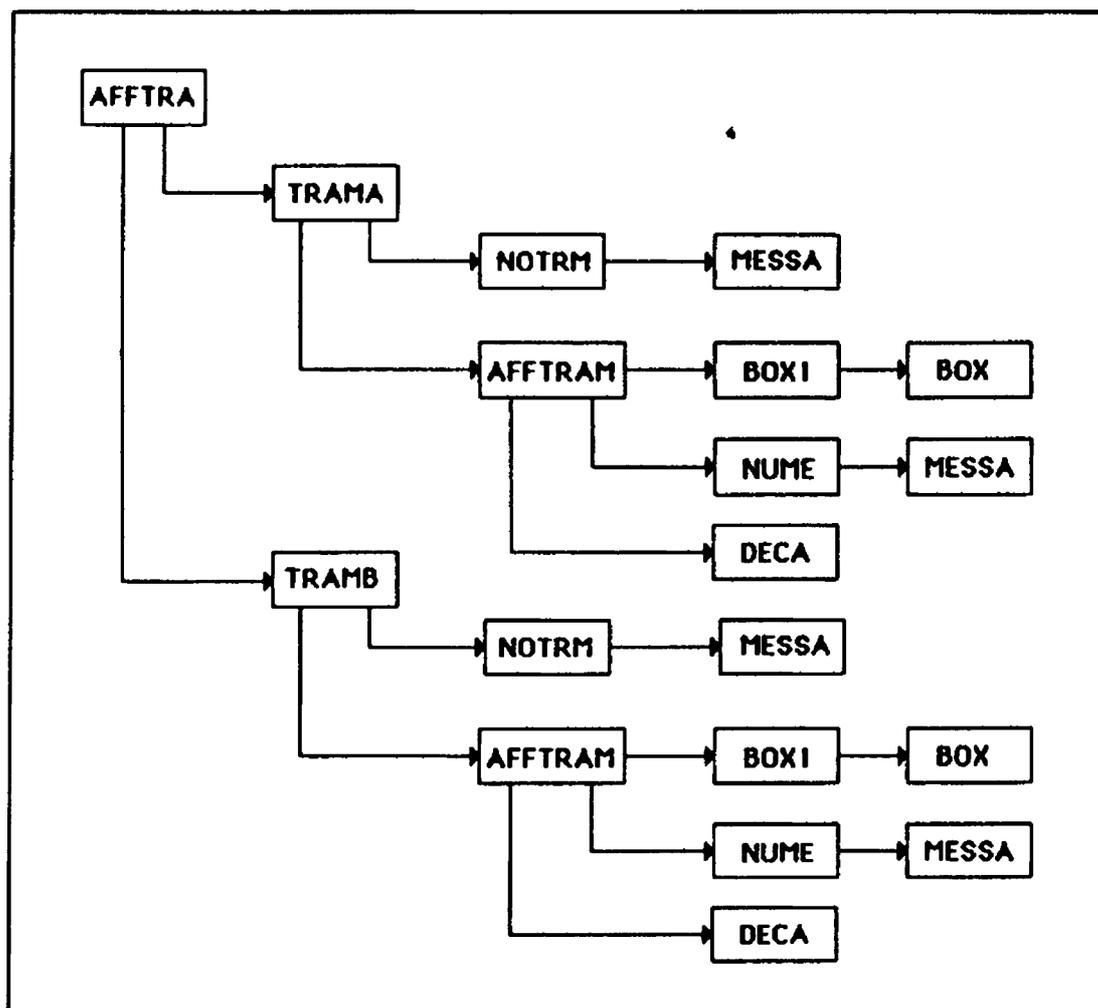
### Organigramme



## AFFTRA:

AFFTRA est le sous-programme d'affichage des deux curseurs (A et B). Les paramètres d'affichage sont extraits des tables TABT1 et TABT2. AFFTRA doit être précédé de TRASTA si l'on souhaite modifier l'affichage.

## Tableau des dépendances de AFFTRA:

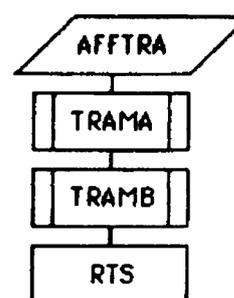


### Listing source

```

AFFTRA JSR      TRAMA  Curseur A
        JSR      TRAMB  Curseur B
        RTS
  
```

### Organigramme



## TRAMA:

TRAMA initialise pour la trame A (curseur A) les paramètres d'entrée nécessaires aux sous-programmes appelés par TRAMA.

## Registres et drapeaux utilisés ou modifiés

**NUTRA:** Chaîne de caractères pointée par NUTRA qui permet le positionnement et l'affichage du numéro de la trame A.

```
NUTRA    FCB          $1B,$44,$1B,$53,$1B,$4E
          FCB          $1F,$43,$55,$30,$31,$1B
          FCB          $4C,00
```

**AFIG:** Chaîne de caractères pointée par AFIG qui permet le passage d'attributs de couleur et de positionnement.

```
AFIG     FCB          $1B,$40,$1B,$50
PODC     FCB          $1F,$40,$40,00
```

**TRAM:** Registre 16 bits spécifiant la position du point zéro des barres analogiques du curseur

**TABT:** Pointeur chargé à TABT1 ou TABT2

**FTAB1:** Fin de la table TABT1

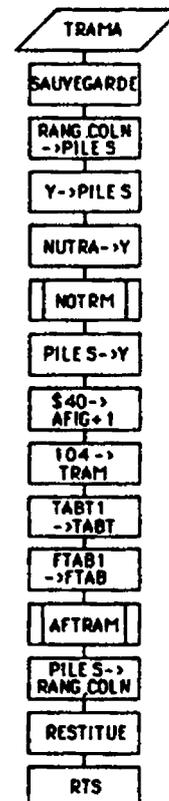
**FTAB2:** Fin de la table en cours. Doit être chargé à FTAB1 ou FTAB2

**RANG-COLN-TABT1**

## Listing source

```
TRAMA    PSHS    D,X
          LDA     RANG    Sauvegarde la der-
* niere Position du curseur
          LDB     COLN
          PSHS    D
          PSHS    Y      Sauve l'etat de Y
          LDY     #NUTRA  Chaîne de car. qui
* Permet le positionnement et l'afficha-
* -ge du numero de trame A
          JSR     NOTRM   No de trame
          PULS    Y      Restitue Y
          LDA     #$40    Ecriture noire
* Pour effacer le nb Precedent.
          STA     AFIG+1
          LDX     #104T   Point zero de la
* trame A ....
          STX     TRAM    ..dans de debut de
* trame A
          LDX     #TABT1  Table TABT1....
          STX     TABT    ..dans le registre
* de table en cours de traitement
          LDX     #FTAB1  Fin de TABT1....
          STX     FTAB    ..dans le registre
* de fin de trame
          JSR     AFTRAM  Affichage du cur-
* seur A(ou trame A)
          PULS    D      Restitue la Posi-
* tion du curseur
          STA     RANG
          STB     COLN
          PULS    D,X,PC
```

## Organigramme



## NOTRM:

NOTRM est le sous-programme de génération et d'affichage du numéro de la trame. En entrant dans NOTRM, le registre Y doit spécifier la chaîne à prendre en compte (NUTRA ou NUTRB). En fonction de l'état de DECINC, le numéro de la trame est incrémenté ou décrétementé de un par modification des octets de la chaîne. Si DECINC vaut zéro, le numéro de la trame est inchangé.

### Registres et drapeaux utilisés ou modifiés

NUTRX: Chaîne de caractères qui définit les codes ASCII des limites nécessitant un report sur le digit suivant ou précédent, en fonction d'un comptage ou d'un décomptage. NUTRX spécifie également le code ASCII du caractère de remplacement.

NUTRX FCB \$3A,\$30,\$2F,\$39

NUTRB: Chaîne de caractères pointée par NUTRB qui permet le positionnement et l'affichage du numéro de la trame B.

NUTRB FCB \$1B,\$41,\$1B,\$53,\$1B,\$4E  
 FCB \$1F,\$43,\$64,\$30,\$31,\$1B  
 FCB \$4C,00

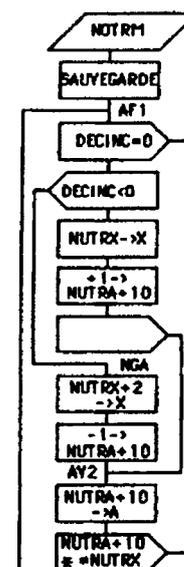
### DECINC-NUTRA

#### Listing source

```

NOTRM PSHS X,Y,D
AF1 TST DECINC Teste si le no
* de la trame doit etre modifie
      BEQ FIN Non.Alors FIN
      BMI NGA DECINC negatif?
      LDX #NUTRX Non.
      INC 10,Y Incrmente
      BRA AV2
NGA LDX #NUTRX+2 Oui.
      DEC 10,Y Decrmente
AV2 LDA 10,Y Comparaison aux
* limites Pointees Par X
      CMPA ,X
      BNE FIN Non e9ale.FIN
  
```

#### Organigramme



```

LDA 1,X      Charge l'unité
STA 10,Y
LEAY -1,Y    Pointe sur les
* dizaines
CMPY #NUTRA+8  Dépassement des
* des limites 01->99
BEQ FIN
CMPY #NUTRB+8
BEQ FIN      Idem pour les
* dizaines
BRA AF1
FIN LDX 4,S   Charge X avec l'
* adresse passée par Y
JSR MESSA    Envoie le message
PULS X,Y,D,PC

```



\* ou NUTRX+2      \* ou NUTRX+3

## AFTRAM:

L'affichage de la trame, pointée par le registre TABT, est assuré par le sous-programme AFTRAM. A la sortie de AFTRAM, la totalité des paramètres est affichée à l'écran sous forme NUMERIQUE et représentée en ANALOGIQUE par une barre de longueur proportionnelle à la valeur du paramètre.

### Registres et drapeaux utilisés ou modifiés

FORME: Registre qui permet la mise en couleur  
TAPT-FTAB

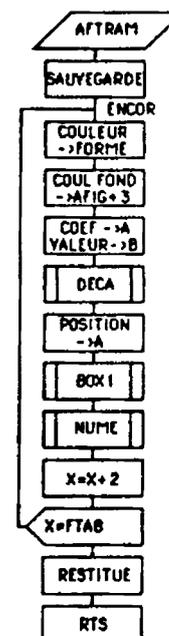
### Listing source

```

AFTRAM PSHS X,Y,A
LDX TABT   Charge le pointeur
ENCOR LDA ,X+   Charge la couleur.
STA FORME  ..dans FORME
ADDA #$50  Force le fond avec
* la couleur spécifiée
STA AFIG+3
LDD ,X++  Coefficient d'échelle
BSR DECA  Calcul
LDA ,X+   Position
JSR BOX1  Barre Analogique
JSR NUME  Affichage numérique
LEAX 2,X  Pointe sur le Para
* metre suivant
CMPX FTAB Fin?
BNE ENCOR Non. Alors ENCOR
PULS A,X,Y,PC Oui.

```

### Organigramme



## DECA:

Le sous-programme DECA effectue la correction d'échelle en affectant la valeur d'un coefficient spécifique, pour chaque paramètre. DECA élabore le produit ( $2 * coef * val$ ) si le coefficient est positif et la division ( $val/2 * coef$ ) si le coefficient est négatif. Le coefficient est passé au sous-programme par le registre A tandis que le registre B passe la valeur. Au retour, A et B contiennent la valeur formatée. Les autres registres ne sont pas modifiés.

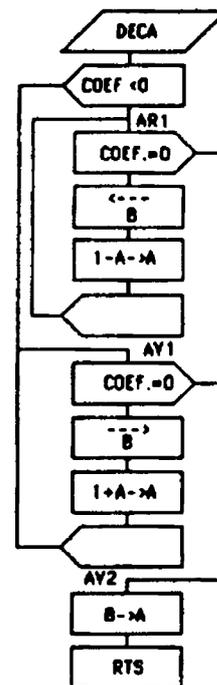
### Listing source

```

DECA  TSTA          Le coefficient est
* il negatif?
      BMI          AV1   Oui.
AR1   TSTA          Non.Alors est-il
* nul?
      BEQ          AV2   Oui.Alors il n'af-
* fecte Pas la valeur
      LSLB         On decale a gauche.
      DECA         Decremente le coef.
      BRA          AR1
AV1   TSTA          Coeff nul?
      BEQ          AV2   Oui.
      LSRB         Non.On decale a droite
      INCA         Incrmente le coef.
      BRA          AV1
AV2   TFR          B,A   Produit(2 coef*val)
* dans B
      RTS

```

### Organigramme



## BOX1:

BOX1 est la procédure qui détermine les paramètres d'entrée de BOX. Le sous-programme BOX est appelé deux fois par BOX1. Le premier passage dans BOX permet d'effacer le tracé précédent en écrivant, en noir sur fond noir, une barre d'élongation maximum. Le second passage trace la barre de couleur et de longueur définies par les paramètres d'entrée de BOX1. Le registre A passe la position en ordonnée de la barre à tracer. Le registre B passe l'amplitude du paramètre qui détermine la longueur de la barre. La couleur est spécifiée par le registre FORME.

La seconde partie du sous-programme détecte le tracé du P.I. car le point zéro de ce paramètre est spécifié au milieu de la plage. Ceci afin de

représenter les valeurs positives à droite et les valeurs négatives à gauche. Le point zéro du P.I. est positionné au milieu de l'échelle, soit un décalage de 24 pixels (ou points) par rapport au zéro des autres paramètres. Le registre TRAM est donc incrémenté de 24, quand il s'agit de tracer la barre du P.I. Au retour, TRAM est repositionné à sa valeur initiale.

### Registres et drapeaux utilisés ou modifiés

EFFA: Drapeau qui indique le passage dans le premier BOX pour annuler le décalage du P.I.

FORME-TRAM

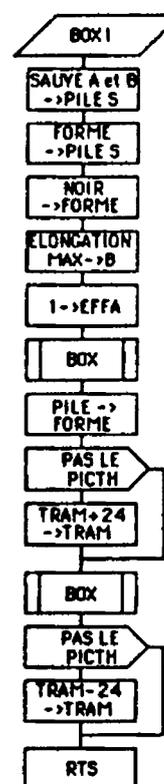
### Listing source

```

BOX1   PSHS   D
        LDB   FORME   Sauve le registre
* forme
        PSHS   B
        LDB   #-1     Ecriture noire
        STB   FORME
        LDB   #49T    Elongation maximum
        CLR   EFFA
        INC   EFFA    Efface le Pitch
        JSR   BOX     Efface le trace
* precedent
        PULS   B
        STB   FORME   Restitue forme
        PULS   D
        CMPA  #46T    Test si c'est le
* PictH
        BNE   AR1     Non
        LDY   TRAM    Oui. Alors..
        LEAY  24T, Y  on decale le ..
        STY   TRAM    ..debut
        CLR   EFF     Valide le Pitch
        JSR   BOX     Trace
        CMPA  #46T    Test si c'est le
* PictH
        BNE   *+13T   dans trame
        LDY   TRAM
        LEAY  -24T, Y Retablit la valeur
* de Y
        STY   TRAM
        RTS

```

### Organigramme



### BOX:

BOX permet de tracer des rectangles (barres) dont les caractéristiques sont définies à l'entrée du sous-programme. Le registre TRAM spécifie l'abscisse (axe des X) du tracé. Le premier point en ordonnée (axe des Y) est donné par le registre A et le registre B passe la longueur de la barre. La largeur des barres est constante et est initialisée à 4 points dans le programme. La couleur est passée par le registre FORME.

## Registres et drapeaux utilisés ou modifiés

CHDRAW: Registre qui détermine le mode de tracé (point ou caractère)

CHDRAW = 0 → mode point    CHDRAW ≠ 0 → mode caractère

PLOTX: Abscisse du dernier point allumé

PLOTY: Ordonnée du dernier point allumé

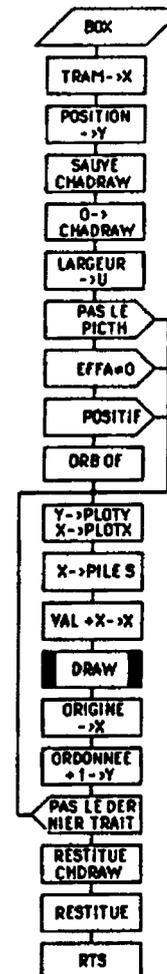
TRAM-EFFA

## Listing source

```

BOX    PSHS   X,Y,U,D
      LDX   TRAM   Abscisse dans X
      LDY   #0
      LEAY  A,Y    Ordonnee dans Y
      LDA   CHDRAW Sauve CHDRAW
      PSHS  A
      CLR  CHDRAW Mode graphique
      LDU  #4      Largeur
      EXG  U,Y
      CMPU #46T   PictH?
      BNE  MACH    Non.
      TST  EFFA   Oui.Alors est-11
* valide
      BNE  MACH    Non
      CMPB #510   Oui.Alors est-11
* negatif
      BLS  MACH    Non.
      ORB  #5F0   Oui.Alors on Prend
* le complement
MACH  EXG  U,Y
      STX  PLOTX  Abscisse dans PLOTX
      STY  PLOTY  Ordonnee dans PLOTY
      PSHS X      Sauve l'origine
      LEAX B,X    Calcul la fin de
* la barre
      JSR  DRAW   Un trait
      PULS X     Restitue l'origine
      LEAY 1,Y   Ordonnee du Prochain
* trait
      EXG  Y,U
      LEAY -1,Y  Decompte le nombre
* de traits
      BNE  MACH  Fin?
      PULS A    Oui.Restitue CHDRAW
      STA  CHDRAW
      PULS X,Y,U,D,PC
  
```

## Organigramme



## NUME:

Le sous-programme NUME permet l'affichage des paramètres sous forme numérique. La position d'affichage sur l'axe des Y est passée par le registre A du 6809. Le registre B contient la valeur binaire du paramètre. Le registre X pointe sur le tableau des chaînes de caractères ASCII spécifiant les valeurs que peut prendre le paramètre. Les couleurs de forme et fond sont définies par la chaîne de caractères AFIG. Au retour de NUME, les registres ne sont pas modifiés.

## Registres et drapeaux utilisés ou modifiés

### PODC-TRAM-BISTA-AFIG

#### Listing source

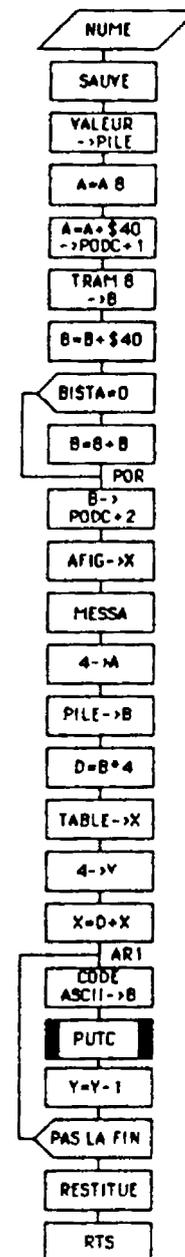
```

*AFFICHAGE NUMERIQUE
NUME  PSHS  X,Y,D,U
      PSHS  B,X    B,X au sommet de la
* Pile S
      LSRA          Divise Par 8 Pour
* trouver les coordonnees caracteres
      LSRA
      LSRA
      ADDA  #$40    Dans la chaine
      STA   PODC+1
      LDD   TRAM    Pour trouver les
* coordonnees caracteres
      LSRB
      LSRB
      LSRB
      ADDB  #$40
      TST  BISTA    Pour definir le
* lieu d'affichage
      BNE  POR
      ADDB  #$8      Plus 8 caracteres
* en X
      POR  STB  PODC+2  Dans la chaine
      LDX  #AFIG
      JSR  MESSA    Definie la couleur
* et la Position du curseur
      LDA  #4      4 octets Par chaine
* du tableau des valeurs numeriques
      PULS B,X    Recupere les valeurs
* au sommet de la Pile
      MUL          Calcul la valeur
* qu'il convient d'ajouter au debut du
* tableau Pour Pointer sur le Premier
* octet de la chaine

      LDX  D,X      Adresse
      LDY  #4      4 octets Par chaine
AR1    LDB  ,X+    Octet de la chaine
      JSR  PUTC    Affichage
      LEAY -1,Y    Decompte
      BNE  AR1    Fin?Alors AR1
      PULS X,Y,D,U,PC  Oui

```

#### Organigramme



## TRAMB:

TRAMB est le sous-programme d'initialisation des paramètres nécessaires à l'affichage du curseur B. TRAMB est symétrique à TRAMA et a la même structure logique.

## Registres et drapeaux utilisés ou modifiés

NUTRB: Chaîne de caractères pointée par NUTRB qui permet le positionnement et l'affichage du numéro de la trame B

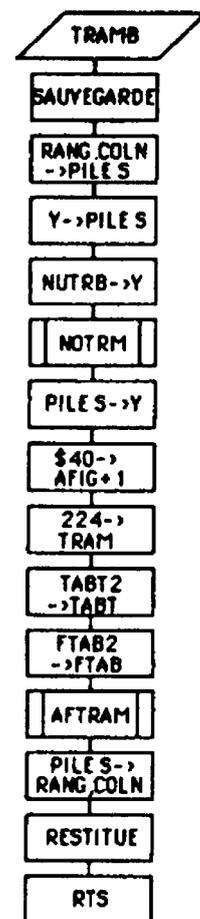
```
NUTRB    FCB        $1B,$41,$1B,$53,$1B,$4E
          FCB        $1F,$43,$64,$30,$31,$1B
          FCB        $4C,0
```

FTAB2: Fin de la table TABT2  
RANG-COLN-TABT2-AFIG

## Listing source

```
TRAMB    PSHS      D,X
          LDA      RANG      Sauvegarde la der-
* niere Position du curseur
          LDB      COLN
          PSHS     D
          PSHS     Y          Sauve l'etat de Y
          LDY     #NUTRB     Chaîne de car. qui
* Permet le Positionnement et l'afficha-
* -ge du numero de trame B
          JSR     NOTRM      No de trame
          PULS    Y          Restitue Y
          LDA     #$40      Ecriture noire
* Pour effacer le no Precedent.
          STA     AFIG+1
          LDX     #224T     Point zero de la
* trame B ....
          STX     TRAM      ..dans debut de
* trame
          LDX     #TABT2     Table TABT2....
          STX     TABT      ..dans le registre
* de table en cours de traitement
          LDX     #FTAB2     Fin de TABT2....
          STX     FTAB      ..dans le registre
* de fin de trame
          JSR     AFTRAM     Affichage du cur-
* seur B(ou trame B)
          PULS    D          Restitue la posi-
* tion du curseur
          STA     RANG
          STB     COLN
          PULS    D,X,PC
```

## Organigramme



## POSCUR:

POSCUR permet de repositionner le curseur sur la ligne d'édition des caractères correspondants aux phonèmes. POSCUR redéfinit également les couleurs précédentes de fond et de forme.

## Registres et drapeaux utilisés ou modifiés

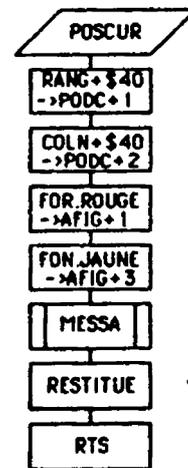
### RANG-COLN-PODC-AFIG

#### Listing source

```

POSCUR PSHS   D,X,CC  Sauvegarde
        LDA   RANG   Acquisition des
* dernieres coordonnees stokees dans
* RANG et COLN...
        LDB   COLN   ..et transfert
* dans la chaine de Positionnement PODC
COLON   ADDA   #$40
        ADDB  #$40
        STA   PODC+1
        STB   PODC+2
        LDA   #$41   Forme rouge
        STA   AFIG+1
        LDA   #$53   Fond Jaune
        STA   AFIG+3
        LDX   #AFIG
        JSR   MESSA  Envoie des chaines
* AFIG et PODC
        PULS  D,X,CC,PC
    
```

#### Organigramme

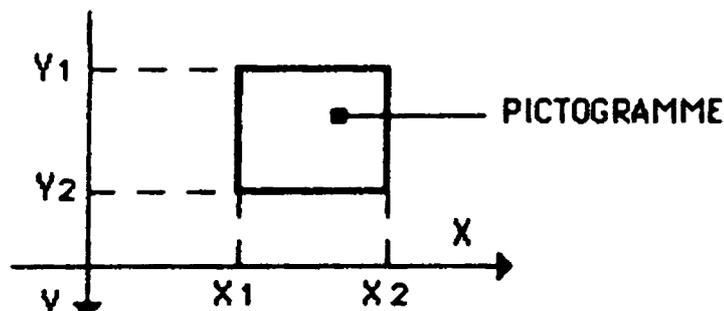


## PIC:

Le tronçon de programme PIC permet d'aiguiller le programme principal vers les modules de traitement spécifiques aux commandes utilisant le crayon optique. A l'entrée du module PIC, les registres X et Y contiennent les coordonnées pointées par le crayon optique à l'instant de la fermeture du contact situé à la pointe du crayon. Ces coordonnées sont comparées aux valeurs consignées dans une table étiquetée PICTO dont le format est le suivant:

```
PICTO      FDB      X1,X2,Y1,Y2,ADR
```

X1, X2, Y1, Y2 sont des valeurs binaires sur 16 bits (2 octets) qui représentent les limites du pictogramme. ADR est l'adresse de traitement de la commande.



Si le point, dont les valeurs en X et Y sont passées par les registres équivalents, n'est pas inscrit dans un pictogramme de la table PICTO, le programme branche à l'étiquette ART et attend une autre commande à partir du crayon optique ou à partir du clavier.

### Registres et drapeaux utilisés ou modifiés

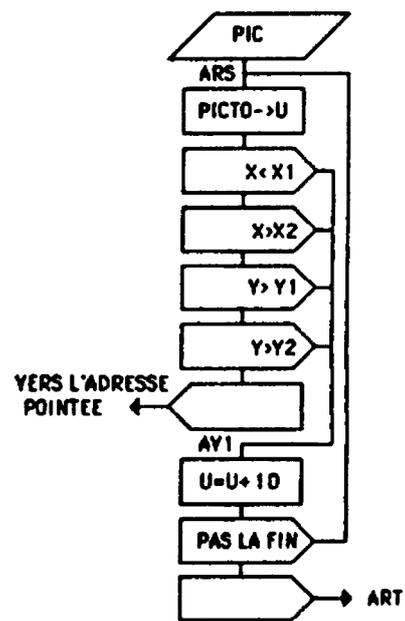
PICTO: Début de la table donnant les limites des pictogrammes suivies de l'adresse de traitement de la commande ou de la fonction associées  
 FPICTO: Fin de la table PICTO

### Listing source

```

PIC   LDU   #PICTO  Debut de la table
* de Pointage dans le registre U
ARS   CMPX  ,U Plus Petit que l'abs-
* cisse gauche?
      BMI   AV1
      CMPX  2,U
      BPL   AV1 Plus grand que l'abs-
* cisse droite?
      CMPY  4,U Plus Petit que l'ordon
* nee haute ?
      BMI   AV1
      CMPY  6,U Plus grand que l'ordon
* nee basse ?
      BPL   AV1
      JMP   [8,U] Branchement vers
* le Programme de traitement du Picto-
* gramme trouve
AV1   LEAU  10T,U Pointe sur les co-
* ordonnees du Pictogramme suivant
      CMPU  #FPICTO Fin de la table
* PICTO ?
      BNE   ARS Non. Alors on compare
* les valeurs contenues dans X et Y avec
* les coordonnees du Pictogramme suivant
      BRA   ART Oui. ON retourne a ART.
* Les coordonnees Pointees Par le crayon
* optique ne correspondent Pas a un Pic-
* togramme de la table PICTO
  
```

### Organigramme



## INCRA:

INCRA est la partie du programme principal aiguillée par PIC qui permet l'incréméntation du curseur A. L'état courant du curseur A est passé par le pointeur de trame A (PTA) qui pointe sur le premier octet de la trame précédente. Pour positionner le curseur sur la trame suivante, le pointeur doit être incrémenté de 4, car il y a 4 octets par trame.

INCRA permet d'incrémenter le curseur A dans les deux types de représentation sur l'écran (graphique et numérique). Le curseur A ne peut dépasser le curseur B. Si le curseur A rejoint le curseur B, les hauteurs "avant et après" dans TABT1 sont recopiées dans TABT2 afin que les deux curseurs soient identiques.

## Registres et drapeaux utilisés ou modifiés

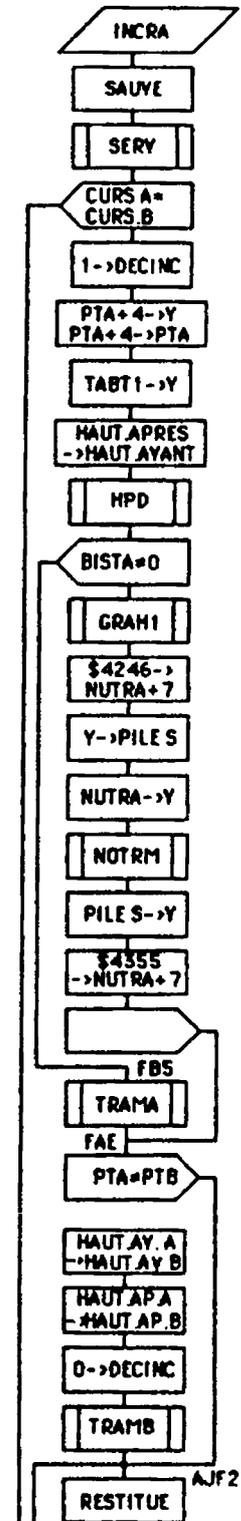
### NUTRA-NUTRB-DECINC-PTA-TABT1-BISTA-TABT2

#### Listing source

```

INCRA PSHS X,Y,D Sauvegarde
      JSR SERV Pour annuler les
* etats Precedents et remettre le tour
* en bleu clair
      LDD NUTRA+9 Compare le N0 du
* curseur A avec le N0 du curseur B
      CMPD NUTRB+9
      LBHS AZE Si Plus grand ou
* egal Pas d'incrementation
      CLR DECINC Positionne le
* drapeau en incrementation (DECINC=1)
      INC DECINC
      LDX PTA Charge le Pointeur
* de trame A dans X
      LEAX 4,X Incrmente de 4
* Pour pointer sur la trame suivante
      STX PTA Sauve le Pointeur
* Pour la Prochaine fois
      LDY #TABT1 Debut deTABT1 dans
* le registre Y
      LDA 68,Y Pour transferer
* la "hauteur apres" la trame Precedente
* comme "hauteur avant" la trame suivante
      STA 2,Y
      JSR HPD Affecte les Para-
* metres dans la table TABT1
      TST BISTA Quel est le type
* d'ecran en cours
* BISTA=0--> Ecran "numerique"
* BISTA#0--> Ecran "Graphique"
      BEQ FB5 Numerique
      JSR GRAH1 Ecran Graphique
      LDD #$4246 Position d'affich-
* age du numero de trame
      STD NUTRA+7
      PSHS Y Sauve le Pointeur Y
      LDY #NUTRA Pour modifier le
* numero de la trame
      JSR N0TRM N0 de trame
      PULS Y Restitue le Poin-
* teur Y
      LDD #$4355 Pour restituer la
* position d'affichage en mode non Graph
* ique
      STD NUTRA+7
      BRA FAE
FB5 JSR TRAMA Curseur A
FAE LDD NUTRA+9 Si PTA=PTB copi
* e des hauteurs
      CMPD NUTRB+9
      BNE AJF2 Non egale
      LDA 2,Y Transfert des "hau-
* teur avant" de la trame ou curseur A
* vers le curseur B
  
```

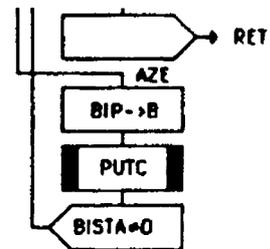
#### Organigramme



```

        STA     TABT2+2
        LDA     68T,Y   Transfert des "haut-
* teur apres" du curseur A vers le cur-
* seur B
        STA     TABT2+68T
        CLR     DECINC  Clear le DECINC
* Pour ne pas incrementer le curseur B
* dans NQTRM
*
        JSR     TRAMB  Affiche le curseur B
        AJF2    PULS   X,Y,D
        LBRA    RET    Branche en RET
        AZE     LDB    #07   Bip
                JSR    PUTC
                BRA    AJF2

```



## SERV:

SERV est un sous-programme de réinitialisation du drapeau BLOQ à 1 afin d'annuler une éventuelle commande de variation de paramètres. Le tour de l'écran est remis en couleur bleu clair ce qui signifie: "commande de variation de paramètres non active".

### Registres et drapeaux utilisés ou modifiés

BLOQ: Drapeau de validation des commandes de variation de paramètres

BLOQ = 0    variation active            BLOQ = 1    variation inactive

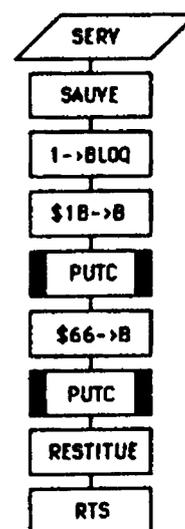
### Listing source

```

SERV   PSHS    B,X
        CLR    BLOQ   Annule la varia-
* tion de Parametre
        INC    BLOQ
        LDB    #$1B   Pour mettre le
* tour bleu clair
        JSR    PUTC
        LDB    #$66
        JSR    PUTC
        PULS   B,X,PC

```

### Organigramme



## INCRB:

INCRB est la section de programme qui permet d'incrémenter le curseur B. L'état courant du curseur est passé par le pointeur PTB. Avant d'incrémenter le curseur B, le pointeur (PTB), incrémenté de 4, est comparé au pointeur de fin de MOTX (POITX). Si PTB est supérieur ou égal à POITX, le curseur B n'est pas incrémenté.

### Registres et drapeaux utilisés ou modifiés

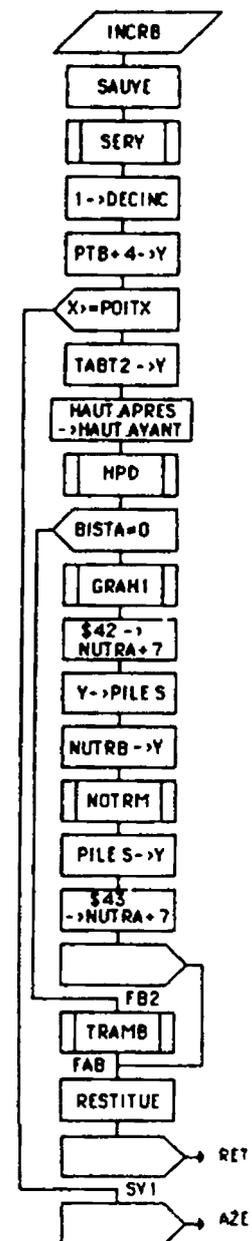
### DECINC-PTB-POITX-TABT2-BISTA-NUTRB

#### Listing source

```

INCRB PSHS X,Y,D Sauvegarde
      JSR  SERV Pour annuler les
* etats Precedents et remettre le tour
* en bleu clair
      CLR  DECINC Positionne le
* drapeau en incrementation (DECINC=1)
      INC  DECINC
      LDX  PTB Charge le Pointeur
* de trame B dans X
      LEAX 4,X Incrmente de 4
* Pour pointer sur la trame suivante
      CMPX POITX Pour ne Pas dePas-
* ser la limite la derniere trame
      BGE  SV1 DePasse!
      STX  PTB Sauve le Pointeur
* Pour la Prochaine fois
      LDY  #TABT2 Debut deTABT2 dans
* le registre Y
      LDA  68,Y Pour transferer
* la "hauteur apres" la trame Precedente
* comme "hauteur avant" la trame suivante
      STA  2,Y
      JSR  HPD Affecte les Para-
* metres dans la table TABT1
      TST  BISTA Quel est le type
* d'ecran en cours
*      BISTA=0--> Ecran "numerique"
*      BISTA#0--> Ecran "graphique"
      BEQ  FB2 Numerique
      JSR  GRAH1 Ecran graphique
      LDA  #$42 Position d'affich-
* age du numero de trame
      STA  NUTRB+7
      PSHS Y Sauve le Pointeur Y
      LDY  #NUTRB Pour modifier le
* numero de la trame
      JSR  N0TRM N0 de trame
      PULS Y Restitue le Poin-
* teur Y
      LDA  #$43 Pour restituer la
* position d'affichage en mode non graph
* ique
      STA  NUTRB+7
      BRA  FAB
FB2 JSR  TRAMB Curseur B
FAB PULS X,Y,D
     LBRA RET Branche en RET
SV1 LBRA AZE
  
```

#### Organigramme



## DECRA:

DECRA est le tronçon du programme principal qui permet de positionner le curseur A sur la trame précédente. Le pointeur PTA pointe sur le premier octet de la trame. DECRA est actif avec les deux modes de représentation de l'écran (graphique et numérique). Si le curseur A est positionné sur la trame 1, la "hauteur avant" de la trame est recopiée dans l'octet de MOTX représentant la hauteur initiale du mot ou de l'expression.

### Registres et drapeaux utilisés ou modifiés

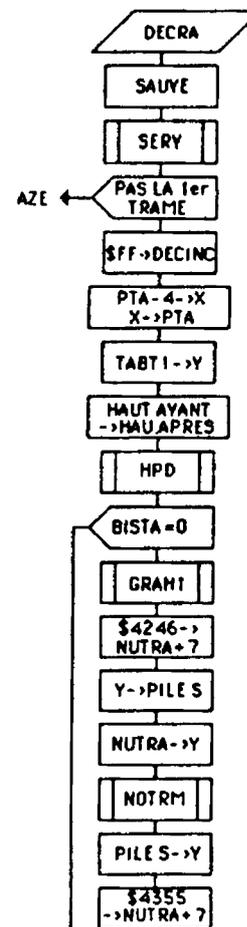
#### NUTRA-DECINC-PTA-BISTA-MOTX

#### Listing source

```

DECRA PSHS X,Y,D Sauvegarde
      JSR  SERV  Pour annuler les
* etats precedents et remettre le tour
* en bleu clair
      LDD  NUTRA+9 Est-ce la trame 1?
      CMPD ##3031
      LBL  AZE Si Plus Petit ou
* egal Pas de decrementation
      CLR  DECINC Valide le drapeau
* en decrementation ($FF)
      DEC  DECINC
      LDX  PTA Charge le Pointeur
* de trame A dans X
      LEAX -4,X Pointe sur la trame
* precedente
      STX  PTA Sauve le Pointeur
      LDY  #TABT1 Debut de la table
* TABT1 dans Y
      LDA  2,Y Pour transferer
* la hauteur avant la trame suivante com
* me hauteur apres la trame precedente
      STA  68,Y
      JSR  HPD Transcrit les Par
* ametres dans la table TABT1
      TST  BISTA Quel est l'ecran
* en cours?
* BISTA=0--> Ecran "numerique"
* BISTA=1--> Ecran "graphique"
      BEQ  FB3 Numerique
      CLR  BISTA Inverse le BISTA
* avant le passage dans GRAH1
      JSR  GRAH1 Ecran graphique
      LDD  #$4246 Position d'affi-
* chage du numero de trame
      STA  NUTRA+7
      STB  NUTRA+8
      PSHS Y Sauve Y
      LDY  #NUTRA Pour modifier le
* numero de la trame
      JSR  NOTRM NO de trame
      PULS Y Restitue Y
      LDD  #$4355 Restitue la Posi-
* tion d'affichage
  
```

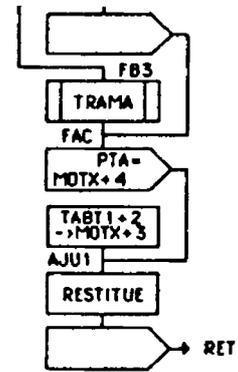
#### Organigramme



```

        STA     NUTRA+7
        STB     NUTRA+8
        BRA     FAC
FB3     JSR     TRAMA   Curseur A
FAC     CMPX   #MOTX+4 Debut du mot?
        BNE    AJU1    Non.
        LDA    2,Y     Oui. Alors il faut
* copier la hauteur de la table TABT1
* dans l'octet de hauteur initiale du
* mot
        STA     MOTX+3
AJU1    PULS   X,Y,D   Restitue
        LBRA   RET

```



## DECRB:

DECRB permet au curseur B de pointer sur la trame précédente. L'état courant du curseur est passé par le pointeur PTB. Tout comme DECRA, DECRB recopie la "hauteur avant" de la trame dans l'octet de hauteur initiale du mot à prononcer.

### Registres et drapeaux utilisés ou modifiés

NUTRB-NUTRA-DECINC-PTB-TABT2-BISTA-MOTX

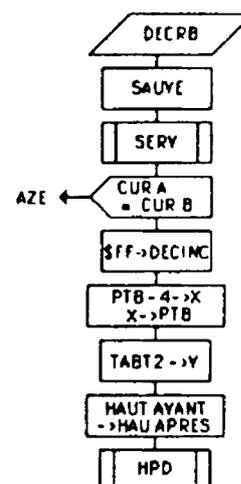
### Listing source

```

DECFB  FSHS   X,Y,D   Sauvegarde
        JSR   SERV   Pour annuler les
* etats Precedents et remettre le tour
* en bleu clair
        LDD   NUTRB+9 Compare le numero
* de la trame B (curseur B) avec la
* trame A (curseur A)
        CMPD  NUTRA+9
        LELS  AZE     Si Plus Petit ou
* egal Pas de decrementation
        CLP  DECINC  FF on decremente
        DEC  DECINC
        LDX  PTB     Pointeur de trame B
* dans le registre X
        LEAX -4,X   Pointe sur la trame
* Precedente
        STX  PTB     Sauve le Pointeur
* Pour la Prochaine fois
        LDY  #TABT2 Debut de la table
* TABT2 dans le registre Y
        LDA  2,Y     Pour transferer
* la hauteur avant la trame suivante com
* me hauteur apres la trame Precedente
        STA  68,Y
        JSR  HPD     Affecte les Para-
* metres de la trame dans la table TABT2
        TST  BISTA  Quel est le mode

```

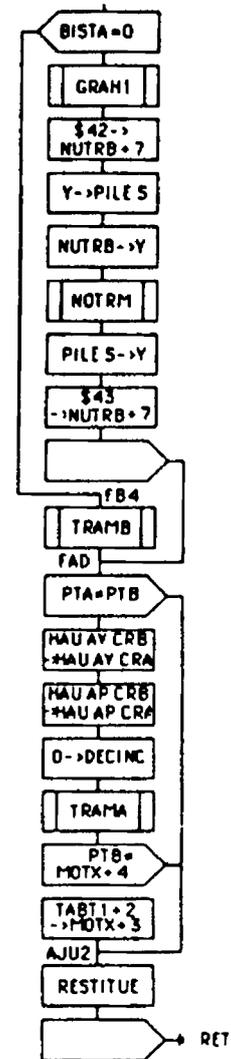
### Organigramme



```

* d'affichage en cours? "Numerique ou
* "Graphique"?
*      0--> BISTA --> "numerique"
*      1--> BISTA --> "Graphique"
      BEQ   FB4   Numerique
      CLR   BISTA  Graphique.Comme
* BISTA est modifie dans GRAH1, il faut
* l'inverser avant, pour ne pas changer
* le mode d'affichage
      JSR   GRAH1  Ecran Graphique
      LDA   #$42  Position d'affi-
* se du numero de la trame
      STA   NUTRB+7
      PSHS  Y      Sauve le Pointeur Y
      LDY   #NUTRB Pour modifier le
* numero de la trame
      JSR   N0TRM  N0 de trame
      PULS  Y      Restitue le Poin-
* teur
      LDA   #$43  Pour restituer
* la position d'affichage Precedente
      STA   NUTRB+7
      BRA   FAD
FB4   JSR   TRAMB  Curseur B
FAD   LDD   NUTRB+9  Si PTB=PTA copie
* des hauteurs
      CMPD  NUTRA+9
      BNE  AJU2
      LDA  Z,Y  "Hauteur avant" du
* curseur A dans la "Hauteur avant" du
* curseur B
      STA  TABT1+2
      LDA  68T,Y  "Hauteur apres"
      STA  TABT1+68T
      CLR  DECINC  0->DECINC Pour ne
* pas modifier le no de la trame
      JSR  TRAMA  Affiche le curseur A
      CMPX #MOTX+4  Debut de l'exPres-
* sion ?
      BNE  AJU2  Non. Alors fin
      LDA  Z,Y  Oui. Alors la hau-
* teur initiale du curseur B est egale a
* la hauteur initiale de l'expression
      STA  MOTX+3
AJU2  PULS  X,Y,D  Restitue
      LBRA RET  Branche a RET

```



## PARAP-PARAM:

PARAP-PARAM permettent de positionner les drapeaux BLOQ et MPAR. BLOQ est positionné à 0 pour valider les commandes spécifiques à chaque paramètres. MPAR est positionné à 0 pour une variation positive du paramètre et à 1 pour une variation négative. Le passage dans l'une des deux parties du programme est matérialisé par le tour de l'écran.

ROUGE → Variation positive

BLEU → Variation négative

## Registres et drapeaux utilisés ou modifiés

MPAR: Drapeau qui indique le sens de la modification des paramètres

MPAR = 0 → augmentation

MPAR = 1 → diminution

BLOQ: Est le drapeau qui permet de valider ou d'invalider les pictogrammes de modification des paramètres du curseur A ou B

BLOQ = 0 → commandes valides

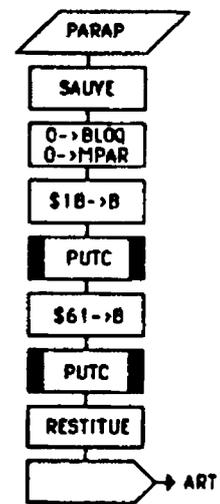
BLOQ = 1 → commandes invalides

### Listing source

```

PARAP  PSHS  B      Sauvegarde
        CLR   BLOQ  Bloque les autres
* entree
        CLR   MPAR  0--> Plus
        LDB  ##1B  Tour rouge
        JSR  PUTC
        LDB  ##61
        JSR  PUTC
        PULS B      Restitue
        LBRA ART    Branche a ART
    
```

### Organigramme

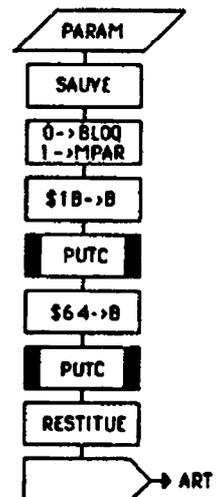


### Listing source

```

PARAM  PSHS  B      Sauvegarde
        CLR   BLOQ  Bloque les autres
* entrees
        CLR   MPAR  1--> moins
        INC  MPAR
        LDB  ##1B  Tour bleu
        JSR  PUTC
        LDB  ##64
        JSR  PUTC
        PULS B      Restitue
        LBRA ART    Branche a ART
    
```

### Organigramme



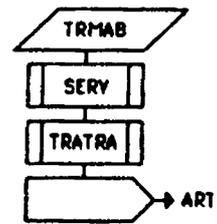
## TRAMAB:

TRAMAB est la partie du programme principal qui permet de prononcer le tronçon de mot ou d'expression compris entre le curseur A et le curseur B. A l'issue de TRAMAB, le programme branche à l'étiquette ART afin d'attendre une autre commande.

### Listing source

```
TRMAB JSR     SERV     Pour annuler les
* etats precedents et remettre le tour
* en bleu clair
      JSR     TRATRA   Prononce de la
* trame A a la trame B
      LBR    ART     BRAnche a ART
```

### Organigramme



## TRATRA:

TRATRA est le sous-programme, appelé par TRAMAB, qui permet la prononciation du tronçon de mot compris entre la trame A et la trame B.

Les pointeurs PTA et PTB indiquent la position de chaque curseurs (trames A et B) sur le mot ou l'expression. PTB doit être incrémenté d'une trame, avant de spécifier la limite (LIMIT) de la partie du mot à prononcer, car PTB pointe sur le premier octet de la dernière trame.

La hauteur initiale de cette partie de mot est prélevée dans la table TABT1 (TABT1 + 2) qui contient la "hauteur avant" de la trame en cours et calculée à chaque mouvement du curseur A. La temporisation en fin de sous-programme est nécessaire pour laisser le temps à l'utilisateur de retirer le crayon optique du pictogramme, afin d'éviter une répétition.

### Registres et drapeaux utilisés ou modifiés

LIMIT: Registre 16 bits qui contient la limite base de la partie du mot ou de l'expression à prononcer

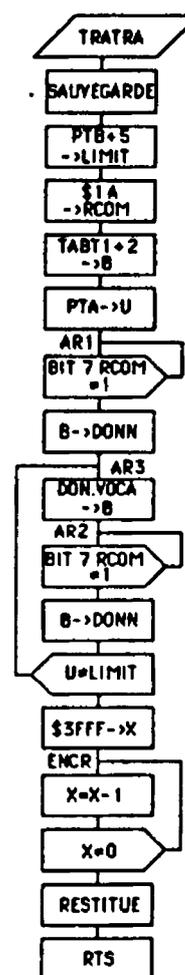
RCOM-TABT1-DONN-PTA-PTB

## Listing source

```

TRATRA PSHS D,U,X Sauvegarde
        LDU PTB Pour calculer la
* Position de la dernière trame à Pron-
* noncer et.....
        LEAU 5,U
        STU LIMIT ..Stocke le resul-
* tat dans LIMIT
        LDB #1A Positionne le MEA
* 8000 en ARRÊT LENT et IRQ invalide
        STB RCOM
        LDB TABT1+2 Hauteur initiale,
* a l'entrée de la trame A Prelevée dans
* la table TABT1 (a TABT1+2)
        LDU PTA Le registre U Poin-
* te sur le premier octet de la trame A
AR1 TST RCOM Le MEA reclame-t-il
* des données?
        BPL AR1 Non, Alors on boucle
* en AR1
        STB DONN Oui, On envoie l'oc-
* tet contenu dans B
AR3 LDB ,U+ On charge un nouvel
* octet
AR2 TST RCOM Encore des données?
        BPL AR2 Non, Alors on boucle
        STB DONN Oui.
        CMPI LIMIT Le dernier octet
        BNE AR3 Non, Alors on conti-
* nue
        LDX #3FFF Tempo Pour éviter
* la redite du mot
ENCR LEAX -1,X
        BNE ENCR
        PULS D,U,X,PC Restitue
    
```

## Organigramme



## GRAH:

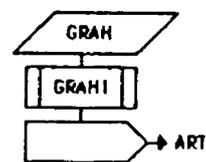
GRAH permet d'appeler le sous-programme GRAH1. Au retour de GRAH1, GRAH branche le programme principal à l'étiquette ART.

## Listing source

```

GRAH JSR GRAH1 Sous Programme GRAH1
        LBR4 ART
    
```

## Organigramme



## GRAH1:

GRAH1 permet de passer d'un affichage dit "numérique" à un affichage dit "graphique" et vice versa. Le type de l'affichage en cours est spécifié par la valeur du drapeau BISTA.

**BISTA = 0** Ecran "numérique"

**BISTA = 1** Ecran "graphique"

Ainsi, si BISTA est à 0 à l'entrée de GRAH1, il convient de passer l'écran en mode graphique. Si BISTA vaut 1, il faut passer en mode numérique. Ce qui revient à compléter BISTA dans tous les cas. Si GRAH1 est sollicité, alors qu'aucun phonème n'a été stocké dans la zone mémoire "MOTX", on sort immédiatement de GRAH1.

Afin d'utiliser au mieux la surface de l'écran, la hauteur initiale est spécifiée en haut et en bas de la fenêtre graphique. Ainsi les valeurs inférieures à la hauteur initiale sont représentées par rapport au haut de l'écran, tandis que les valeurs supérieures sont référencées au bas de l'écran. On obtient ainsi la représentation d'une dynamique double avec la même surface d'écran. Soit une dynamique, crête à crête de 240 Hz.

Chaque trame est visualisée par des barres de couleurs différentes à l'exception des trames générant un bruit blanc qui sont représentées en blanc. La fenêtre de l'écran peut visualiser, au maximum, 35 trames successives situées indifféremment sur l'expression. Le début de chaque phonème est matérialisé par l'affichage, sur l'axe des X, du caractère représentant le phonème. On peut ainsi repérer la trame que l'on souhaite modifier.

### **Registres et drapeaux utilisés ou modifiés**

**POTC1:** Chaîne de caractères qui est pointée par POTC1 et permet le passage d'attributs de couleur et de positionnement.

<b>PODC1</b>	<b>FCB</b>	<b>\$1B,\$44,\$1B,\$53</b>
	<b>FCB</b>	<b>\$1F,\$40,\$40,00</b>

**MSG1:** Chaîne de caractères, pointée par MSG1, qui spécifie une séquence de définition, d'effacement de fenêtre et de positionnement du curseur.

<b>MSG1</b>	<b>FCB</b>	<b>\$1B,\$50,\$1F,\$11,\$18,\$1F</b>
	<b>FCB</b>	<b>\$20,\$22,\$0C,\$1F,\$31,\$38,00</b>

**HAUC:** Registre qui contient la "hauteur avant" de la trame pointée par le curseur A et qui est utilisé comme référence dans l'affichage graphique. Ce registre est aussi appelé "hauteur courante".

**CORX:** Abscisse de l'affichage graphique

**SIGN:** Signe de la représentation graphique

SIGN = 0 → positif (référence en bas de l'écran)  
 SIGN = 1 → négatif (référence en haut de l'écran)

POTLEC: Pointeur de lecteur sur la table de positionnement TABPOS  
 USERAF: Registre 16 bits qui pointe sur la table des caractères utilisateur

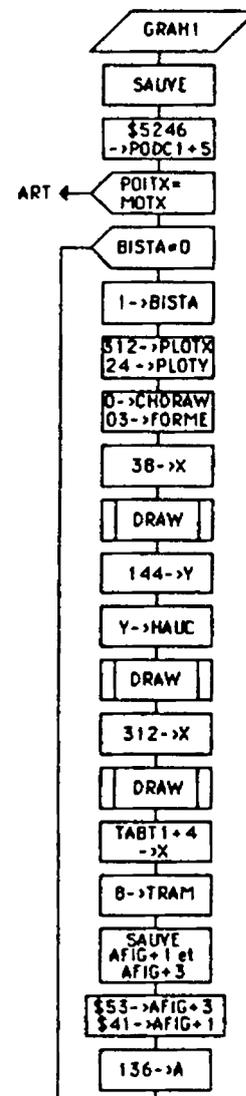
**MOTX-POITX-BISTA-PLOTX-PLOTY-CHDRAW-FORME-TRAM-AFIG-PTA-DECINC**

**Listing source**

```

GRAH1 PSHS X,Y,D,U Sauvegarde
      LDB #52 Pour positionner
* le curseur au debut de la ligne
      STB PDC1+5
      LDB #546
      STB PDC1+6
      LDX #MOTX A-t-on deja rentre
* un Phoneme dans MOTX?
      CMPX POITX
      LBEQ ART Non.Alors on sort
      LDX #MSG1 Oui.Alors on efface
* l'ecran entre la ligne 02 et 18
      JSR MESSA
      TST BISTA Mode numerique?
      LBNE ENUM Oui.
      CLR BISTA Non.BISTA a 1
      INC BISTA
      LDX #312T 1er Point du cadre
      STX PLOTX
      LDY #24T
      STY PLOTY
      CLR CHDRAW Mode graphique
      LDA #03
      STA FORME Couleur Jaune
      LDX #38T Second Point.Y est
* inchange
      JSR DRAW 1 trait
      LDY #144T 3em Point.X est
* inchange
      STY HAUC Point zero du graphe
      JSR DRAW 2em trait
      LDX #312T 4em Point.Y inchange
      JSR DRAW 3em trait
      LDX #TABT1+4 Debut des valeurs
* numeriques de hauteur
      LDD #8
      STD TRAM Change TRAM en
* donnees graphiques(8=1point caractere)
      LDA AFIG+3 Sauve le contenu
* de AFIG+3
      PSHS A
      LDA #53 Fond jaune
      STA AFIG+3
      LDA AFIG+1 Sauve AFIG+1
      PSHS A
      LDA #541 Forme rouge
      STA AFIG+1
      LDA #136T Ordonnee d'affichage
  
```

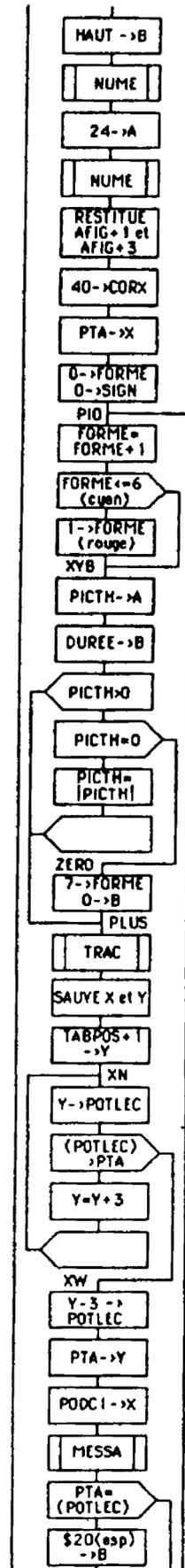
**Organigramme**



```

LDB -2,X Hauteur avant la
* trame Pointee Par le curseur A
JSR NUME Affichage
LDA #24T Ordonnee d'affichage
JSR NUME Affichage
PULS A Restitue les valeurs
* de AFIG+1 et AFIG+3
STA AFIG+1
PULS A
STA AFIG+3
LDX #40T
STX CORX 1er Point du Graph
* (5 caracteres)
LDX PTA Pointeur A -> X
CLR FORME Noir
CLR SIGN Positif
PIO INC FORME Pour modifier les
* couleurs a chaque trame
LDB #6
CMPB FORME Cyan?
BPL XVB Non
LDB #01 Oui. Alors on force
* en rouge
STB FORME
XVB LDA 3,X 3em octet de la
* trame dans A....
TFR A,B ... et dans B
ANDA #51F Masque sur A
* Pour conserver que le Pitch dans A
ROLB 4 decalages Pour
* cadrer la duree a droite
ROLB
ROLB
ROLB
* masque sur B
ANDB #03 Duree dans B
CMPA #15T Pitch Positif?
BLS PLUS Oui. Alors PLUS
CMPA #16T Pitch nul?
BEQ ZERO Oui. Alors ZERO
NEGA Donc negatif
ANDA #50F Pour Prendre la
* valeur absolue
NEGA
BRA PLUS
ZERO PSHS B Force en blanc la
* trame "bruit blanc"
LDB #07 Blanc-> FORME
STB FORME
PULS B
CLRA Duree nulle
PLUS JSR TRAC Trace la trame
PSHS X,Y Sauve X et Y
LDY #TABPOS+1 Pointe sur la
* premiere adresse de la table de Posi-
* tionnement et.....
XN STY POTLEC ..charge POTLEC
* qui est le pointeur de lecteur
CMPX [POTLEC] Compare PTH (X)
* avec l'adresse contenue dans POTLEC
BMI XW A-t-on depasse?
LEAY 3,Y Non. Alors on Passe
* a l'adresse suivante de la table TABPOS
BRA XN On continue
XW LEAY -3,Y On a depasse. Alors
* on Pointe sur l'adresse Precedente ..

```

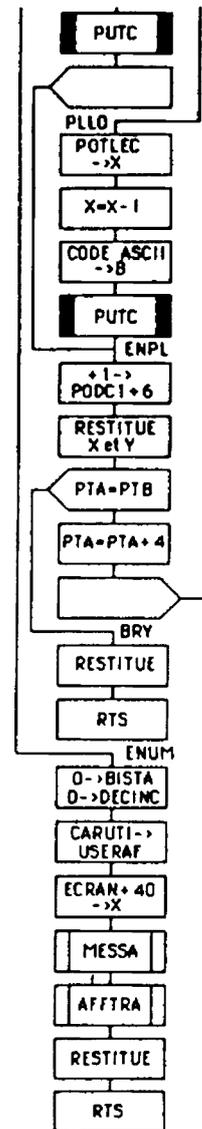


```

        STY    POTLEC .. que l'on charge
* dans POTLEC
  TFR    X,Y    Sauvegarde de X
  LDX    #PODC1 Positionne le cur-
* seur et les couleurs
  JSR    MESSA
  CMPY   [POTLEC] Compare PTA avec
* l'adresse contenue dans POTLEC
  BEQ    PLLO   Egale, alors on doit
* imprimer le caractere du Phoneme
  LDB    #B20   Inegale, alors on
* imprime un espace
  JSR    PUTC
  BRA    ENPL
PLLO   LDX    POTLEC Adresse du caractere
* tere dans X ...
  LEAX   -1,X
  LDB    ,X     Code ASCII->B
  JSR    PUTC
ENPL   INC    PODC1+6 Incrmente la
* Position du curseur
  PULS   X,Y   Restitue X et Y
  CMPX   PTB   Fin?

        LBEQ   BRY   Oui.
        LEAX   4,X   Non. on Pointe sur
* la trame suivante
  LBRA   PIO
BRY    PULS   Y,X,D,U,PC
*ECRAN"NUMERIQUE"
ENUM   CLR    BISTA  0->BISTA
        CLR    DECINC 0->DECINC
        LDX    #CARUTI Pour afficher la
* Partie fixe de l'ecran
  STY    USERAF
  LDX    #ECRAN+40T
  JSR    MESSA
  JSR    AFFTRA Affiche les trames
  PULS   Y,X,D,U,PC

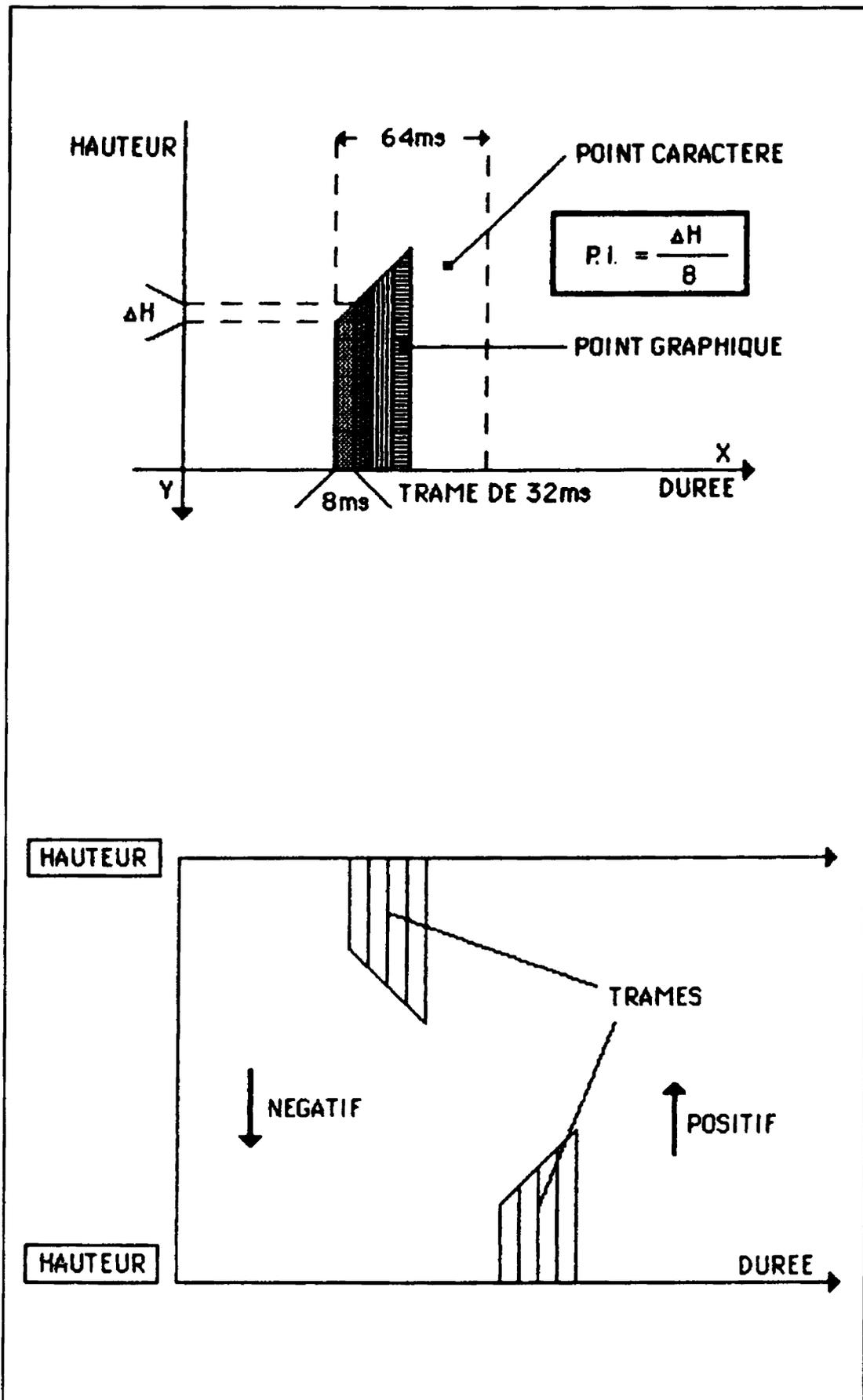
```



## TRAC:

TRAC est le sous-programme qui permet de dessiner les trames en mode "écran graphique". La durée de la trame est représentée sur l'axe des X tandis que la hauteur est exprimée sur l'axe des Y. Une trame peut prendre une durée de 8, 16, 32 ou 64 ms, soit un rapport de 8 entre la durée minimum et la durée maximum, ce qui correspond au nombre de points graphiques contenus dans un point caractère. A chaque colonne caractère est affectée une trame qui remplit plus ou moins la largeur de la colonne en fonction de la durée de la trame.

Le P.I. est le paramètre qui modifie la hauteur et s'exprime en Hz/8 ms. Ainsi, à chaque pas de 8 ms, correspondant à un point graphique, la hauteur sera modifiée d'une quantité (positive ou négative) égale à la valeur du P.I.



## Registres et drapeaux utilisés ou modifiés

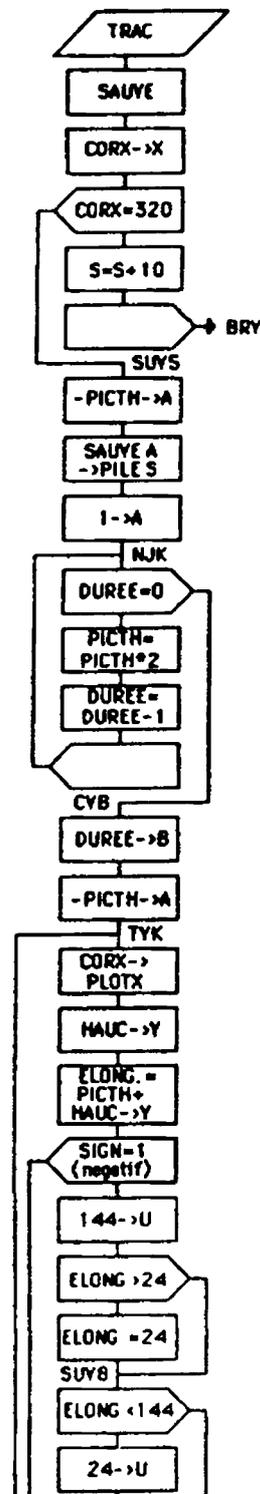
### CORX-PLOTX-HAUC-SIGN-PLOTY

#### Listing source

```

TPAC  PSHS  A,B,X,U,Y Sauvegarde
      LDX  CORX  Position courante
* sur l'axe des X
      CMPX #320T  Superieur a la
* limite de la fenetre
      BNE  SUV5  Non
      LEAS 10T,S  Depasse les limite
* de l'ecran Affichage impossible.On re-
* Positionne le Pointeur de Pile S et on
* branche en BRY
      LBRA  BRY
SUV5  NEGA  A      Moins Pitch->A
      PSHS  A      Sauve
      LDA  #1      1-> A
NJK   TSTB  A      Duree egale 0?
      BEQ  CVB     Oui.Alors fin
      LSLA  A      Non.Alors multiPlie
* A Par deux
      DECB  A      Duree moins 1
      BRA  NJK
*La duree(dans A) vaut desormais 1,2,4
* ou 8 Pour 8,16,32 ou 64 ms
CVB   TFR  A,B     Duree dans B
      PULS  A      -Pitch dans A
TYK   STX  PLOTX  Derniere Position
      LDY  HAUC   Hauteur courante
      LEAY A,Y    Calcul l'elongation
      TST  SIGN   Positif?
      BNE  SUV3   NON.
      LDU  #144T  Oui
      CMPY #24T   Limite max
      BHI  SUV8   Superieur?
      LDY  #24T   Non.Alors max 24T
SUV8  CMPY #144T  Limite min
      BLO  SUV2   Inferieur?
      LDU  #24T   non.
    
```

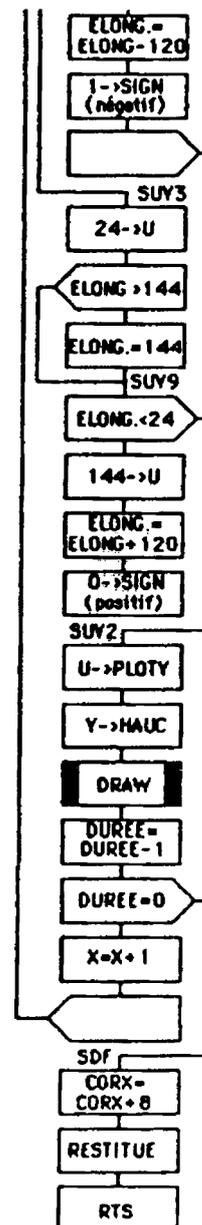
#### Organigramme



```

LEAY -120T,Y Ajoute un decalage
* Pour modifier le zero
INC SIGN 1-> SIGN=negatif
BRA SUV2
SUV3 LDU #24T Point zero en nega
* tif
CMPY #144T Limite MAX
ELO SUV9 Inferieur?
LDY #144T Non.Alors max 144T
SUV9 CMPY #24T Limite min
BHI SUV2 Superieur?
LDU #144T Non.
LEAY 120T,Y Alors on ajoute un
* decalage pour modifier le zero
CLR SIGN 0-> SIGN=Positif
SUV2 STU PLOTY Point de depart du
* trait
STY HAUC Sauvegarde la hau-
* teur courante
JSR DRAW Un trait
DECB DecompPte le nombre
# de traits
TSTB Fin?
BEQ SDF Oui.
LEAX 1,X Non.Nouvel abscisse
BRA TYK
SDF LDX CORX Positionne CORX
* Pour la Prochaine trame
LEAX 8,X Plus 8
STX CORX
PULS A,B,X,Y,U,PC

```



## ECHA:

ECHA permet l'échange de codes vocaux entre la zone MOTX et la zone auxiliaire intitulée MOTY. L'échange se fait indépendamment des longueurs des expressions contenues dans MOTX et MOTY. La zone MOTY doit avoir été préalablement chargée. Le drapeau CHAR indique l'état de la zone auxiliaire MOTY.

La temporisation en début de programme évite deux échanges successifs (ce qui équivaut à pas d'échange du tout), si un utilisateur distrait laisse la pointe du crayon optique sur le pictogramme. Un bip indique que l'opération d'échange est terminée.

## Registres et drapeaux utilisés ou modifiés

CHAR: Indique l'état de la mémoire auxiliaire

CHAR = 0 → mémoire non chargée

CHAR = 1 → mémoire chargée

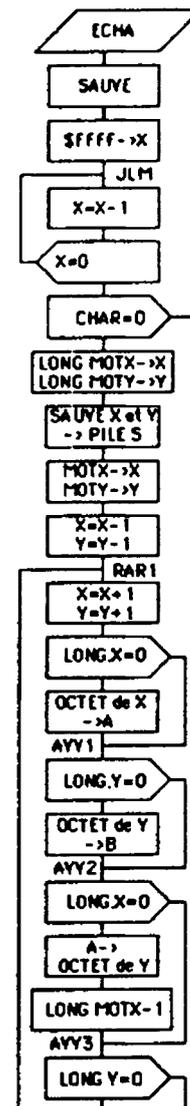
## MOTY-MOTX

### Listing source

```

ECHA PSHS X,Y,D,U Sauvegarde
      LDX #FFFF
JLM  LEAX -1,X Tempo
      BNE JLM
NIRA TST CHAR La memoire Y est
# elle chargee?
      BEQ AVV5 Non.Alors Pas d'
# echanges Possibles
      LDX MOTX Oui.Longueurs
# des mots stockes dans MOTX et MOTY
      LDY MOTY
      PSHS X,Y Sauve les longu-
# eurs dans la Pile
      LDX #MOTX Pointeurs sur les
# zones MOTX et MOTY
      LDY #MOTY
      LEAX -1,X Facilites de Pro
# gramation
      LEAY -1,Y
RAR1 LEAX 1,X Pointe sur l'octet
# suivant
      LEAY 1,Y
      LDU ,S La longueur restante
# a transferer de MOTX est-elle egale a
# zero ?
      BEQ AVV1 Oui.Alors on ne
# fait rien
      LDA ,X Non.On charge l'
# octet suivant dans A
AVV1 LDU 2,S La longueur restante
# a transferer de MOTY est-elle egale a
# zero?
      BEQ AVV2 Oui.Alors on ne
# fait rien
      LDB ,Y Non.On charge l'
# octet suivant dans B
AVV2 LDU ,S La longueur restante
# a transferer de MOTX est-elle egale a
# zero ?
      BEQ AVV3 Oui.alors on ne
# fait rien
      STA ,Y Non.On charge l'
# octet de MOTY avec la valeur de A
      LEAU -1,U On decremente la
# longueur de MOTX
      STU ,S
AVV1 LDU 2,S La longueur restante;
# a transferer de MOTY est-elle egale a
# zero?
      BEQ AVV4 Oui.Alors on ne
# fait rien
  
```

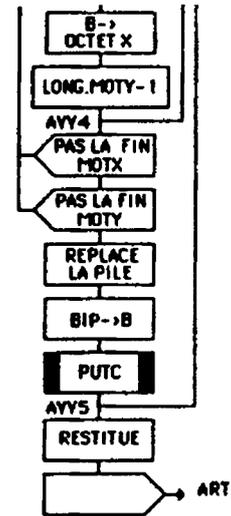
### Organigramme



```

      STB      ,X      Non.On charge l'
* octet de MOTX avec la valeur de B
      LEAU    -1,U      On decremente la
* longueur de MOTY
      STU      2,S
AVV4  CMPU    #0      Fin de MOTY?
      BNE     RAR1 Non.Alors on continue
      LDU     ,S      Fin de MOTX
      CMPU    #0
      BNE     RAR1 Non.Alors ON continue
      LEAS   4,S RePositionne la Piles
      LDB     ##07     Bip
      JSR     PUTC
      HVV5    PULS     X,Y,U,D
      LBRA   ART

```



## VMEM:

VMEM est la partie du programme principal qui permet le transfert des codes vocaux contenus dans la zone MOTX dans la zone de mémoire pointée par l'étiquette MOTY. Seuls les codes contenus dans la partie de mémoire comprise entre PTA et PTB sont transférés dans MOTY. Les quatre octets de tête, spécifiant la longueur, la hauteur finale et la hauteur initiale, sont positionnés afin de respecter le format d'un mot tel qu'il a été défini dans les pages précédentes.

Si aucun phonème n'a été chargé dans la zone MOTX, le pointeur POITX est égal à MOTX ce qui a pour conséquence, après avoir généré deux bips, de brancher la sortie du test en fin du programme. Après le chargement effectif de MOTY, un seul bip est généré.

## Registres et drapeaux utilisés ou modifiés

MOTX-POITX-PTB-TABT1-PTA-MOTY-CHAR

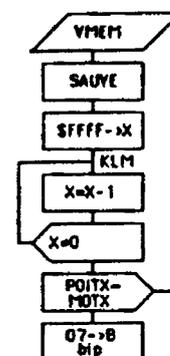
## Listing source

```

VMEM  PSHS     X,Y,D     Sauvegarde
      LDX     ##FFFF
KLM    LEAX    -1,X      Temporisation
      BNE     KLM
      LDX     #MOTX     A-t-on deja entre
* un Phoneme?
      CMPX   POITX
      BNE     GUI      Oui.
      LDB     ##07     Non.Alors 2 bips
      JSR     PUTC
      LDB     ##07
      JSR     PUTC

```

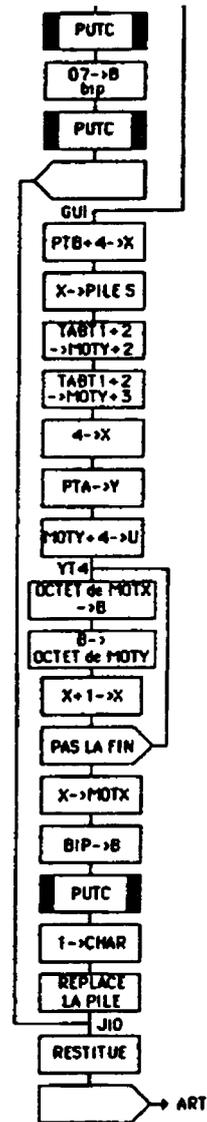
## Organigramme



```

      BRA      J10      On saute a la fin
GUI   LDX      PTB      Calcul la limite
* du dernier mot et...
      LEAX     4,X
      PSHS    X      .la stocke dans la Pile
      LDA     TABT1+2 Hauteur initiale
* Prelevee dans TABT1...
      STA     MOTY+2 et stockee dans la
* memoire auxilliaire MOTY
      STA     MOTY+3 idem
      LDX     #4      Initialise la lon-
* gueur du mot a 4
      LDY     PTA      Le registre Y con-
* tient le pointeur du curseur A
      LDU     #MOTY+4 4em octet de la
* zone MOTY
YT4   LDB     ,Y+ 1 octet de MOTX dans B
      STB     ,U+      B dans un octet de
* MOTY
      LEAX    1,X      Incrmente la lon-
* gueur du nouveau mot
      CMPY    ,S      Ne pointe-t-on pas
* sur la limite ?
      BNE     YT4      Non.Alors on con-
* tinue
      STX     MOTY     Oui,Alors sauve la
* longueur du nouveau mot dans le deux
* Premier octets de MOTY
      LDB     #$07     BIP
      JSR     PUTC
      CLR     CHAR     La memoire MOTY est
* chargee
      INC     CHAR
      LEAS    2,S      RePositionne la
* Pile S
J10   PULS    X,Y,D     Restitue
      LBRA    ART

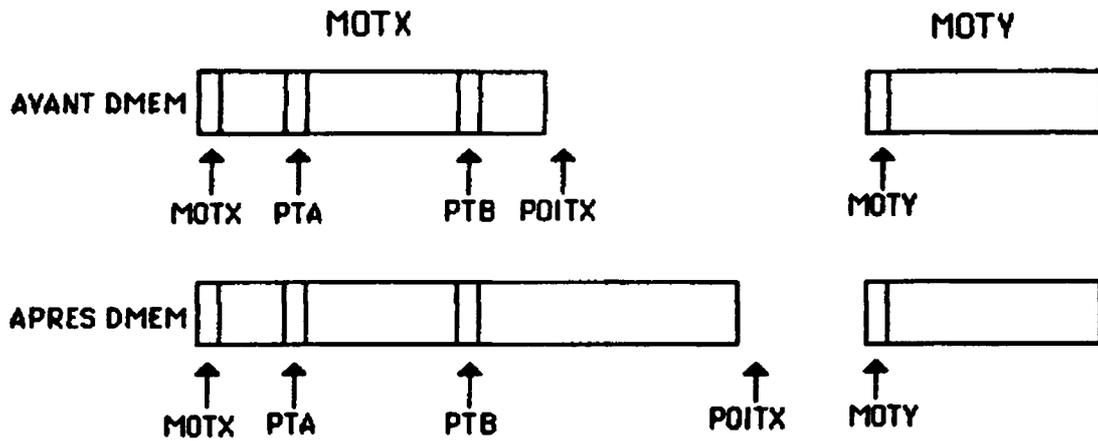
```



## DMEM:

DMEM permet la concaténation des codes vocaux contenus dans la zone MOTY avec ceux contenus dans MOTX. Les codes vocaux de MOTY sont positionnés, sur MOTX, à partir de la trame pointée par le curseur B. Les codes "recouverts" de MOTX sont perdus. La "hauteur avant" de la trame pointée par le curseur B est chargée avec la hauteur initiale du mot contenu dans MOTY.

En sortie du programme DMEM, le pointeur POITX pointe sur la fin du mot issu de la concaténation qui devient le mot contenu dans MOTX. La longueur de MOTX est modifiée en conséquence. Notez que le contenu de MOTY n'est pas perdu et peut être utilisé pour une nouvelle concaténation ou un échange avec MOTX.



### Registres et drapeaux utilisés ou modifiés

MOTX-POITX-PTB-TABT2-PTA-MOTY-CHAR

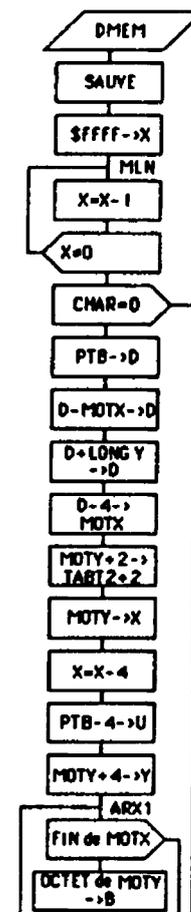
### Listing source

```

DMEM  PSHS   X,Y,D,U  Sauvegarde
      LDX   #$FFFF  Temporisation
MLN   LEAX   -1,X
      BNE   MLN
      TST   CHAR    La zone MOTY est-
* elle chargée
      BEQ   AVX1    Non, Alors rien
*CALCUL LA LONGUEUR
      LDD   PTB    Pointeur du curseur B
      SUBD  #MOTX  Calcul le nombre
* d'octets entre le PTB et le debut de
* la zone MOTX...
      ADDD  MOTY  ..Puis on ajoute la
* longueur du MOTY
      SUBD  #4    Il y a 4 Octets en
* trop, car les 4 octets de debut de mot
* sont comptés 2 fois
      STD   MOTX  Longueur des deux
* mots concatenes
      LDB   MOTY+2 Hauteur initiale
* du mot colle...
      STB   TABT2+2 ..dans la table
* TABT2
      LDX   MOTY  Longueur de MOTY
* transferer
      LEAX  -4,X  Moins les 4 Pre-
* miers octets
      LDU   PTB   Pointeur du curs-
* eur B
      LEAU  4,U   Plus les 4 octets
* de la dernière trame

```

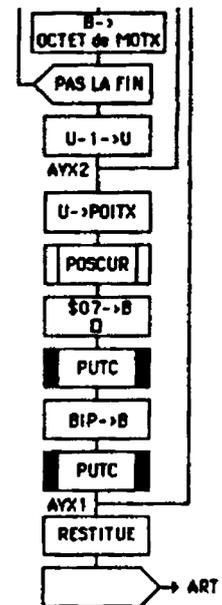
### Organigramme



```

LDY #MOTY+4 Pointe sur le
* Premier octet de la zone a Transférer
ARX1 CMPU #MOTX+250 Depassement de
* la zone MOTX?
BHS AVX2 Oui.
LDB ,Y+ Non.Alors on char-
* se B avec le Premier octet de MOTY ...
STB ,U+ ..que l'on transfere
* dans MOTX
LEAX -1,X Fin?
BNE ARX1 Non.Alors on con-
* tinue
LEAU -1,U Oui,on retable U
* qui Pointe trop loin
AVX2 STU POITX Nouveau POITX
JSR POSCUR Positionne le
* curseur
LDB ##7F Print un carre
JSR PUTC
LDB ##07 Un bip
JSR PUTC
AVX1 PULS X,Y,D,U
LBPA ART

```



## CASSS et DISQS:

CASSS et DISQS sont les parties du programme principal qui permettent la sauvegarde sur la disquette (DISQS) ou sur la cassette (CASSS) de l'expression contenue dans la zone mémoire MOTY. Ces routines ont la particularité de retourner sous contrôle du BASIC pour accéder à l'instruction de sauvegarde sur un périphérique (SAVEM). Au retour du BASIC (étiquette EFAC), la zone de saisie matérialisée par un message, suivie d'un champ de 8 caractères, est effacée avant le branchement à ART.

Les suffixes des descripteurs de fichiers spécifiant des codes vocaux sont forcés à .SYT afin de limiter les risques de confusion entre les fichiers. Si la longueur du nom du fichier est inférieure à 8 caractères, le nom est complété à 8 par une suite d'espaces.

### Registres et drapeaux utilisés ou modifiés

PARSO: Le registre PARSO indique au programme BASIC la routine objet appelante.

```

PARSO = 1 → DISQS          PARSO = 2 → DISQL
PARSO = 3 → CASSS         PARSO = 4 → CASSL
PARSO = 5 → Retour définitif sous BASIC

```

**LIGN1:** LIGN1 est une chaîne de caractères, pointée par LIGN1, qui spécifie la position du curseur et envoie un message invitant l'opérateur à entrer le nom qu'il souhaite donner à l'expression sauvegardée.

```
LIGN1      FCB      $1F,$52,$4A
           FCC      /"NOM" A SAUVER:/
           FCB      $7F,$7F,$7F,$7F
           FCB      $7F,$7F,$7F,$7F
           FCB      $1B,$45
           FCB      $1F,$52,$59,00
```

**FIC1:** Début de la zone de 8 octets contenant le nom du fichier.

**MSG1-3:** Chaîne de repositionnement de la couleur courante d'affichage.

**SGAR:** Zone utilisée par la pile U pour sauvegarder l'état des registres S et DP afin de les repositionner avant le retour au BASIC.

**ADF:** Registre 16 bits contenant l'adresse de fin de la zone MOTY

**LIGNO:** Chaîne de caractères permettant d'effacer la zone de saisie.

```
LIGNO      FCB      $1F,$52,$49,$1B,$4C
           FCB      $1B,$40,$1B,$50
           FCC      /oooooooooooooooooooooooooooo/
           FCB      00
```

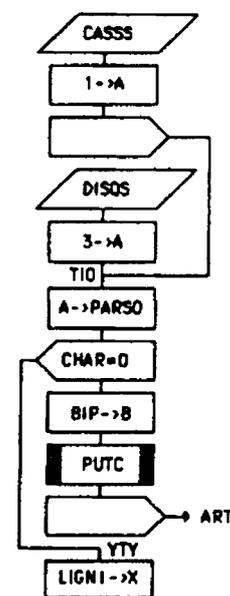
**SELEC:** registre de données du PIA système

**CHAR-MOTY**

### Listing source

```
*Point d'entree cassette
CASSS LDA #3 3-->A Pour Posi-
* tionner le registre PARSO
      BRA TIO
*Point d'entree disquette
DISQS LDA #1 1-->A Pour Posi-
* tionner le registre PARSO
TIO STA PARSO Charge PARSO
   TST CHAR La zone (MOTY)
* est-elle chargée?
   BNE YTY Oui.
   LDB #07 Non. Alors on action-
* ne un BIP
      JSR PUTC
   LBRA ART Retour en ART
YTY LDX #LIGN1 Message de sauve-
* garde
```

### Organigramme



```

        JSR    MESSA
        LDX    #FIC1  Debut de la zone
* nom de fichier dans X
        LDA    #8     8 caracteres Par
* nom de fichier
        JSR    MAJUS  Force en majuscule
VBMS   JSR    CLAY1  Rentre un caractere
        CMPB   #50D   Ce caractere est-il
* "ENTREE"
        BEQ    VBNS   Oui.
        STB    ,X+    Non.Alors on l'af-
*fiche
        JSR    PUTC
        DECA
* de caracteres
        BNE    VBMS   La fin? Non.Alors
* on rentre un nouveau caractere
        BRA    MUIS   Oui.
VBNS   LDB    #520   On complete Par
* des espaces
        STB    ,X+
        DECA
        BNE    VBNS
MUIS   LDX    #MSG1-3 Remise en rouge
        JSR    MESSA
        LDX    #MOTY  Debut de zone dans X
        LDD    MOTY   Longueur dans D
        LEAX   D,X    Calcul la fin et..
        STX    ADF    ..la stocke dans ADF
        LDU    #SGAR  Repositionne le
* Pointeur de Pile S et le registre de
* Page dans l'etat qu'ils etaient en en-
* trant dans le sous-Programme
        PULJ   S,DP
        RTS
        Retour au BASIC

```

```

*
* ICI EXECUTION en BASIC
*

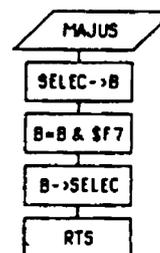
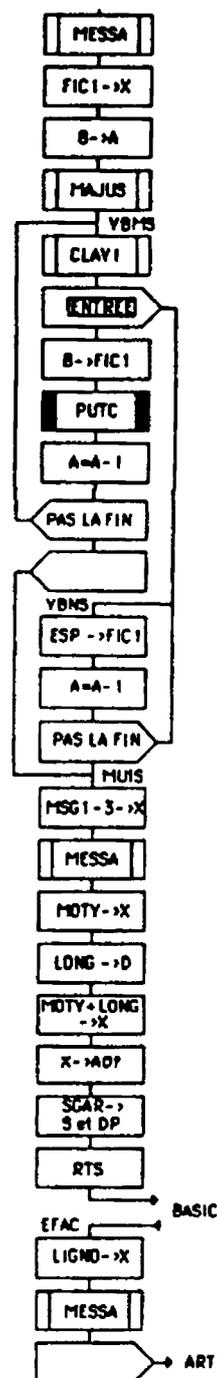
```

```

EFAC   LDX    #LIGN0  APres le BASIC.
* Pour effacer la ligne du "NOM"
        JSR    MESSA
        LBRA   ART     Branche en ART

MAJUS  PSHS   D
        LDB    SELEC  Registre SELEC-->B
        ANDB   #F7    Force 0 dans bit3
        STB    SELEC  B--> SELEC
        PULS   D,PC   Restitue et RTS

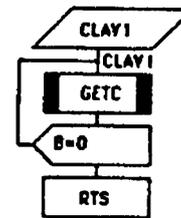
```



```

CLAV1 JSR   GETC   Routine clavier
      TSTB   Une touche a ete
* sollicitee?
      BEQ   CLAV1  Non.Alors on boucle
      PULS  PC     RTS

```



## CASSL et DISQL:

CASSL et DISQL sont les parties du programme principal qui permettent de charger, à partir de la cassette (CASSL) ou de la disquette (DISQS), des codes vocaux précédemment sauvegardés sur le périphérique spécifié. A l'issue de CASSL ou DISQL, les codes vocaux sont chargés dans la zone MOTY et le drapeau CHAR est positionné à 1. Les codes éventuellement contenus dans MOTY sont "écrasés".

Tout comme CASSS et DISQS, ces routines de chargement de codes vocaux retournent sous contrôle du BASIC pour utiliser l'instruction de chargement (LOADM). Le retour du BASIC s'effectue par le branchement à l'étiquette EFAC.

### Registres et drapeaux utilisés ou modifiés

LIGN: LIGN est une chaîne de caractères, pointée par LIGN, qui spécifie la position du curseur et envoie un message invitant l'opérateur à entrer le nom de l'expression qu'il souhaite charger.

```

LIGN      FCB      $1F,$52,$4A
          FCC      /"NOM" A CHARGER:/
          FCB      $7F,$7F,$7F,$7F
          FCB      $7F,$7F,$7F,$7F
          FCB      $1B,$45
          FCB      $1F,$52,$59,00

```

PARSO-FIC1-MSG1-3-CHAR-SGAR

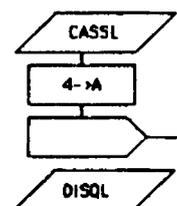
### Listing source

```

*Point d'entree cassette
CASSL LDA   #4      4-->A Pour Pos1-
* tionner le registre PARSO
      BRA   TYU
*Point d'entree disquette
DISQL LDA   #2      1-->A Pour Pos1-
* tionner le registre PARSO
TYU   STA   PARSO   Charge PARSO

```

### Organigramme



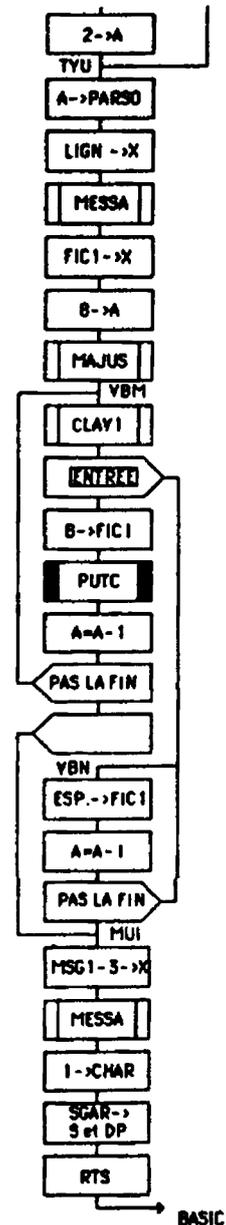
```

LDX #LIGN Message de charge-
* ment
JSR MESSA
LDX #FIC1 Debut de la zone
* nom de fichier dans X
LDA #8 8 caracteres Par
* nom de fichier
JSR MAJUS Force en majuscule
VBM JSR CLAV1 Entre un caractere
CMPB #0 Ce caractere est-il
* "ENTREE"?
BEQ VBN Oui.
STB ,X+ Non. Alors on l'af-
* fiche
JSR PUTC
DECA Decompte le nombre
* de caracteres
BNE VBM La fin? Non. Alors
* on entre un nouveau caractere
BRA MUI Oui.
VBN LDB #20 On complete Par
* des espaces
STB ,X+
DECA
BNE VBN

MUI LDX #MSG1-3 Remise en rouge
JSR MESSA
CLR CHAR Positionne CHAR a 1
INC CHAR
LDU #SGAR RePositionne le
* Pointeur de Pile S et le registre DP
* dans l'etat qu'ils avaient en entrant
* dans le sous programme
PULU DP,S Retour au BASIC
RTS

*
* ICI execution sous BASIC
*
* et retour en EFAC
RTS

```



## BW1A-BW1B-BW2A-BW2B-BW3A-BW3B-BW4A-BW4B:

Les tronçons de programme référencés ci-dessus permettent de modifier les largeurs de bandes passantes des formants. Le drapeau BLOQ autorise ou interdit l'action de ces parties de programme. Le registre X doit contenir le pointeur de trame (PTA ou PTB). Le registre U définit les constantes de masquages utilisées dans le sous-programme MBWT. Le registre B spécifie la constante de variation utilisée, en tenant compte de l'emplacement dans l'octet des bits affectés à ce paramètre.

### Registres et drapeaux utilisés ou modifiés

BLOQ-PTA-PTB

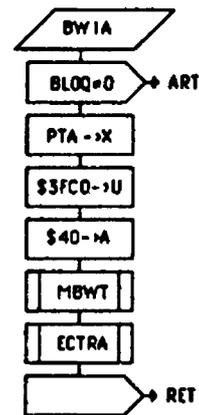
## BW1A:

### Listing source

```

BW1A  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloquées?
      LBNE    ART     Oui.Alors Pas d'
* action
      LDX    PTA     Non.On charge le
* Pointeur A dans le registre X
      LDU    #$3FC0  Constantes de mas-
* quages et de limite du Parametre
      LDB    #$40    Constante de modi-
* fication
      JSR    MBWT
      JSR    ECTRA
      LBRA   RET
  
```

### Organigramme



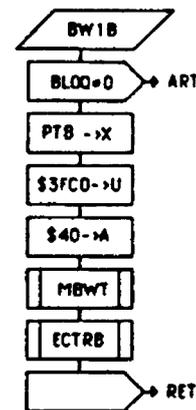
## BW1B:

### Listing source

```

BW1B  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloquées?
      LBNE    ART     Oui.Alors Pas d'
* action
      LDX    PTB     Non.On charge le
* Pointeur B dans le registre X
      LDU    #$3FC0  Constantes de mas-
* quages et de limite du Parametre
      LDB    #$40    Constante de modi-
* fication
      JSR    MBWT
      JSR    ECTRB
      LBFA   RET
  
```

### Organigramme



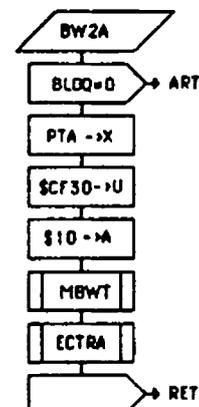
## BW2A:

### Listing source

```

BW2A  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloquées?
      LBNE    ART     Oui.Alors Pas d'
* action
      LDX    PTA     Non.On charge le
* Pointeur A dans le registre X
      LDU    #$CF30  Constantes de mas-
* quages et de limite du Parametre
      LDB    #$10    Constante de modi-
* fication
      JSR    MBWT
      JSR    ECTRA
      LBRA   RET
  
```

### Organigramme

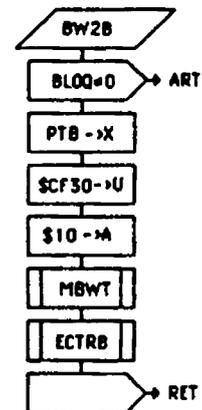


## BW2B:

### Listing source

```
BW2B  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloques?
      LBNE   ART     Oui.Alors Pas d'
* action
      LDX   PTB     Non.On charge le
* Pointeur B dans le registre X
      LDU   #$F30   Constantes de mas-
* quages et de limite du Parametre
      LDB   #$10    Constante de modi-
* fication
      JSR   MBWT
      JSR   ECTRB
      LBRA  RET
```

### Organigramme

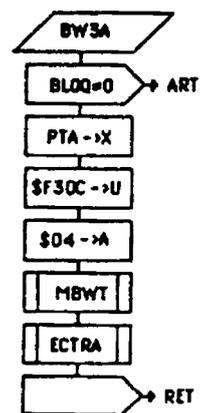


## BW3A:

### Listing source

```
BW3A  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloques?
      LBNE   ART     Oui.Alors Pas d'
* action
      LDX   PTA     Non.On charge le
* Pointeur A dans le registre X
      LDU   #$F30C  Constantes de mas-
* quages et de limite du Parametre
      LDB   #$04    Constante de modi-
* fication
      JSR   MBWT
      JSR   ECTRA
      LBRA  RET
```

### Organigramme

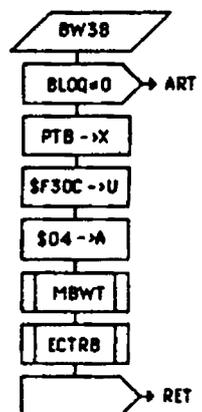


## BW3B:

### Listing source

```
BW3B  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloques?
      LBNE   ART     Oui.Alors Pas d'
* action
      LDX   PTB     Non.On charge le
* Pointeur B dans le registre X
      LDU   #$F30C  Constantes de mas-
* quages et de limite du Parametre
      LDB   #$04    Constante de modi-
* fication
      JSR   MBWT
      JSR   ECTRB
      LBRA  RET
```

### Organigramme

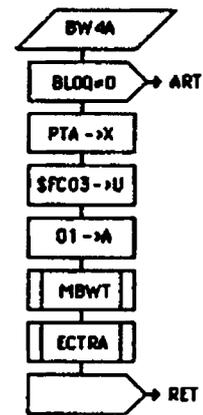


## BW4A:

### Listing source

```
BW4A  TST   BLOQ   Les commandes de
* variation de Parametres sont-elles
* bloquées?
      LEHE   ART    Oui.Alors Pas d'
* action
      LDX   PTA    Non.On charge le
* Pointeur A dans le registre X
      LDU   ##FC03 Constantes de mas-
* quages et de limite du Parametre
      LDB   ##01   Constante de modi-
* fication
      JSR   MBWT
      JSR   ECTRA
      LBRA  RET
```

### Organigramme

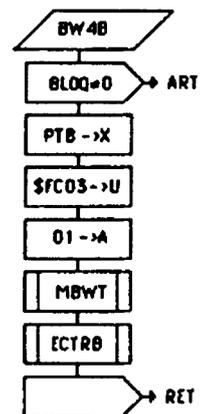


## BW4B:

### Listing source

```
BW4B  TST   BLOQ   Les commandes de
* variation de Parametres sont-elles
* bloquées?
      LEHE   ART    Oui.Alors Pas d'
* action
      LDX   PTB    Non.On charge le
* Pointeur B dans le registre X
      LDU   ##FC03 Constantes de mas-
* quages et de limite du Parametre
      LDB   ##01   Constante de modi-
* fication
      JSR   MBWT
      JSR   ECTRB
      LBRA  RET
```

### Organigramme



## MBWT:

MBWT est le sous-programme qui modifie l'octet pointé par le registre X, en tenant compte de la position du paramètre sur l'octet et du sens de variation demandé. Le sens de variation est passé par le registre MPAR.

Les limites d'évolution des paramètres sont définies par la borne "00" et une valeur dite "limite haute" passée par les poids faibles du registre U.

Le paramètre ne peut être modifié que s'il est compris entre ces deux limites, en excluant les égalités. Si le paramètre est égal à l'une des deux limites, le sous-programme "actionne" un bip et branche à la fin du sous-programme.

## Registres et drapeaux utilisés ou modifiés

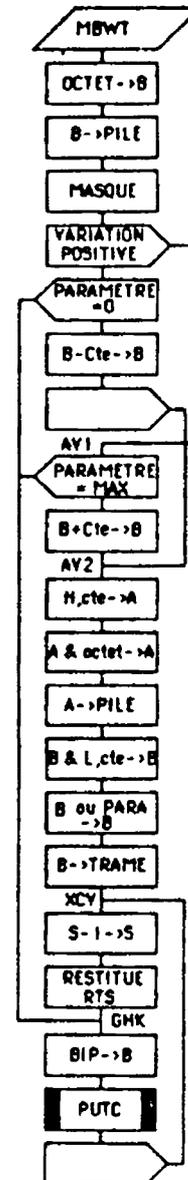
### MPAR

#### Listing source

```

MBWT  PSHS  U,B
      LDB  ,X      Lecteur de l'octet
* contenant le Parametre
      PSHS  B      Sauvegarde dans la
* Pile
      ANDB  3,S    Masque Pour extraire
* le Parametre
      TST  MPAR    La variation du
* Parametre doit-elle etre Positive?
      BEQ  AV1     Oui.
      TSTB                    Non.Alors le Para-
* metre est-il egale a zero?
      BEQ  GHK     Oui.Alors Pas de
* modification Possible.
      SUBB  1,S    Non.On soustrait
* la constante de variation.Passee Par B
* et stockee dans la Pile S
      BRA  AV2
AV1   CMPB  3,S    Le Parametre est-il
* egal a la valeur max.
      BEQ  GHK     Oui.Alors Pas de
* modification Possible
      ADDB  1,S    Non.On additionne
* la constante de variation.Passee Par B
* et stockee dans la Pile S
AV2   LDA  2,S    Poids fort de la
* constante de masquage
      ANDA  ,S     Pour recuperer le
* reste les autres bits de l'octet
      STA  ,S
      ANDB  3,S    Masque
      ORB  ,S     Pour "reformer"
* l'octet et....
      STB  ,X     ... le rePlacer
* dans la trame
XCV   LEAS  1,S    RePositionne le
* Pointeur de Pile S
      PULS  U,B,PC
GHK   LDB  #$07   Un bIP quand la
* modification est impossible
      JSR  PUTC
      BFA  XCV
  
```

#### Organigramme



## ECTRA et ECTRB:

Les sous-programmes ECTRA et ECTRB permettent la transcription de la trame pointée par PTA ou PTB sur la zone MOTX dans les tables TABT1 ou TABT2. Ces sous-programmes assurent également l'affichage des trames, pointées par les curseurs, après modification. Si PTA est égal à PTB, ce qui indique que les deux curseurs pointent sur la même trame,

la modification d'un curseur est alors reportée sur l'autre curseur afin de conserver un affichage identique. Après la modification du paramètre, l'appel de la routine TRATRA permet à l'opérateur d'écouter les conséquences des modifications qu'il vient d'effectuer.

### Registres et drapeaux utilisés ou modifiés

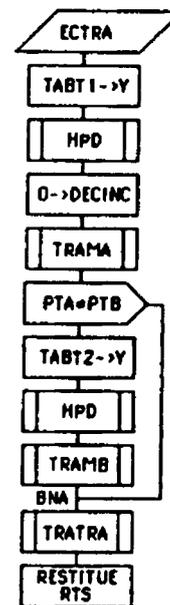
TABT1-TABT2-DECINC-PTB-PTA

ECTRA:

#### Listing source

```
ECTRA PSHS Y
      LDY #TABT1 Table TABT1
      JSR HPD Transcription de
* MOTX vers TABT1
      CLR DECINL La trame ne doit
* Pas etre incrementee ni decrementee
      JSR TRAMA Affiche la trame A
      CMPX PTB Le curseur B et le
* curseur A Pointent-ils sur la meme
* trame ?
      BNE ENA Non.
      LDY #TABT2 Oui. Alors il faut
* modifier l'affichage de la trame B
      JSR HPD Transcription de
* MOTX vers TABT2
      JSR TRAMB Affiche la trame B
BNA JSR TRATRA Prononce le troncon
* d'expression compris entre le curseur
* A et le curseur B
      PULS Y,PC
```

#### Organigramme

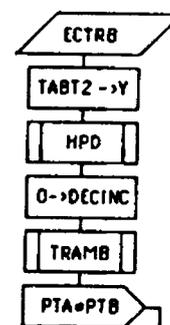


ECTRB:

#### Listing source

```
ECTRB PSHS Y
      LDY #TABT2 Table TABT2
      JSR HPD Transcription de
* MOTX vers TABT2
      CLR DECINC La trame ne doit
* Pas etre incrementee ni decrementee
      JSR TRAMB Affiche la trame B
      CMPX PTA Le curseur A et le
* curseur B Pointent-ils sur la meme
* trame ?
      BNE BNB Non.
      LDY #TABT1 Oui. Alors il faut
```

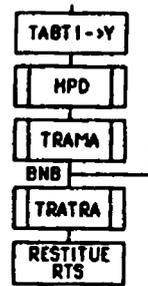
#### Organigramme



```

* modifier l'affichage de la trame A
      JSR   HPD      Transcription de
* motx vers TABT1
      JSR   TRAMA   Affiche la trame A
BNB   JSR   TRATRA  Prononce le tronçon
* d'expression compris entre le curseur
* A et le curseur B
      PULS  Y,PC

```



## FR1A-FR1B-FR2A-FR2B-FR3A-FR3B :

Les tronçons du programme principal ci-dessus modifient la valeur du formant qui leur est affecté. Si le drapeau BLOQ est positionné sur une valeur différente de zéro, la modification du formant ne peut être effectuée. Le registre X doit contenir la valeur du pointeur qui est, ensuite, positionné sur l'octet contenant les données binaires du formant à modifier. Le registre U est chargé avec les constantes de masquage et de limite utilisées dans le sous-programme MBWT. Le registre B indique la constante à ajouter ou à retrancher à la valeur binaire du formant, en tenant compte de son positionnement sur l'octet.

### Registres et drapeaux utilisés ou modifiés

#### BLOQ-PTB-PTA

#### FR1A:

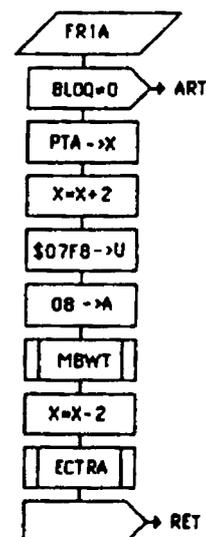
#### Listing source

```

FR1A  TST    BLOQ   Les commandes de
# variation de Parametres sont-elles
# bloquées ?
      LBNE   ART     Oui. Alors pas d'
# action
      LDX   PTA     Non. On charge le
# pointeur A dans le registre X.
      LEAX  2,X     Le Parametre est
# positionné sur le 3em octet de la tra-
# me soit: X+2
      LDU   #$07F8  Constantes de mas-
# quages et limites du Parametre
      LDB   #$08    Constante de modi-
# fication
      JSR   MBWT
      LEAX -2,X     Pour repositionner
# le registre X avec le pointeur A
      JSR   ECTRA
      LBRA  RET

```

#### Organigramme

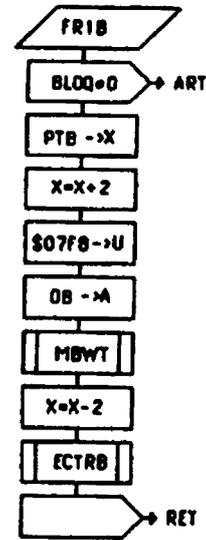


## FR1B:

### Listing source

```
FR1B  TST    BLOC    Les commandes de
* variation de Parametres sont-elles
* bloquee ?
      LBNE   ART     Oui. Alors Pas d'
* action
      LDX   PTB     Non. On charge le
* Pointeur B dans le registre X.
      LEAX  2,X     Le Parametre est
* Positionne sur le 3em octet de la tra-
* me soit: X+2
      LDU   #$07F8  Constantes de mas-
* quages et limites du Parametre
      LDB   #$08    Constante de modi-
* fication
      JSR   MBWT
      LEAX  -2,X    Pour repositionner
* le registre X avec le Pointeur B
      JSR   ECTRB
      LBRA  RET
```

### Organigramme

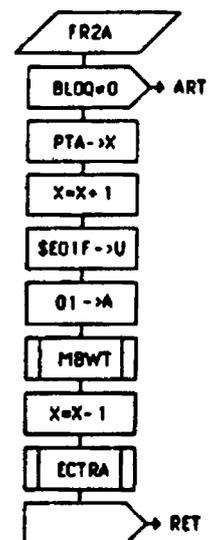


## FR2A:

### Listing source

```
FR2A  TST    BLOC    Les commandes de
* variation de Parametres sont-elles
* bloquee ?
      LBNE   ART     Oui. Alors Pas d'
* action
      LDX   PTA     Non. On charge le
* Pointeur A dans le registre X.
      LEAX  1,X     Le Parametre est
* Positionne sur le 2em octet de la tra-
* me soit: X+1
      LDU   #$E01F  Constantes de mas-
* quages et limites du Parametre
      LDB   #$01    Constante de modi-
* fication
      JSR   MBWT
      LEAX  -1,X    Pour repositionner
* le registre X avec le Pointeur A
      JSR   ECTRA
      LBRA  RET
```

### Organigramme

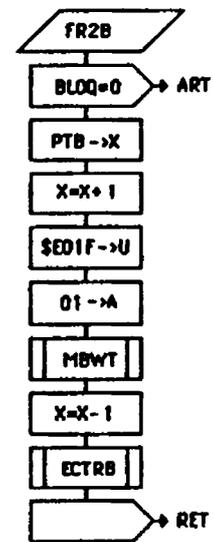


## FR2B:

### Listing source

```
FR2B  TST    BLOQ  Les commandes de
* variation de Parametres sont-elles
* bloqueees ?
      LBNE   ART    Oui. Alors Pas d'
* action
      LD%    PTB    Non. On charge le
* Pointeur B dans le registre X.
      LEAX   1,%    Le Parametre est
* Positionne sur le 2em octet de la tra-
* me soit: X+1
      LDU    #$E01F Constantes de mas-
* quages et limites du Parametre
      LDB    #$01  Constante de modi-
* fication
      JSR    MBWT
      LEAX   -1,%   Pour rePositionner
* le registre X avec le Pointeur B
      JSR    ECTRB
      LBRA   RET
```

### Organigramme

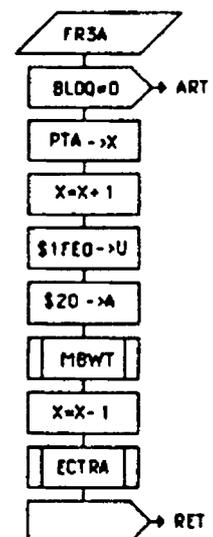


## FR3A:

### Listing source

```
FR3A  TST    BLOQ  Les commandes de
* variation de Parametres sont-elles
* bloqueees ?
      LBNE   ART    Oui. Alors Pas d'
* action
      LD%    PTA    Non. On charge le
* Pointeur A dans le registre X.
      LEAX   1,%    Le Parametre est
* Positionne sur le 2em octet de la tra-
* me soit: X+1
      LDU    #$1FE0 Constantes de mas-
* quages et limites du Parametre
      LDB    #$20  Constante de modi-
* fication
      JSR    MBWT
      LEAX   -1,%   Pour rePositionner
* le registre X avec le Pointeur A
      JSR    ECTRA
      LBRA   RET
```

### Organigramme

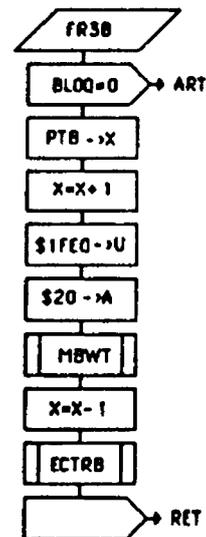


## FR3B:

### Listing source

```
FR3B  TST    BLOQ    Les commandes de
# variation de Parametres sont-elles
# bloquées ?
      LBNE    ART     Oui. Alors Pas d'
# action
      LDX    PTB     Non. On charge le
# Pointeur B dans le registre X.
      LEAX   1,X     Le Parametre est
# Positionne sur le 2em octet de la tra-
# me soit X+1
      LDU    #$1FE0  Constantes de mas-
# quages et limites du Parametre
      LDB    #$20    Constante de modi-
# fication
      JSR    MBWT
      LEAX   -1,X    Pour rePositionner
# le registre X avec le Pointeur B
      JSR    ECTRB
      LBRA   RET
```

### Organigramme



## APLA-APLB:

Les parties du programme principal intitulées APLA et APLB permettent de modifier les valeurs binaires de l'amplitude des trames pointées par les curseurs A ou B. Le registre X spécifie la trame pointée (PTA ou PTB). L'action de APLA ou APLB peut être inhibée en chargeant le drapeau BLOQ à une valeur différente de zéro.

### Registres et drapeaux utilisés ou modifiés

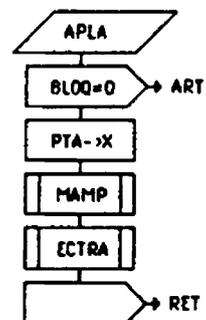
#### BLOQ-PTB-PTA

#### APLA:

### Listing source

```
APLA  TST    BLOQ    Les commandes de
# variation de Parametres sont-elles
# bloquées?
      LBNE    ART     Oui. Alors Pas d'
# action
      LDX    PTA     Non. On charge le
# Pointeur A dans le registre X
      JSR    MAMP    Sous Programme
# de modification l'amplitude
      JSR    ECTRA  Affichage
      LBRA   RET
```

### Organigramme

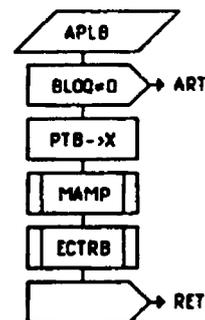


## APLB:

### Listing source

```
APLB  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloques?
      LBNE   ART     Oui. Alors Pas d'
* action
      LDX   PTB     Non. On charge le
* Pointeur B dans le registre X
      JSF   MAMP    Sous Programme
* de modification l'amplitude
      JSR   ECTRB   Affichage
      LBRA  RET
```

### Organigramme



## MAMP:

MAMP est la routine appelée par FR1A ou FR1B qui permet de modifier la valeur binaire représentant l'amplitude de la trame pointée par le curseur A ou B. Le curseur est spécifié par le registre X du 6809. Le registre MPAR indique le sens de la modification de l'amplitude (augmentation ou diminution). La valeur binaire représentant l'amplitude peut évoluer de 0 à 15 en décimal ou 0 à 0F en hexadécimal. Au delà de ces limites, l'amplitude ne peut pas être modifiée et le sous-programme branche à la fin de la routine en signalant l'action par un bip.

Remarquez que les valeurs binaires de l'amplitude des trames sont implantées "à cheval" sur deux octets (octet 3 et 4 de la trame) ce qui a nécessité la rédaction d'un sous-programme de traitement spécifique à l'amplitude.

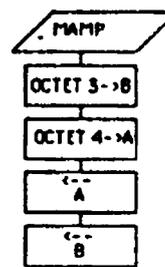
### Registres et drapeaux utilisés ou modifiés

## MPAR:

### Listing source

```
MAMP  PSHS   D       Sauvegarde
      LDB   2,X     3em octet de la
* trame dans B
      LDA   3,X     4em octet de la
* trame dans A
      LSLA                      Le 1er bit de l'am
* plitude Passe dans la retenue (bit C)
      ROLB                      L'amplitude est cadree
* sur les 4 bits de Poids faibles de B
```

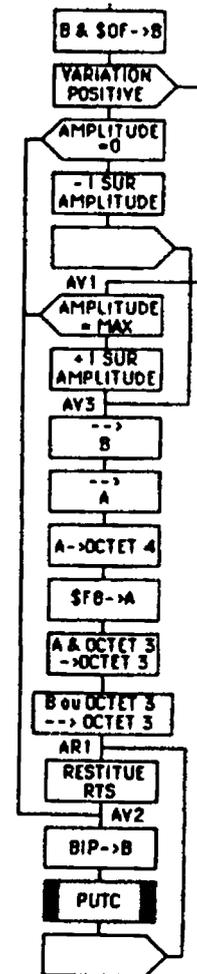
### Organigramme



```

    ANDB    ##0F    Masque Pour forcer
* les Poids forts a zero
    TST    MPAR    La variation du
* Parametre est- elle Positive?
    BEQ    AV1    Oui. Alors on saute
    TSTB
* tude est -elle egle a zero?
    BEQ    AV2    Oui. Alors l'ampli-
* tude reste ce qu'elle est.
    DECB
    BRA    AV3    Non. Alors moins 1
AV1    CMPB    ##0F    L'amplitude est-
* elle egle a la valeur maximum?
    BEQ    AV2    Oui. Alors Pas de
* modification Possible
    INCB
    AV3    LSRB
* faible est mis dans la retenue (C)
    RORA
* A par le bit de Poids fort
    STA    3,X    RePositionne le
* 4em octet de la trame
    LDA    ##F8    Pour "recupere"
* les autres bits du 3em octet
    ANDA    2,X    Masque avec le 3em
* octet de la trame
    STA    2,X    Sauve
    ORB    2,X    Fait ou avec la
* derniere valeur de l'amplitude
    STB    2,X    RePositionne le
* 3em octet
AR1    PULS    D,PC    Restitue et RTS
AV2    LDB    ##07    Un biP quand la
* modification est impossible
    JSR    PUTC
    BRA    AR1

```



## DURA-DURB:

DURA et DURB sont les parties du programme principal qui autorisent la modification de la durée de la trame pointée par le curseur A ou B. L'adresse du premier octet de la trame est passée par le registre X auquel on doit ajouter 3 pour pointer sur l'octet contenant le code binaire représentant la durée.

Le drapeau BLOQ permet d'interdire l'action de DURA ou DURB. Les constantes de masquage et de limite sont chargées dans le registre U, tandis que B contient la valeur de l'incrément qu'il convient d'ajouter ou de retrancher à la valeur binaire de la durée.

Nous avons vu, en étudiant le MEA 8000, que la hauteur à la fin d'une trame (Hap) était fonction de deux données: la hauteur en entrant dans la trame (Hav) et la variation de hauteur générée par le produit du P.I. par la durée de la trame( $\Delta$ ).

Soit:  $Hap = Hav + \Delta$  avec  $\Delta = P.I. * DUREE$

(Le P.I. peut être positif ou négatif).

Une variation de la hauteur de la trame conduit donc à une modification de la hauteur en sortie de cette trame qui entraîne à son tour un changement dans la suite de l'expression.

Il convient donc de modifier le P.I. en sens inverse de la durée pour conserver la même hauteur en sortie. Cette fonction est réalisée par le sous-programme DEPT.

## Registres et drapeaux utilisés ou modifiés

### BLOQ-PTA-PTB

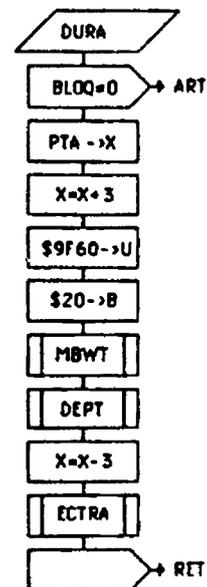
### DURA

#### Listing source

```

DURA  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloquées
      LBNE    ART      Oui.Alors Pas
* d'action
      LDX     PTA      Non.On charge le
* pointeur A dans le registre X
      LEAX    3,X      La duree est imp-
* lantée sur le 4em octet de la trame
      LDU     ##$9F60  Constantes de mas-
* quage et de limite
      LDB     ##$20    Constante de vari-
* ation
      JSR     MBWT     Variation de la DUREE
      JSR     DEPT     Ajuste le PITCH
      LEAX    -3,X     RePositionne le
* pointeur
      JSR     ECTRA    Affichage
      LBRA    RET
  
```

#### Organigramme



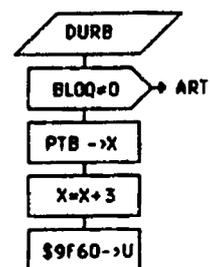
### DURB

#### Listing source

```

DURB  TST    BLOQ    Les commandes de
* variation de Parametres sont-elles
* bloquées
      LBNE    ART      Oui.Alors Pas
* d'action
      LDX     PTB      Non.On charge le
* pointeur B dans le registre X
      LEAX    3,X      La duree est imp-
* lantée sur le 4em octet de la trame
  
```

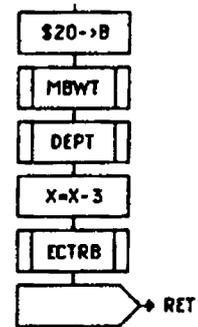
#### Organigramme



```

LDU    ##9F60  Constantes de mas-
* quage et de limite
LDB    ##20    Constante de vari-
* ation
JSR    MBWT    Variation de la DUREE
JSR    DEPT    Ajuste le PITCH
LEAX   -3,X    RePositionne le
* Pointeur
JSR    ECTRB   Affichage
LBRA   RET

```



## DEPT:

DEPT est le sous-programme qui ajuste le P.I. à l'opposé de la variation de la durée de la trame afin d'obtenir une hauteur en sortie constante. La durée est codée sur 2 bits ce qui autorise 4 valeurs de durée soit 8 ms, 16 ms, 32 ms et 64 ms. Chaque variation correspond donc à un doublement ou à une division par 2 de la durée. Dans ces conditions, il convient de multiplier ou de diviser également le P.I. par 2 à chaque variation de la durée de la trame. Si le doublement du P.I. conduit à un dépassement de la plage autorisée, le P.I. est forcé aux valeurs maximum positive (\$0F) ou négative (\$11). Les divisions de nombre impairs conduisent à une modification de la hauteur en sortie égale 1Hz/8 ms de durée.

### Registres et drapeaux utilisés ou modifiés

#### MPAR

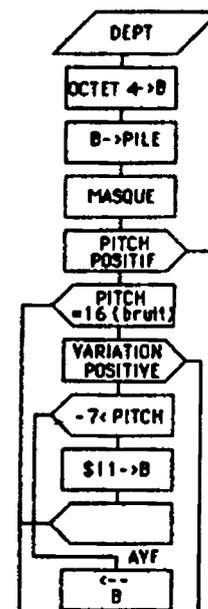
#### Listing source

```

DEPT  PSHS  D      Sauvegarde
      LDB   ,X      4em octet de la
* trame dans B
      PSHS  B      Sauvegarde de l'octet
      ANDB  ##1F    Masque pour ne
* conserver que le Pitch dans B
      BITB  ##10    Le Pitch est-il
* positif?(valeur hexa comprise entre
* 0 et 0F)
      BEQ   PPOS    Oui, Alors Pitch
* Positif
      CMPB  ##10    Non, Alors le
* Pitch est-il egal a 10 en hexa ?
      BEQ   SUIT2   Oui, Alors son non
* voise
      TST   MPAR    La variation de
* la duree doit-elle etre Positive?
      BEQ   INCN    Oui.
      BITB  ##08    Non, Alors le Pitch
* peut-il etre multiplie Par 2 sans de-
* Passer la limite (-15 ou 11 en hexa)

```

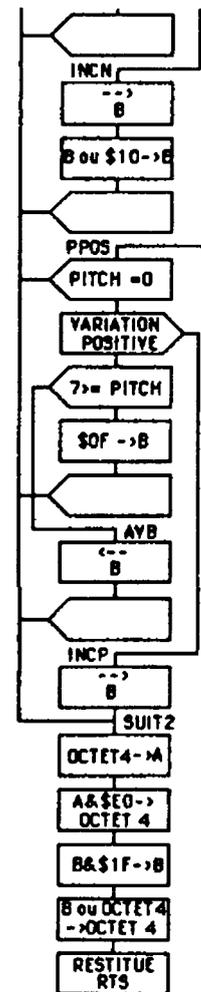
#### Organigramme



```

        BEQ     AVF     Oui
        LDB     #$11   Non.Alors on force
* le Pitch a -15T=$11
        BRA     SUIT2
AVF     ASLB                    Le Pitch est multi-
* Plie Par 2
        BRA     SUIT2
INCN   LSRB                    Le Pitch est divise
* Par deux
* tif
        ORB     #$10   Pour forcer en nega
        BRA     SUIT2
PPOS   TSTB                    Le Pitch est -11
* egal a zero?
        BEQ     SUIT2   Oui,
        TST     MPAR    La variation de
* la duree doit-elle etre positive?
        BEQ     INCP    Oui,
        BIT     #$08   Non.Alors le Pitch
* Peut-il etre multiplie Par deux sans
* depasser la limite (+15T=$0F)?
        BEQ     AVB     Oui,
        LDB     #$0F   Non.Alors on force
* le Picht a +15T=0F en hexa.
        BRA     SUIT2
AVB     LSLB                    Le Pichth est multi-
* Plie Par deux
        BRA     SUIT2
INCP   LSRB                    Le Pichth est divise
* Par deux
SUIT2  PULS   A                Restitue l'octet
* d'origine
        ANDA   #$E0   Masque le Pichth..
        STA    ,X     .. et sauve
        ANDB  #$1F   Masque au dela du
* format du Pichth
        ORB   ,X     Ou logique pour
* inserer la nouvelle valeur du Pichth
* dans l'octet
        STB   ,X
        PULS  D,PC   Restitue et RTS

```



## HAUA:

Le tronçon du programme principal étiqueté HAUA modifie la “hauteur avant” de la trame pointée par le curseur A. Le registre BLOQ permet d’inhiber l’action de HAUA. Si le curseur pointe sur la première trame du mot, la variation de la “hauteur avant” du curseur A est reportée sur la hauteur initiale de l’expression.

Seule est autorisée la variation de “hauteur avant” sur le curseur A, mais cette modification est recopiée sur le curseur B afin de conserver la même allure à la courbe hauteur/temps et une similitude entre la hauteur affichée et la hauteur réelle calculée par le MEA 8000 à chaque trame, en fonction du P.I. et de la durée de la trame. A l’issue du sous-programme HAUA, l’expression est prononcée en totalité si le curseur A pointe sur la première trame et partiellement (du curseur A au curseur B) si le curseur A pointe sur une trame différente.

## Registres et drapeaux utilisés ou modifiés

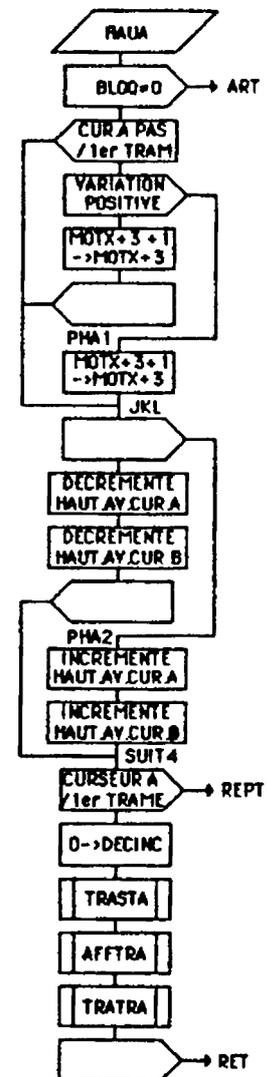
BLOQ-NUTRA-MPAR-MOTX-TABT1-TABT2-DECINC

### Listing source

```

HAUA  TST  BLOQ  Les commandes de
* variation de Parametre sont-elles
* bloquées?
      LBNE  ART  Oui. Alors Pas d'
* action
      LDD  #3031  Le curseur A Point-
* te-t-il sur la Première trame du mot?
      CMPD  NUTRA+9
      BNE  JKL  Non. Alors on ne
* modifie Pas la hauteur initiale
      TST  MPAR  Oui. Quel est le
* sens de la variation?
      BEQ  PHA1  Positif
      DEC  MOTX+3  Negatif. On decrem-
* te la hauteur initiale du mot.
      BRA  JKL
PHA1  INC  MOTX+3  On incremente la
* hauteur initiale du mot
JKL  TST  MPAR  Quel le sens de la
* variation demandee?
      BEQ  PHA2  Positif
      DEC  TABT1+2  Negatif. Alors on
* decremente les "hauteurs avant" des
* trames Pointees Par les curseurs A et B
      DEC  TABT2+2
      BRA  SUIT4
PHA2  INC  TABT1+2  On incremente les
* "hauteurs avant" des trames Pointees
* Par les curseur A et B
      INC  TABT2+2
SUIT4  CMPD  NUTRA+9  Le curseur A est-
* il sur la Première trame du mot
      BEQ  REPT  Oui. Alors on bran
* che a REPT pour Prononcer le mot en
* totalite
      CLR  DECINC  Non. Les trames ne
* doivent Pas etre incrementees
      JSR  TRASTA  Transcription
      JSR  AFFTRA  Affiche les trames
      JSR  TRATRA  Prononce la Portion
* du mot compris entre les deux curseurs
      LBRA  RET
  
```

### Organigramme



## PICA-PICB:

PICA et PICB sont les deux parties du programme principal qui permettent de modifier le P.I. de la trame pointée par l'un des deux curseurs. La trame pointée est passée par le registre PTA ou PTB en fonction du curseur spécifié. Si le drapeau BLOQ est positionné à une valeur autre que zéro, les commandes de variation du P.I. sont inhibées.

## Registres et drapeaux utilisés ou modifiés

PICHB: Drapeau utilisé dans MBWP pour indiquer la section du programme appelante (PICA ou PICB).

PICHB = 0 → PICA      PICHB ≠ 0 → PICB

## BLOQ-PTA-PTBT-TABT1-TABT2-DECINC

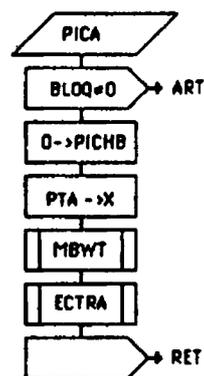
### PICA

#### Listing source

```

PICA  TST    BLOQ  Les commandes de
* variation de Parametres sont-elles
* bloquées
      LBNE   ART    Oui. Alors Pas d'
* action
      CLR    PICHB Pour indiquer que
* l'on s'adresse au Pitch A
      LDX    PTA    Pointeur A dans le
* registre X
      JSR    MBWP  Variation du Pitch
      JSR    ECTRA Affichage de la
* trame A
      LBRA   RET
  
```

#### Organigramme



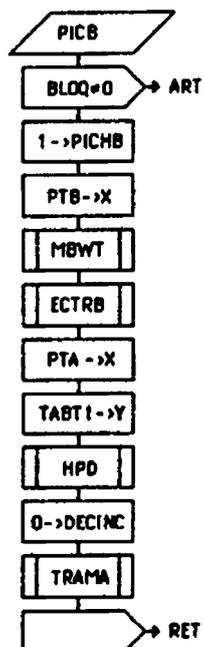
### PICB

#### Listing source

```

PICB  TST    BLOQ  Les commandes de
* variation de Parametres sont-elles
* bloquées
      LBNE   ART    Oui. Alors Pas d'
* action
      LDX    PTB    Pointeur B dans X
      CLR    PICHE  1 dans PICHE Pour
* indiquer que l'on s'adresse au Pitch B
      INC    PICHE
      JSR    MBWP  Variation du Pitch
      JSR    ECTRB Affiche la trame B
      LDX    PTA    Pointeur A dans X
      LDY    #TABT1
      JSR    HPD    Transcription
      CLR    DECINC Pas de modification
* du no de trame
      JSR    TRAMA  Affiche la trame A
      LBRA   ART
  
```

#### Organigramme

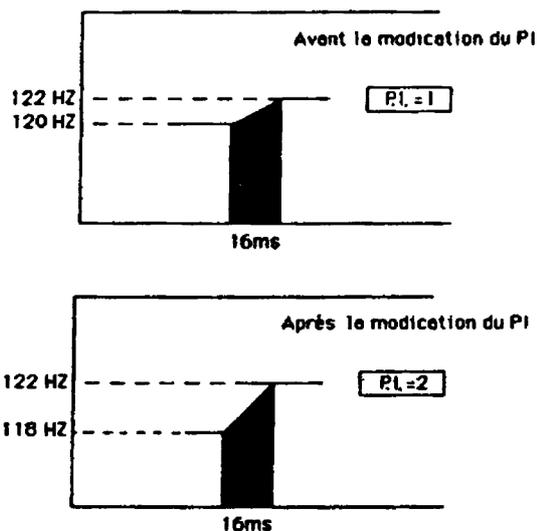


## MBWP :

MBWP est le sous-programme qui modifie la valeur du P.I. et ajuste la "hauteur avant" de la trame en tenant compte de la durée de celle-ci afin de conserver la même hauteur à la sortie de la trame. Cette façon d'agir permet de ne pas modifier la prononciation des trames qui suivent la trame pointée. Nous avons vu précédemment que la "hauteur après" la trame (Hap) était fonction de trois paramètres: la "hauteur avant" la trame (Hav), le P.I. et la durée de la trame.

$$Hap = Hav + (P.I. \cdot \text{durée})$$

Si l'on souhaite conserver Hap constant, pour une variation du P.I. d'une unité, il faut faire varier Hav d'une quantité égale à la durée mais en sens opposé à la variation du P.I. Ainsi une variation du P.I. positive va se traduire par une augmentation de l'incrément du pitch mais aussi par une diminution de la hauteur avant la trame, si le curseur spécifié pointe sur une trame autre que la première trame de l'expression, mais également une diminution de la hauteur initiale, si le curseur pointe sur la première trame de l'expression. Prenons un exemple.



Si l'on souhaite que la hauteur avant la trame ou la hauteur initiale soit réellement à 120 Hz, il suffit de modifier les paramètres appropriés dans le sens souhaité.

Un cas particulier apparaît quand la durée de la trame vaut 8 ms. En effet, ce cas conduit à une variation de hauteur de 1 Hz. Or, l'incrément minimum de la hauteur est de 2 Hz. Cet état nous a conduit à introduire une retenue (RTNU) qui est positionnée à 1 dans ce cas. Cette retenue est prise en compte dans la procédure de modification du P.I. suivante afin de rétablir la hauteur de fin de trame précédente.

## Registres et drapeaux utilisés ou modifiés

RTNU: Registre positionné à 1 pour indiquer que la variation de hauteur doit être incrémentée d'une unité.

VRPI: Suite de constantes qui spécifie la valeur binaire qu'il convient d'ajouter ou de retrancher à la "hauteur avant" de la trame, en fonction de la durée, pour obtenir une hauteur constante à la sortie de la trame.

VRPI           FCB           \$00,\$01,\$02,\$04

MPAR-MOTX-TABT1-TABT2-PICHB-PTB-NUTRA

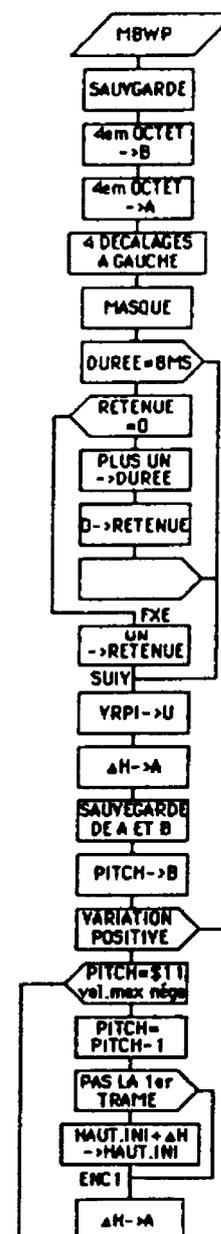
### Listing source

```

MBWP   PSHS   U,B       Sauvegarde
         LDB   3,X       4em octet de la
* trame dans B et...
         TFR   B,A       ...dans A
         ROLA           4 decalages circulaires
* a gauche pour cadrer la duree a droite
* de l'octet
         ROLA
         ROLA
         ANDA   #$03     Masque pour ne
* conserver que la duree dans l'octet
         TSTA           La duree est-elle
* egale a 8ms
         BNE   SUIY     Non.
         TST   RTNU     Oui, Alors la retenue
* est-elle egale a zero?
         BEQ   FXE     Oui.
         INCR           Non, Alors on incre
* mente le registre A
         CLR   RTNU     Clear la retenue
         BRA   SUIY
FXE     INC    RTNU     1 dans la retenue
SUIY    LDU   #$VRPI   Charge le Pointeur
* U avec du Premier octet de la table de
* variation de hauteur
         LDA   A,U     Variation de hauteur
* dans A
         PSHS   B       Sauve le 4em octet
         PSHS   A       sauve la variation
* de hauteur
         ANDB   #$1F     Masque pour ne con
* server que le Pitch dans B
         TST   MPAR     La sens de la
* variation demandee est -elle Positive?
         BEQ   PICP     Oui.
         CMPB   #$11     Non, Alors le Pitch
* est-il egal a la valeur negative max ?
         BEQ   TERM     Oui, Alors Pas de
* modification de Pitch Possible
         SUBB   #01     Non, Alors moins un
* sur le Pitch
         LDU   #$3031   La trame Pointee
* est-elle la Premiere trame?

```

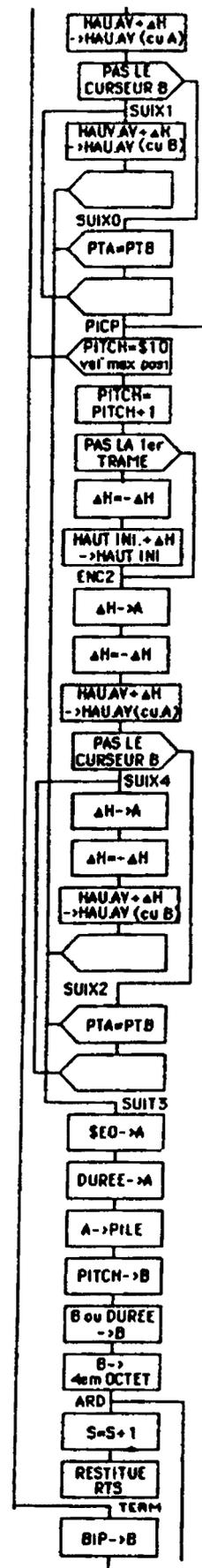
### Organigramme



```

        CMPU   NUTRA+9
        BNE   ENC1   Non.
        ADDA  MOTX+3  Oui. Alors on rePor
* te la variation de hauteur sur la hau-
* teur initiale du mot
        STA   MOTX+3
ENC1   PULS   A      Restitue la varia-
* tion de hauteur
        ADDA  TABT1+2 Modification de la
* "hauteur avant" de la trame A
        STA   TABT1+2
        TST   PICHB  A-t-on Pointe sur
* le Pitch du curseur B
        BEQ   SUIX0  Non.
SUIX1  LDA   -1,S    Oui. Restitue la va
* riation de hauteur dans A...
        ADDA  TABT2+2 ..et modifie la
* "hauteur avant" de la trame B
        STA   TABT2+2
        BRA   SUIX3
SUIX0  CMPX   PTB     Les curseurs
* Pointent-ils sur la meme trame?
        BNE   SUIX3  Non.
        BRA   SUIX1  Oui. Alors il faut
* modifier aussi, la "hauteur avant" de
* du curseur B. Pour etre coherent
PICP   CMPB   #$10   Le Pitch est-il
* egal a la valeur Positive maximum.
        BEQ   TERM   Oui. Alors Pas de
* modification Possible
        ADDB  #01    Plus un sur le
* Pitch
        LDU   #$3031  La trame Pointee
* est -elle la Premiere trame
        CMPU  NUTRA+9
        BNE  ENC2    Non.
        NEGA  Inverse la varia-
* tion de hauteur et...
        ADDA  MOTX+3  rePorte cette va
* riation sur la hauteur initiale du mot
        STA   MOTX+3
ENC2   PULS   A      Restitue la varia
* tion de hauteur
        NEGA  Inverse et..
        ADDA  TABT1+2 .. rePorte sur la
* "hauteur avant" du curseur A
        STA   TABT1+2
        TST   PICHB  A-t-on Pointe sur
* le Pitch du curseur B?
        BEQ   SUIX2  Non.
SUIX4  LDA   -1,S    Oui. Restitue la
* variation de hauteur
        NEGA  Inverse et..
        ADDA  TABT2+2 ..modifie la "hau-
* teur avant" du curseur B
        STA   TABT2+2
        BRA   SUIX3
SUIX2  CMPX   PTB     Les curseurs A et
* B Pointent-ils sur la meme trame?
        BNE  SUIX3  Non.
        BRA  SUIX4  Oui. Alors il faut
* modifier egalement la "hauteur avant"
* du curseur B

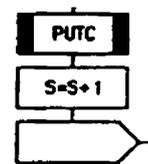
```



```

SUIT3 LDA    ##E0 Constante de masquage
      ANDA   ,S    Recupere le 4em octet
      STA    ,S    Duree dans la Pile
      ANDB  ##1F   Pitch dans B
      ORB   ,S    Reforme le 4em oc-
* tet avec le Pitch et la duree..
      STB   3,X    .. et le rePlace
ARD   LEAS  1,S    RePositionne le
* Pointeur de Pile
      PULS  U,B,PC Restitue et RTS
TERM  LDB   ##07   Un BIP
      JSR   PUTC
      LEAS  1,S    RePlace la Pile
      BRA   ARD

```



# Chapitre 7

---

## NORMAPHON

NORMAPHON est un logiciel utilitaire fonctionnant sous BASIC permettant de générer des tables de codes vocaux normalisées semblables à celle décrite en page 38 . NORMAPHON est la suite logique de PHONETRAM puisqu'il utilise les fichiers de mots issus de PHONETRAM et les réunit en une table composée des codes vocaux de chaque mot ou expression, dans l'ordre de leur lecture , et précédée de la table des adresses relatives. Ces tables pourront être, par la suite, utilisées dans toutes les applications utilisant le synthétiseur MEA 8000.

NORMAPHON est d'un emploi relativement simple. Il peut lire ou écrire des fichiers à partir ou depuis la disquette<sup>1</sup> ou bien du lecteur de cassettes. Les fichiers des mots ou expressions lus doivent impérativement se terminer par le suffixe .SYT, suffixe généré par PHONETRAM.

A la restitution des tables, les fichiers portent le suffixe .PAR. La table des adresses relatives est implantée (sur TO7) à partir de l'adresse \$A000 tandis que les codes vocaux commencent à l'adresse \$A080, soit 128 octets réservés pour l'adressage relatif. Pour des raisons de présentation à l'écran, NORMAPHON ne peut admettre plus de 50 mots ou expressions de longueur indifférente. La profondeur de table n'étant pas limitée par le logiciel, l'utilisateur devra surveiller l'adresse d'implantation (à gauche et en haut de l'écran) du mot suivant, afin de ne pas tenter une écriture dans une zone réservée (exemple : moniteur sur TO7). La longueur totale de la table (adresses relatives plus codes vocaux) est spécifiée en haut et à droite de l'écran.

---

<sup>1</sup> *Disquette 0 uniquement.*

En phase de restitution de la table normalisée, le logiciel sauvegarde, sur le périphérique assigné, la totalité de la table comprise entre l'adresse \$A000 et la dernière adresse d'implantation des codes vocaux. Le périphérique assigné ne pouvant être modifié au cours de l'utilisation de NORMAPHON, la table doit être sauvegardée sur le périphérique qui a fourni les mots ou expressions la composant. Notez qu'avec le lecteur 0 un changement de disquettes est toujours possible.

# Utilisation de NORMAPHON

---

Nous partons de l'hypothèse que le logiciel est chargé en mémoire. Le lancement est obtenu par un RUN classique. L'écran apparaît. Vous devez alors choisir, à l'aide du crayon optique, avec quel périphérique vous souhaitez travailler. Votre choix se limite à deux possibilités :

1— Le lecteur de disquettes 0 **[D]** (les autres lecteurs 1, 2, 3 ne sont pas admis par le logiciel).

2— Le lecteur de cassettes LEP **[K]**.

Avant la sélection du périphérique, les pictogrammes sont représentés sur fond blanc. Après la sélection, le pictogramme représentant le périphérique assigné est représenté en couleurs inversées. A l'issue de cette phase, le périphérique choisi ne peut plus être modifié.

Le logiciel vous invite alors (à gauche et en bas de l'écran) à entrer sur la zone de saisie le nom du mot ou de l'expression que vous souhaitez voir apparaître dans la table. Les noms doivent avoir au maximum 8 caractères. La touche **[ENTREE]** valide la commande. A l'issue de cette commande, le nom du fichier est complété à 8 caractères par des espaces.

Le suffixe .SYT est ajouté au nom du fichier. Le nom du mot ou expression, précédé du rang qu'il occupe dans la table, est affiché sur l'écran. Un nouveau mot peut alors être introduit dans la table, puis un autre, etc. jusqu'à concurrence de 50.

Une pression sur la touche RAZ permet de quitter la phase de concaténation des expressions et de passer en phase de sauvegarde de la table. Le logiciel vous invite à entrer le nom du fichier de sauvegarde de la table. Ce nom ne doit pas dépasser 8 caractères et doit être exclusivement composé de lettres majuscules et de chiffres. La touche **[ENTREE]** termine la commande. Le fichier de la table, suffixé .PAR, est sauvé sur le périphérique assigné précédemment. Les numéros des erreurs potentielles sont affichés en rouge dans la fenêtre gauche en bas de l'écran. Pour connaître leur signification, il convient de se reporter au manuel de référence du BASIC.

## Listing

```
5 CONSUI F 0,24:SCREEN6,6,6
8 CLEAR,&HA000:CLS
10 ON ERROR GOTO 800
11 LOCATE1,22:COLOR6,6:ATTRB0,0:PRINT " ";LOCATE1,23:
PRINT " ";
12 ATTRB1,1:COLOR4,3:LOCATE 12,1:PRINT "NORMAPHON":CONSOLE,,
,2
14 X1=5:X2=5:Y1=7:Y2=7
15 GOSUB 215
17 CONSOLE 4,20,0:SCREEN2,0,6:LOCATE10,22:ATTRB0,0
19 PRINT"UTILISER LE CRAYON";LOCATE10,23:PRINT" POUR CHOISI
R --> ";
20 INPUTPEN X,Y
22 IF Y>168 THEN GOTO 23 ELSE GOTO 20
23 IF Y<183 THEN GOTO 24 ELSE GOTO 20
24 IF X>248 THEN GOTO 25 ELSE GOTO 20
25 IF X<296 THEN GOTO 26 ELSE GOTO 20
26 IF X<268 THEN GOTO 30 ELSE GOTO 35
30 X1=7:X2=5:Y1=5:Y2=7
31 W$="O:"
32 GOSUB 215
33 GOTO 39
35 X1=5:X2=7:Y1=7:Y2=5
37 GOSUB 215
38 W$="CASS:"
39 Z=&HA080:K=&H0080:AD=&HA000:GOSUB 700
40 FOR M=0 TO 50
50 GOSUB 100
52 IF V=1 GOTO 80
53 GOSUB 300
55 X=2+(M MOD 3)*12:Y=4+(M@3)
56 C$=STR$(M+1):L=LEN(C$)
57 A$=MID$(C$,1,L):B$=A$+"-"+B$
58 IFM<9 THEN B$=" "+B$
60 ATTRB0,0:COLOR2,0:LOCATE X,Y:PRINT B$
70 NEXTM
80 GOSUB 500
85 ATTRB0,0:COLOR2,0:LOCATE 8,20:PRINT "ENREGISTREMENT TERMI
NE";
99 END
100 B$="":C$="":BOXF (72,174)-(252,193),-5
101 LOCATE1,22:COLOR2,0:ATTRB0,0:PRINT "ENTRER ";LOCATE1,23
:PRINT"UN MOT ";
102 C$=STR$(M+1):L=LEN(C$)
105 A$=MID$(C$,2,L)
106 IF M<9 THEN A$=" "+A$
107 ATTRB1,1:COLOR1,3:LOCATE 10,23:PRINT A$;
108 LOCATE 15,23:COLOR5,3:ATTRB1,1
109 FOR I=1 TO 8
110 A$=INPUT$(1)
111 IF ASC(A$)=&H02 THEN GOTO 201 ELSE V=0
112 IF ASC(A$)=&H0D GOTO 160
120 PRINT A$;
130 B$=B$+A$
150 NEXTI
155 GOTO 190
160 FOR A=I TO 8
```

```

170 A$=B$+" "
180 NEXTA
190 A$=B$+".SYT"
200 RETURN
201 V=1:GOTO 200
215 BOXF (248,174)-(300,193),-5
220 LOCATE 32,23:COLORX1,Y1:ATTRB1,1:PRINT "D"
221 LOCATE 34,23:COLOR4,4:PRINT " ";
222 LOCATE 35,23:COLORX2,Y2:PRINT "K";
223 LOCATE 37,23:COLOR4,4:ATTRB0,1:PRINT " "
224 LOCATE 38,23:COLOR6,6:ATTRB0,1:PRINT " "
225 RETURN
300 A$=W$+A$:LOCATE03,3:ATTRB0,0:COLOR0,1:
301 PRINT"ADRESSE:";:PRINTSTR$(Z);
302 LOCATE23,3:PRINT"LONGUEUR:";:PRINTSTR$(K);
305 C=&HB56E:LOADM A$, (Z-C)
310 I=256*(PEEK(Z)+(PEEK(Z+1)))
312 AR=Z-(&HA000):POKE AD,INT(AR/256)
313 POKE (AD+1),AR-INT(AR/256)*256
314 AD=AD+2
320 Z=Z+I:K=K+I
330 LOCATE03,3
340 PRINT"ADRESSE:";:PRINTSTR$(Z);
341 LOCATE23,3:PRINT"LONGUEUR:";:PRINTSTR$(K);
400 RETURN
500 BOXF (72,174)-(252,193),-5
501 LOCATE1,22:COLOR2,0:ATTRB0,0:PRINT "ENTRER ";:LOCATE1,23
:PRINT"FICHER";
508 LOCATE 15,23:COLOR1,2:ATTRB1,1
509 FOR I=1 TO 8
510 A$=INPUT$(1)
512 IF ASC(A$)=&HOD GOTO 560
520 PRINT A$;
530 B$=B$+A$
550 NEXTI
555 GOTO 590
560 FOR A=I TO 8
570 A$=B$+" "
580 NEXTA
590 A$=W$+B$+".PAR"
592 SAVEM A$,&HA000,Z,000
600 RETURN
700 FOR I=&HA000 TO &HA080
710 POKE I,&HFF
720 NEXTI
730 RETURN
800 LOCATE1,22:COLOR1,0:ATTRB0,0:PRINT "ERREUR ";:LOCATE1,23
:PRINT ERR ;
810 RESUME B5

```



# Annexes

---

Ce chapitre rassemble des informations utiles, voire indispensables, à la mise en œuvre du synthétiseur MEA 8000. Il se compose de 4 parties.

## Vocabulaire pour MEA 8000

---

Sont rassemblés sous cette rubrique les codes d'un vocabulaire relativement important, organisé en blocs de 1 à 4 Kilo-octets chacun. Pour chaque bloc, on trouvera :

- 1) Une liste donnant le numéro, l'adresse de début, l'adresse de fin et la description (texte) de chacun des éléments du vocabulaire. Ceci permettra d'en extraire aisément un ou plusieurs éléments pour composer un vocabulaire destiné à une application particulière.
- 2) Le listage hexadécimal du code correspondant à ce vocabulaire, par lignes de 16 octets dont la somme (décimale) est donnée à la fin de chaque ligne. Ceci permet de vérifier d'éventuelles erreurs d'entrée manuelle au niveau de chaque ligne au moyen d'un petit programme dont un exemple est donné (voir chapitre 5 "Applications", paragraphe ORIC).

Les différents blocs de vocabulaire sont les suivants :

— PHONEMES 4.2. On trouvera ici les éléments utilisés par la plupart des programmes de composition phonétique décrits dans ce livre. Ces phonèmes sont codés avec une durée de trame fixe de 16 ms, et occupent un peu moins de 1,5 Ko.

No	Ad.Deb	Ad.Fin	Texte
00	0054	0077	a
01	0078	009B	e
02	009C	00BF	i
03	00C0	00E3	O (ex. <u>eau</u> )
04	00E4	0107	u
05	0108	012B	è
06	012C	014F	é
07	01F8	021B	eu
08	021C	023F	ou
09	0150	0173	an
10	0174	0197	in
11	0198	01BB	on
12	01BC	01F7	oi
13	0240	025B	b
14	025C	0277	d
15	0278	029B	f
16	029C	02AF	g (ex. <u>gai</u> )
17	02B0	02C7	J (ex. <u>Joie</u> ; <u>gite</u> )
18	02C8	02DB	k (ex. <u>car</u> ; <u>quat</u> re)
19	02DC	02F3	l
20	02F4	0317	m
21	0318	032B	n
22	032C	0343	p
23	0344	0357	R
24	0358	036F	r (ex. <u>cour</u> )
25	0370	038F	S (ex. <u>sauce</u> )
26	0390	03A3	t
27	03A4	03C3	v
28	03C4	03D7	z (ex. <u>zoé</u> ; <u>asie</u> )
29	03D8	03F7	ch
30	03F8	042F	gn (ex. <u>bagne</u> )
31	0430	046F	ail
32	0470	04AF	eil
33	04B0	04EF	euil

34	04F0	052F	ien
35	0530	055E	oin
36	055C	057F	o (ex. bord)
37	05A0	05B7	s (très <u>b</u> ref)
38	0580	058B	silence 32ms
39	058C	059F	silence 64ms

PHONEMES 4.2

0000:	00 54 00 78 00 9C 00 C0 00 E4 01 08 01 2C 01 F8	1083
0010:	02 1C 01 50 01 74 01 98 01 BC 02 40 02 5C 02 78	852
0020:	02 9C 02 80 02 C8 02 DC 02 F4 03 18 03 2C 03 44	1151
0030:	03 58 03 70 03 90 03 A4 03 C4 03 DB 03 F8 04 30	1241
0040:	04 70 04 B0 04 F0 05 30 05 5C 05 A0 05 80 05 8C	1133
0050:	FF FF FF 00 00 24 3C 3C 86 B3 CD A0 86 B3 CD A0	2277
0060:	86 B2 D6 A0 86 B2 D6 A0 96 B2 CE A0 96 B2 CE A0	2760
0070:	97 B1 CD A0 97 B1 CD A0 00 24 3C 3C AF B3 85 20	2061
0080:	AF B3 85 20 AB B3 7E 20 AB B3 7E 20 AB B3 86 20	2051
0090:	AB B3 86 20 FF B3 85 20 FF B3 85 20 00 24 3C 3C	1870
00A0:	AF DA 5D 70 AF DA 5D 20 AF DA 5E 20 AF DA 5E 20	2074
00B0:	AF DA 66 20 AF DA 66 20 AF DA 65 20 AF DA 65 20	2106
00C0:	00 24 3C 3C C7 AC 84 A0 C7 AC 84 A0 C4 AB 85 20	2016
00D0:	C6 AB 85 20 C6 AB 85 20 C6 AB 85 20 D6 AB 84 A0	2279
00E0:	D6 AB 84 A0 00 24 3C 3C B7 B7 5D 20 B7 B7 5D 20	1815
00F0:	B7 B7 66 A0 B7 B7 66 A0 B7 B7 66 A0 B7 B7 66 A0	2512
0100:	B7 B7 5D 20 B7 B7 5D 20 00 24 3C 3C A6 B6 85 20	1699
0110:	A6 B6 85 20 B6 B6 BF 20 B6 B6 BF 20 B6 B6 BF 20	2314
0120:	B6 B6 B7 20 FA B7 A5 20 FA B7 A5 20 00 24 3C 3C	1995
0130:	B8 B8 85 A0 B8 B8 85 A0 B8 B8 87 20 B8 B8 87 20	2404
0140:	B8 B8 87 20 B8 B8 87 20 F8 B8 85 A0 F8 B8 85 A0	2532
0150:	00 24 3C 3C 52 CA 8D 20 52 CA 8D 20 97 CA 8D A0	1868
0160:	97 CA 8D A0 97 CA 8D A0 97 CA 8D A0 83 CA 8D 20	2660
0170:	83 CA 8D 20 00 24 3C 3C 66 B4 8D A0 66 B4 8D A0	1972
0180:	66 B5 BF 20 66 B5 BF 20 65 B5 C7 20 65 B5 C7 20	2038
0190:	61 B5 D5 A0 61 B5 D5 A0 00 24 3C 3C 82 D1 A5 20	1994
01A0:	82 D1 A5 A0 83 CF AE A0 C3 CE AE A0 C3 D1 9E A0	2793
01B0:	D2 D3 96 20 92 D2 95 20 92 D2 95 20 00 3C 3C 3C	1857
01C0:	5A 84 9C 20 5A 84 9C 20 5A 84 9D 20 5A 84 9D 20	1642
01D0:	5A 8A B6 A0 5A 8A B6 A0 87 AD C6 A0 87 AD C6 A0	2472
01E0:	47 B1 CE A0 47 B1 CE A0 97 B2 C6 20 97 B2 C6 20	2346
01F0:	9B B1 C4 20 9B B1 C4 20 00 24 3C 3C D6 B3 6C A0	1937
0200:	D6 B3 6D 20 FA B4 66 20 FA B4 66 20 FA B4 66 20	2226
0210:	FA B4 66 20 F6 B4 65 20 EA B4 64 A0 00 24 3C 3C	1953
0220:	9B AD 6C A0 9B AD 6C A0 DB AE 6D 20 DB AE 6D 20	2260
0230:	DB AE 6D 20 DB AE 6D 20 9B AE 6C A0 9B AE 6C A0	2262
0240:	00 1C 3C 3C 80 D6 18 20 80 D6 1A A0 90 B2 13 A0	1575
0250:	A4 B2 13 A0 A5 B1 24 A0 EA 91 76 20 00 1C 3C 3C	1736
0260:	80 B5 18 20 80 B5 1B A0 10 D5 04 A0 A0 D5 0C 20	1671
0270:	40 D2 2C A0 40 D9 5E 20 00 24 3C 3C 16 B7 FD B0	1675
0280:	16 B7 FE 30 16 B7 FE B0 16 B7 FE B0 16 B7 FE B0	2412
0290:	16 B7 FE B0 16 B7 FE 30 16 B7 FD B0 00 14 3C 3C	1916
02A0:	FF 97 60 20 FF 97 63 A0 FA 97 66 A0 A6 97 66 A0	2441
02B0:	00 18 3C 3C 1E BA 7C 30 1E BA 7D A0 1E BA 76 B0	1543
02C0:	1E BA 76 A0 1E BA 75 B0 00 14 3C 3C 09 97 88 30	1487
02D0:	00 97 8F B0 09 97 8E B0 67 96 86 20 00 18 3C 3C	1524
02E0:	74 B5 55 20 74 B5 55 20 74 B6 2E 20 74 B6 2E 20	1580
02F0:	28 B5 55 20 00 24 3C 3C 4C B4 5C A0 4C B4 5C A0	1510
0300:	4C B4 5D 20 4C B4 5D A0 4A B4 5E A0 4A B4 5E 20	1778
0310:	4A B4 5D A0 4A B4 5D 20 00 14 3C 3C 88 B4 55 20	1459
0320:	4B B4 5E 20 4C B4 5F 20 4A B4 5E 20 00 18 3C 3C	1285
0330:	C2 B9 38 30 C2 B9 38 30 11 B6 97 B0 16 B3 8E 30	1883
0340:	5B B2 8D A0 00 14 3C 3C 3A B3 84 20 36 B3 8D A0	1645
0350:	79 B2 96 20 97 B1 CD 20 00 18 3C 3C F5 B0 A4 A0	1935
0360:	85 8F 86 20 B5 8F 83 A0 54 B0 3C A0 54 B0 3A A0	2063
0370:	00 20 3C 3C 09 F3 8D 30 09 F3 8D 30 09 F3 8D B0	1747
0380:	09 F3 8D B0 09 F3 8D B0 09 F3 8D 30 09 F3 8D 30	2020
0390:	00 14 3C 3C B6 F6 90 30 B6 F6 94 30 B6 F6 96 30	2010
03A0:	9B B4 8D A0 00 20 3C 3C 6A B2 84 A0 6A B2 85 20	1813
03B0:	6A B2 86 20 6A B2 86 A0 6A B2 86 B0 6A B2 86 B0	2216
03C0:	6A B2 86 30 00 14 3C 3C 12 F5 76 20 12 F5 76 A0	1560
03D0:	12 F5 77 30 AA B4 76 A0 00 20 3C 3C 29 BA 8E 30	1627
03E0:	29 BA 8E B0 29 BA 8E B0 29 BA 8E B0 29 BA 8E B0	2180
03F0:	29 BA 8E B0 29 BA 8E 30 00 38 3C 3C 98 D7 5D A0	1758

0400:	98 D7 5D A0 98 D7 5D A0 99 D6 55 20 8E D6 5C 20	2204
0410:	8F D8 5E 20 8F D8 5E 20 AF D8 56 A0 AF D8 56 A0	2244
0420:	AF D8 56 A0 7F D8 5F 20 7F D8 5F 20 6B B7 6F A0	2138
0430:	00 40 3C 3C 57 B3 D0 A0 46 B4 E7 A0 46 B4 DF A0	2105
0440:	AB B4 CF A0 AB B4 CF A0 EA B5 C7 20 EA B5 C7 20	2728
0450:	FA B6 B6 A0 FA B6 B6 A0 BA B7 9E 20 BA B7 9E 20	2666
0460:	BA D8 7D 20 BA D8 7D 20 FA D0 73 20 FA D8 73 20	2344
0470:	00 40 3C 3C B8 B7 B6 20 B8 B7 B6 20 FB B7 AF 20	1993
0480:	FB B7 AF 20 FB B7 AF A0 FB B7 AF A0 B8 B7 9F A0	2868
0490:	B8 B7 9F A0 B4 B8 9F 20 BA D8 8F 20 BA D8 8F 20	2400
04A0:	BA D8 7E A0 BA D8 7E A0 FA D8 75 A0 FA D8 75 A0	2862
04B0:	00 40 3C 3C EB B3 AE 20 EB B3 AE 20 97 B4 A7 20	1954
04C0:	97 B4 A7 20 97 B4 A7 A0 57 B4 A7 A0 67 B6 9F 20	2258
04D0:	67 B6 9F 20 77 B6 B7 20 B8 B7 7E A0 EB B7 7E A0	2304
04E0:	AB B8 7E 20 AF D8 76 20 6B D8 76 20 BA B7 8C A0	2192
04F0:	00 40 3C 3C 7F D8 6D 20 7F D6 6E A0 7F D8 6E A0	1894
0500:	BF D7 87 20 FB D7 8F 20 B8 D7 9F A0 67 D6 BF A0	2603
0510:	67 D6 BF A0 67 D5 BF 20 67 D5 BF 20 61 D5 BE 20	2278
0520:	61 D5 BE 20 61 D5 BE 20 61 D4 C5 20 61 D4 C5 20	2140
0530:	00 2C 3C 3C B6 94 BD A0 B6 94 BE A0 B6 96 C7 20	1942
0540:	42 B4 D7 A0 53 B5 CF A0 62 B5 C7 20 62 B5 C6 A0	2399
0550:	62 B5 C6 20 62 B5 C5 A0 62 B5 C5 20 00 24 3C 3C	1809
0560:	AD AF A5 20 AD AF A5 20 AD AF A6 20 AD AF A6 20	2182
0570:	AD AD 96 20 AD AD 96 20 F9 AD 8D 20 F9 AD 8D 20	2246
0580:	00 0C 3C 3C AB B3 80 20 AB B3 80 20 00 14 3C 3C	1292
0590:	AB B3 80 20 AB B3 80 20 AB B3 80 20 AB B3 80 20	2040
05A0:	00 18 3C 3C 09 F3 8D 30 09 F3 8D 30 09 F3 8D 8D	1691
05B0:	09 F3 8D 30 09 F3 8D 30 75 30 69 B6 55 A0 AD B7	1935

— PHONEMES 3.3. Ces phonèmes, dont la liste et les numéros d'ordre sont les mêmes que pour la version 4.2, sont codés avec une trame variable, de façon à minimiser leur encombrement qui n'est que de 1 Ko. Ceci interdit toutefois leur utilisation avec les programmes faisant usage de la trame fixe (variation de vitesse, chant...).

N°	Ad.Deb	Ad.Fin	Texte
00	0274	0287	a
01	0288	029B	e
02	0380	0393	i
03	02D8	02EE	O (ex. <u>eau</u> )
04	02E0	02C3	u
05	02C4	02D7	è
06	03DC	03EF	é
07	0308	031B	eu
08	0368	0377	ou
09	033C	0353	an
10	0354	0367	in
11	02EC	0307	on
12	031C	033B	oi
13	03AC	03C3	b
14	0394	03AB	d
15	007C	008B	f
16	008C	00A3	g (ex. <u>gai</u> )
17	00A4	00BB	J (ex. <u>Joie</u> )
18	00BC	00D3	k (ex. <u>car</u> )
19	00D4	00E3	l
20	00E4	00F3	m
21	00F4	0107	n
22	0108	0123	p
23	0124	013B	R (ex. <u>rat</u> )
24	03F0	03FF	r (ex. <u>cour</u> )
25	013C	0147	S (ex. <u>sauce</u> )
26	03C4	03DB	t
27	0148	015B	v
28	015C	0167	z (ex. <u>zoé</u> )
29	0168	0173	ch
30	0174	0197	gn (ex. <u>bagne</u> )
31	0198	01C3	ail
32	01C4	01E7	eil
33	01E8	021B	euil

34	021C	0243	ien
35	0244	0263	oin
36	029C	02AF	o (ex. bord)
37	0378	037F	s (très bref)
38	0264	026B	silence 32ms
39	026C	0273	silence 64ms

PHONEMES 3.3

0000:	02 74 02 88 03 80 02 08 02 80 02 C4 03 DC 03 08	1215
0010:	03 68 03 3C 03 54 02 EC 03 1C 03 AC 03 94 00 7C	976
0020:	00 8C 00 A4 00 8C 00 04 00 E4 00 F4 01 08 01 24	1222
0030:	03 F0 01 3C 03 C4 01 48 01 5C 01 68 01 74 01 98	1044
0040:	01 C4 01 E8 02 1C 02 44 02 9C 03 78 02 64 02 6C	1023
0050:	FF	4080
0060:	FF	4080
0070:	FF 00 00 10 3C 3C	2941
0080:	16 87 FD D0 16 87 FE F0 16 87 FE F0 00 18 3C 3C	2208
0090:	FF 97 60 60 FF 97 63 80 FF 97 65 80 FA 97 66 80	2369
00A0:	A6 97 66 80 00 18 3C 3C 1E BA 78 30 1L BA 7D A0	1627
00B0:	1E BA 76 80 1F BA 76 A0 1E BA 76 80 00 18 3C 3C	1658
00C0:	09 97 88 70 0D 97 8F 90 0D 97 8F 10 09 97 8E 90	1628
00D0:	67 96 86 20 00 10 3C 3C 74 85 55 40 74 86 2E 40	1409
00E0:	28 85 56 C0 00 10 3C 3C 4C 84 5D 60 4A 84 5E E0	1652
00F0:	4A 84 5F A0 00 14 3C 3C 88 84 55 20 48 84 5E 20	1460
0100:	4C 84 5F 20 4A 84 5F A0 00 1C 3C 3C C2 89 38 70	1587
0110:	11 86 93 90 11 86 97 90 11 86 97 10 16 83 8E 10	1709
0120:	58 82 8D A0 00 18 3C 3C 3A 83 80 00 3A 83 84 00	1448
0130:	36 83 8D A0 79 82 96 20 97 81 CD 20 00 0C 3C 3C	1712
0140:	0E F3 8C 70 0E F3 8D 70 00 14 3C 3C 6A 82 85 40	1688
0150:	6A 82 86 C0 6A 82 86 D0 6A 82 86 80 00 0C 3C 3C	1962
0160:	12 F5 77 40 AA 84 76 C0 00 0C 3C 3C 29 8A 8E F0	1847
0170:	29 8A 8E F0 00 24 3C 3C 98 D7 5D C0 98 D7 5D A0	2037
0180:	99 D6 55 20 8E D6 5C 20 8F D8 5E 40 AF D8 56 E0	2182
0190:	7F D8 5F 40 68 87 6F C0 00 2C 3C 3C 57 83 D0 C0	1938
01A0:	46 84 E7 C0 46 84 DF C0 AB 84 CF C0 EA 85 C7 40	2766
01B0:	FA 86 86 C0 8A 87 9E 40 8A D8 7D 40 FA D8 73 40	2633
01C0:	FA D8 70 20 00 24 3C 3C 88 87 86 40 FB 87 AF 40	2055
01D0:	FB 87 AF C0 88 87 9F E0 86 88 9F 20 8A D8 8F 40	2720
01E0:	BA D8 7E C0 FA D8 75 C0 00 34 3C 3C EB 83 AE 40	2319
01F0:	97 84 A7 40 97 84 A7 C0 57 84 A7 C0 67 86 9F 40	2386
0200:	77 86 87 40 88 87 7E C0 EB 87 7E A0 AB 88 7E 20	2405
0210:	AB D8 76 20 68 D8 76 20 8A 87 8C C0 00 28 3C 3C	1871
0220:	7F D8 6D 40 7F D8 6E C0 8F D7 87 20 FB D7 8F 20	2375
0230:	88 D7 9F A0 67 D6 BF E0 67 D5 BF 40 61 D5 BE 40	2588
0240:	61 D4 C5 40 00 20 3C 3C 86 94 8D C0 86 96 C7 A0	2028
0250:	42 84 D7 A0 53 84 CF C0 53 85 CF E0 62 85 C7 40	2520
0260:	62 85 C5 40 00 08 3C 3C 86 83 C8 40 00 08 3C 3C	1373
0270:	96 82 C8 60 00 14 3C 3C 86 83 CD C0 86 82 D6 C0	2192
0280:	96 82 CE C0 97 81 CD C0 00 14 3C 3C AF 83 85 40	2142
0290:	AB 83 7E 40 AB 83 86 40 FF 83 85 40 00 14 3C 3C	1859
02A0:	AD AF A5 40 AD AF A6 40 AD AD 96 40 F9 AD 8D 40	2342
02B0:	00 14 3C 3C 87 87 5D 40 87 87 66 C0 87 87 66 C0	1983
02C0:	87 87 5D 40 00 14 3C 3C A6 86 85 40 86 86 8F 40	1875
02D0:	86 86 87 40 FA 87 A5 40 00 14 3C 3C C7 AC 84 C0	2108
02E0:	C6 AB 85 40 C6 AB 85 40 D6 AB 84 C0 00 1C 3C 3C	1989
02F0:	82 D1 A5 40 83 CF AE C0 C3 CE AF 40 C3 D1 9E C0	2666
0300:	D2 D3 96 40 92 D2 95 40 00 14 3C 3C D6 83 6E 40	1911
0310:	FA 84 67 60 F6 84 66 C0 EA 84 65 C0 00 20 3C 3C	2208
0320:	5A 84 9C 40 5A 84 9D 40 5A 8A 86 C0 87 AD C6 C0	2185
0330:	47 81 CE E0 97 82 C6 40 98 81 C4 40 00 18 3C 3C	2005
0340:	52 CA 8D 40 97 CA 8D C0 97 CA 8D C0 97 CA 8D C0	2739
0350:	83 CA 8D 40 00 14 3C 3C 66 84 8D C0 66 85 8F 40	1927
0360:	65 85 C7 60 61 85 D5 C0 00 10 3C 3C 98 AD 6D 40	1897
0370:	D8 AE 6D E0 98 AE 6D 40 00 08 3C 3C 0E F3 8C 70	1913
0380:	00 14 3C 3C AF DA 5D 40 AF DA 5E 40 AF DA 66 40	1800
0390:	AF DA 65 40 00 18 3C 3C 80 85 18 60 10 D5 04 A0	1524
03A0:	A0 D5 0C 20 40 D2 2C A0 40 D9 5E 20 00 18 3C 3C	1446
03B0:	80 D6 1A 60 90 82 13 20 A4 82 13 20 A5 81 24 20	1640
03C0:	EA 91 75 80 00 18 3C 3C 86 F6 90 70 86 F6 90 10	2088
03D0:	86 F6 96 90 86 F6 96 10 98 84 8D A0 00 14 3C 3C	2092
03E0:	88 88 85 C0 88 88 87 40 88 88 87 40 FB 88 85 C0	2596
03F0:	00 10 3C 3C F5 80 A5 40 85 8F 83 C0 54 80 39 C0	1942

— APHABET. Il s'agit ici de l'alphabet aéronautique international, utilisé en outre par les transmissions maritimes et les radio-amateurs (ALPHA TANGO appelle DELTA ZOULOU...). Il occupe environ 4 Ko.

No	Ad.Deb	Ad.Fin	Texte
00	00AA	0139	ALPHA
01	013A	01B5	BRAVO
02	01B6	025D	CHARLIE
03	025E	02ED	DELTA
04	02EE	0355	ECHO
05	0356	0441	FOX-TROTT
06	0442	04C1	GOLF
07	04C2	057D	HOTEL
08	057E	0619	INDIA
09	061A	06B1	JULIET
10	06B2	0721	KILO
11	0722	07A1	LIMA
12	07A2	0861	MICHAEL
13	0862	092D	NOVEMBER
14	092E	09D5	OSCAR
15	09D6	0A59	PAPA
16	0A5A	0AF1	QUEBEC
17	0AF2	0B91	ROMEO
18	0B92	0C0D	SIERRA
19	0C0E	0C89	TANGO
20	0C8A	0D59	UNIFORM
21	0D5A	0DFD	VICTOR
22	0DFE	0E81	WHISKY
23	003A	00A5	X-RAY
24	0E82	0F25	YANKEE
25	0F26	0F9D	ZOULOU
26	0F9E	0FA5	SILENCE 1
27	0FA6	0FAD	SILENCE2

## ALPHABET

0000:	00	AA	01	3A	01	B6	02	5E	02	EF	03	56	04	42	04	C2	1105
0010:	05	7E	06	1A	06	B2	07	22	07	A2	08	62	09	2E	09	D6	941
0020:	0A	5A	0A	F2	0E	92	0C	0F	0C	8A	0D	5A	0D	FE	00	3A	1113
0030:	0E	B2	0F	26	0F	9E	0F	A6	FF	00	00	6C	2B	34	51	99	1243
0040:	70	10	F4	DA	6D	A0	E9	DA	67	20	E5	BA	67	C0	98	BA	2493
0050:	5F	BF	8B	BA	3D	39	F0	DE	32	R0	8C	B6	62	30	04	BB	2067
0060:	64	B0	04	E7	66	30	14	B8	6D	30	11	F8	75	F0	11	F5	1858
0070:	86	F0	11	F5	8C	D0	24	B3	8D	30	A4	B3	95	A1	D8	B3	2436
0080:	9E	20	D8	B4	A7	40	E0	B5	A7	21	D0	B5	A7	3F	E4	D7	2484
0090:	9F	40	D6	D7	96	BF	64	D8	64	BF	02	98	E4	B0	30	B8	2392
00A0:	99	B0	70	D8	90	30	FF	FF	FF	FF	00	90	2A	37	00	99	2263
00B0:	F8	10	94	D1	CE	20	A8	D1	CF	20	E9	D2	CF	A1	E9	D3	2730
00C0:	CF	A0	D4	D3	CF	20	D4	D3	BF	EF	E8	D4	9F	20	D4	B5	2862
00D0:	8F	20	E5	B5	7E	A0	E1	B5	76	20	E2	B5	65	BF	E5	95	2504
00E0:	5D	3F	E0	95	6C	A0	44	B6	54	30	11	77	54	B0	14	D6	1809
00F0:	5C	B0	40	D9	5E	30	00	D8	65	B0	54	D7	65	B0	00	D8	1976
0100:	6D	30	00	B8	6D	30	00	D6	75	30	00	84	73	B0	90	D1	1701
0110:	7C	30	E8	D1	9D	20	F8	D1	AE	BD	E4	D1	BE	BE	A4	D1	2812
0120:	8E	BE	A4	D1	C6	BF	A4	D1	BE	BE	91	D2	D5	3F	93	B2	2851
0130:	DC	30	42	95	EA	B0	40	55	F8	30	00	7C	2B	29	00	99	1699
0140:	F0	10	F8	CD	94	20	A9	CD	94	A0	AB	CE	AD	21	B8	CE	2541
0150:	B0	A2	B9	CE	BE	A1	A8	CF	C7	24	94	CF	D7	A1	94	CF	2789
0160:	D7	A2	94	D0	CF	A1	94	D0	C7	20	94	D0	AF	21	94	D0	2608
0170:	8E	20	F9	D0	75	BE	A9	B0	65	BF	66	B0	3C	BC	11	B1	2295
0180:	2C	3F	15	AF	2D	3E	16	B0	3D	21	51	B0	56	24	95	AC	1402
0190:	6D	24	E1	AB	75	22	D0	AA	7D	3F	D1	8A	7C	BE	B0	C9	2248
01A0:	7C	BF	81	C6	74	3E	81	C5	6C	3E	90	A5	6B	BC	61	A3	2180
01B0:	6A	A0	00	75	68	20	00	AB	27	38	00	99	F8	18	25	BA	1422
01C0:	F3	B0	25	BA	E4	B0	25	BA	D5	30	25	BA	C5	B0	25	BA	2349
01D0:	B0	B0	25	BA	AE	30	25	BA	9E	30	25	BA	8D	B0	25	BA	2002
01E0:	86	30	25	B9	86	30	18	D6	B5	B0	94	D4	96	A0	94	B2	2129
01F0:	B7	20	AB	B2	BF	3F	E8	B2	C7	A0	E8	B1	C7	A0	AB	D0	2728
0200:	C7	BF	A4	D0	C7	3F	B5	AF	B6	3D	B9	AF	A5	BF	A9	AE	2682
0210:	9D	BF	AD	AD	94	30	59	AD	94	B0	4C	AD	A4	B0	5C	AE	2331
0220:	9D	21	98	B0	95	A0	94	B1	85	3F	90	B3	7D	3F	52	B4	2121
0230:	55	3F	92	B7	65	20	D8	D9	6E	21	D4	DA	66	20	E0	DB	2193
0240:	5D	3F	E4	DB	55	3F	E4	DB	55	20	E4	DC	55	3F	F1	BB	2339
0250:	54	3F	E8	BC	4C	3E	B2	B8	43	3E	44	D8	20	30	00	90	1662
0260:	26	33	B0	D7	20	18	04	B5	4B	B0	40	B7	7D	B0	F8	D7	1831
0270:	75	B0	F2	B7	B5	B0	E8	D7	96	A0	E8	D7	97	A0	E8	D7	2893
0280:	9F	A0	D8	D7	9F	A1	E8	D7	9F	A0	D0	B6	8F	28	E4	B6	2811
0290:	6E	20	E6	B5	6E	20	E7	B6	65	BE	E1	B7	66	20	E5	97	2321
02A0:	4D	BF	E1	B7	4D	3F	F1	B7	44	20	A0	B7	33	30	40	B8	2030
02B0:	39	30	40	B8	38	70	40	B8	38	70	00	B8	E3	30	01	98	1549
02C0:	CD	30	A2	D6	96	27	98	D5	A6	BA	82	D3	BE	BD	93	B3	2581
02D0:	BE	BE	E8	D3	C7	3F	AB	D2	C6	BF	AB	D1	BE	BE	58	D0	2905
02E0:	C5	BE	22	91	F3	B0	11	72	FA	B0	10	B2	E0	30	00	68	2112
02F0:	2B	35	44	D5	88	10	E4	DA	74	A0	C8	BA	76	20	E4	BA	2201
0300:	77	20	F0	DA	77	20	F4	DA	77	A0	E4	DB	6F	20	E0	DB	2534
0310:	5F	20	E0	B8	45	BC	CC	BA	3B	B0	84	B5	59	B0	88	B4	2314
0320:	60	70	B8	B4	60	70	40	B9	E9	30	90	AF	B0	30	50	AC	2070
0330:	73	B0	E5	A9	5C	23	E6	A7	64	A0	E1	C7	64	BE	91	C5	2529
0340:	64	20	90	C4	64	3F	90	C4	63	BF	94	C5	5C	3E	95	C5	2110
0350:	53	3D	20	A0	00	30	00	EC	2C	35	00	99	F0	10	00	97	1277
0360:	D3	B0	1A	B7	C4	30	16	B6	BC	30	16	B6	B4	B0	16	B6	2044
0370:	AC	B0	16	B6	A4	B0	16	B6	95	30	16	B6	8D	30	AA	AF	2031
0380:	7D	20	FF	AF	8E	A0	EF	AF	9F	20	EA	AE	9F	A0	FA	AD	2644
0390:	8E	BF	B5	CC	65	20	75	AD	43	30	90	AF	9A	30	80	B0	2081
03A0:	91	B0	80	B0	90	30	18	B6	C4	B0	60	B7	A5	30	25	D7	2131
03B0:	6E	B0	15	DB	BE	30	05	D7	CD	B0	14	F7	C5	B0	14	D7	2237
03C0:	8D	B0	00	D8	D8	30	00	D8	D8	30	00	D8	D8	30	00	D8	1973
03D0:	D8	30	00	D8	D8	30	00	D8	D8	30	00	D8	D8	30	10	B8	1904
03E0:	AC	B0	00	B9	FE	30	20	D1	05	30	10	D0	44	B0	99	AD	1923
03F0:	9D	B0	E4	AD	A6	A0	A4	CE	B7	3C	A4	AF	B7	3E	E4	D0	2693

0400:	AF 20 18 D1 9F 20 D8 D2 97 3E F8 D3 74 A0 50 AF	2452
0410:	3A 30 40 D2 A2 30 90 B0 A0 30 90 B0 A0 30 90 B0	1966
0420:	A0 30 91 F0 A0 30 91 F0 A0 30 91 F0 A3 B0 91 F0	2503
0430:	A5 30 91 F0 A4 30 91 F0 A3 30 91 F0 A2 30 91 F0	2386
0440:	A0 30 00 80 2C 33 80 B6 30 10 20 B7 EC 20 64 B6	1570
0450:	75 20 C8 D7 75 20 D8 D3 7F 20 D8 D2 8E A0 DD D1	2456
0460:	9F A1 DD D1 9F A0 D9 D1 9F 20 EC B2 9F 20 A7 93	2605
0470:	8F BF A7 93 B6 A0 A7 94 7E EF 96 94 76 3F E1 74	2392
0480:	6D 3E 91 75 65 3E A0 76 6C 30 16 B7 68 B0 16 B7	1723
0490:	83 B0 56 E7 FB B0 16 B7 FC 30 16 B7 DC B0 16 B7	2314
04A0:	FC B0 16 B7 D5 30 16 B7 D5 30 16 B7 D4 B0 16 B7	2158
04B0:	D4 B0 16 B7 D4 B0 16 B7 D4 30 16 B7 EA 30 16 B7	2138
04C0:	C8 10 00 EC 2A 32 00 99 F8 10 F9 AC 95 20 E4 AB	1914
04D0:	95 A0 D0 AC 96 20 90 B0 9E A1 D0 AD BC A1 E4 AD	2560
04E0:	86 A0 E4 AE 7E A0 F9 D1 7F 20 C8 D3 76 3F 40 D3	2466
04F0:	3E B0 00 90 03 30 00 96 F0 30 00 96 F0 30 00 96	1456
0500:	F0 30 00 96 F0 30 00 96 F0 30 00 96 F0 30 00 96	1752
0510:	F0 30 00 96 F0 30 00 96 F0 30 14 D8 CE 30 E2 D7	2095
0520:	8E 3F E4 D7 97 A0 E8 D7 9F EC E8 D7 9F A0 F2 D7	2976
0530:	9F 20 E2 D7 9F A0 D4 D6 97 BF E8 D5 7E BE F5 B5	2906
0540:	6E 20 F6 B5 76 20 F2 B5 76 20 E2 B5 76 20 D2 B5	2240
0550:	7E 20 D6 B4 7E A0 D6 B4 7E 3F C6 B4 7E 20 C2 B3	2330
0560:	7E 20 C6 B3 76 20 D6 B4 75 A0 B4 D5 7E 20 22 96	2043
0570:	8C B0 21 96 68 B0 11 76 9B 30 00 B4 08 30 00 9C	1512
0580:	2B 35 00 99 F8 10 E2 D8 54 A0 CE BA 5D 20 E8 D8	2170
0590:	5E 20 E1 D8 55 A2 F0 B8 55 A1 F2 D8 4E 20 F8 D8	2528
05A0:	4E 3F AA B8 54 B0 88 B2 5C A1 B6 AC 54 A0 96 A9	2204
05B0:	4C BF B6 A7 4C 3F B4 C6 4B BE B0 C5 4B 3E B4 A6	2062
05C0:	4B BF B4 A9 5A B0 80 AE 6A 30 14 B2 52 30 44 D7	1900
05D0:	3A 30 B8 DA 43 B0 F8 DA 44 A0 B2 DA 55 22 F4 D8	2423
05E0:	5D A0 F8 D8 5E 20 F8 DA 6E A0 F8 DA 7F A0 E4 D9	2780
05F0:	8F A0 E8 D8 A7 3E D4 D6 B7 3F 88 D5 B7 3F 88 B3	2562
0600:	8E A0 93 B4 CE 3F 64 D3 A0 30 12 72 8C 30 42 74	1980
0610:	74 30 14 B1 53 30 00 AD 38 30 00 98 2C 35 00 99	1171
0620:	F8 10 1D B8 F8 B0 4B B8 FC 30 61 B8 FC B0 88 D8	2530
0630:	FC B0 A4 DB FD 30 1B B8 FD B0 64 B7 5D 20 B4 B7	2472
0640:	65 20 B4 B1 65 21 99 AE 5C A2 AB AC 5C A0 4B B0	1997
0650:	64 3C 52 B0 33 3F 9C B2 3E A0 06 73 25 20 94 BA	1609
0660:	4E 22 E9 BA 46 22 F0 D8 46 20 F4 D8 56 20 E4 D8	2224
0670:	66 A0 E4 DA 7F 20 E4 D9 B7 3F E8 D8 97 BE E8 D8	2747
0680:	9F BE E8 D7 9F 3E FC D7 96 3D B2 D7 6C BF 52 D8	2685
0690:	49 B0 90 D7 80 70 90 D7 80 70 90 D7 80 70 90 D7	2405
06A0:	80 50 01 F3 C0 30 41 F7 9C 30 61 F7 7A B0 60 D7	2161
06B0:	78 30 00 70 2F 41 00 99 F8 10 00 D8 E4 90 00 D9	1614
06C0:	DF 10 01 BA F6 30 4B FA 64 A0 E8 FB 45 A0 F1 D8	2474
06D0:	46 A0 F5 D8 47 20 F5 D8 47 20 E9 D8 47 3F A5 DA	2333
06E0:	4F 3F 99 B7 46 BE A9 B6 4E A0 EC D6 4F 3F AC B6	2273
06F0:	4F 3F AB B5 4E BE AB B4 4E 3F C6 B2 76 A0 D9 AE	2293
0700:	76 A0 9E AC 76 3E 9D CA 76 3E E8 C8 65 3E D1 C9	2332
0710:	64 BF A4 C7 5B BE A4 C7 52 B0 21 A5 01 30 20 A6	2001
0720:	00 10 00 80 27 2C 45 B7 58 10 D1 B7 5B A0 91 B8	1555
0730:	75 20 E1 B7 65 A0 E0 B7 65 22 D0 D7 55 A1 D4 D8	2457
0740:	45 21 D0 DA 5E 24 D4 D8 56 A2 E4 D8 46 21 E0 D8	2330
0750:	46 20 E4 FC 46 20 E4 D8 46 20 88 B7 5D 20 43 B9	1929
0760:	65 21 43 B9 6D 20 8C B3 55 A0 8C B2 4D BF 8C B2	1995
0770:	4D BE 9C B1 45 A0 E8 B1 86 3F AB D2 AF 3E BC D2	2448
0780:	BF 3F FC D2 C7 BE AB D2 C7 3E AB D2 CE BF AB D1	2896
0790:	C5 B0 AB D1 C5 3E 52 B2 CC B0 12 90 D8 B0 11 71	2349
07A0:	E8 30 00 C0 26 2D 00 99 F8 10 00 99 F8 30 00 99	1574
07B0:	F8 30 00 99 F8 30 00 99 F8 30 F8 B0 4A B0 6C B0	2152
07C0:	4C 20 5C AF 3D 20 B8 AF 95 20 B5 B0 B6 21 B4 B1	1937
07D0:	C6 A1 FD B2 C7 21 FD B3 BF A1 E8 B5 B7 A0 E8 B7	2977
07E0:	9F A0 D8 D8 8F 21 E4 DA 7F 20 F4 D8 6E 20 F4 D8	2600
07F0:	4D A0 F0 FB 44 BF 8D B8 33 30 B4 B5 39 B0 B4 B4	2269
0800:	49 30 B4 E4 4B 30 B4 E4 4B 30 B4 E4 4B 30 B4 E4	1729
0810:	4B 30 B4 E4 4B 30 10 DA FD B0 10 BA 15 B0 D8 B9	2239

0820:	64 21 D8 B0 70 20 E8 B7 8D BC EC B6 95 BF C8 B6	2574
0830:	9D BF D9 B4 95 3F E8 D4 B4 BF F5 B5 6C BF E2 B5	2856
0840:	7D 20 E2 B5 7D 3F A7 B5 7D 20 F2 B5 7C 3F E2 95	2226
0850:	7B BF E2 B5 73 B0 52 D6 42 B0 44 D7 59 B0 41 B8	2347
0860:	90 30 00 CC 27 2B 00 99 F8 10 41 AE 63 30 B1 AE	1584
0870:	63 A0 B6 AC 54 A0 B5 AB 55 21 BA AC 55 21 D4 B4	2051
0880:	76 24 E4 D3 BF A0 E8 D2 97 A1 F8 D1 97 A1 E8 CF	2858
0890:	97 20 E4 CF 97 20 E8 CF BE BF E4 CE 7F 3E AB CF	2570
08A0:	6D EF AB B0 6E 20 FC D1 7E A1 F8 D2 87 21 F8 D2	2618
08B0:	97 A0 F8 D3 97 A0 F8 D3 BF BF F8 B2 7E 3E E8 B2	2930
08C0:	5D BF E8 B2 4D 21 D8 D7 4D 21 D8 B2 4D 3F D0 B2	2260
08D0:	4C BF D8 B3 4D 3F D8 B3 4D 20 D8 B3 4D 20 CC B3	2193
08E0:	4C BF DC B3 44 BF DC B3 44 30 94 B4 7D B0 E8 B4	2481
08F0:	87 21 E8 B4 97 20 E4 D4 9F 3F E8 D5 9F 3F F8 D5	2553
0900:	9F A0 F4 D5 9F 20 E8 D5 9F 3F E8 D4 9F 20 E0 D4	2705
0910:	A6 A0 E4 D4 A6 A0 E4 D3 A6 3F B4 D2 B0 A0 5C D0	2703
0920:	B0 20 24 D0 03 B0 15 B0 03 30 14 AE 00 30 00 AB	1254
0930:	27 2F 00 99 F8 10 A4 CD BC A0 F9 CD 9E 20 F4 CE	2266
0940:	9E A0 E8 D0 9F 20 E8 B1 97 A0 E8 D2 BF 21 B8 D3	2634
0950:	7E 3F 40 B5 5D B0 40 D8 95 30 55 F7 A5 B0 05 F7	2105
0960:	B6 30 04 F8 76 B0 24 F8 06 B0 20 D8 00 30 20 D8	1738
0970:	00 30 20 D8 00 30 20 D8 00 30 20 D8 00 30 20 D8	1184
0980:	00 30 00 92 F1 30 10 B9 CF 30 50 D8 9E BF C4 B7	1963
0990:	7F 28 C4 D2 97 3C D4 B4 9F BE D4 B3 A7 A0 E4 B3	2650
09A0:	B7 BE E8 B3 B7 BF E8 B3 B7 BF AB D3 B7 20 B8 D3	2932
09B0:	BF 3F AB D3 BF 3F AB D3 BF 3F A4 D2 BE BF AB D2	2653
09C0:	C6 3F A4 D1 B0 20 58 D1 C4 A0 24 D0 AB A0 AA 92	2303
09D0:	4B 20 64 B1 90 20 00 B4 29 2F 00 99 F8 10 90 94	1489
09E0:	94 20 AB CF B6 20 A4 D0 C7 20 AB D0 CF BF A2 D0	2516
09F0:	CF 20 A1 D0 C7 20 E4 CF AF 3F 90 AE B4 BF 10 CE	2375
0AA0:	13 30 20 B0 72 30 20 B0 70 30 20 B0 70 30 20 B0	1381
0A10:	70 30 20 B0 70 30 20 B0 70 30 20 B0 70 30 20 B0	1472
0A20:	70 30 41 B2 9D A9 E8 D0 AF 20 E8 D0 BF BA AB D0	2569
0A30:	BF BE F8 D0 C7 A0 F8 D0 C7 3F AB D0 C7 BF EB D0	3120
0A40:	C7 20 AB D0 C6 BE 18 CE D5 30 82 97 F4 B0 B1 77	2435
0A50:	F3 B0 41 96 F9 B0 10 B1 F8 10 00 98 2A 34 00 99	1915
0A60:	F8 10 14 DA DE 30 04 B7 B0 90 DE 6C A0 E0 DE	2350
0A70:	66 20 E0 DA 76 20 E0 DA 7E A0 D4 B9 7F 3F E9 B8	2458
0A80:	77 20 DA B4 5D 3A D9 B2 2B 3F C4 B2 2A B0 D8 B2	2187
0A90:	32 30 D8 B3 32 30 A9 B4 33 B0 F9 B6 7E A5 E9 B6	2304
0AA0:	BF 20 E9 B7 97 A0 E9 B7 97 3F E9 B8 BF BF E9 B8	2700
0AB0:	BF 3E E9 B9 7F 3F EC B9 5D BE A0 B8 6A B0 90 B8	2474
0AC0:	69 B0 90 B8 68 30 90 B8 68 30 90 B8 68 30 90 B8	2049
0AD0:	68 30 90 B8 68 30 90 B8 68 30 90 B8 68 30 90 B8	1920
0AE0:	68 30 05 B8 B4 B0 24 DE C5 30 00 DA B3 B0 00 DA	1943
0AF0:	B0 10 00 A0 2A 2B 95 AE 60 10 E4 AD B2 A0 90 AB	1827
0B00:	B3 A0 52 A7 B3 A0 96 A9 B8 BF B2 A3 93 22 91 A7	2266
0B10:	9C 20 D9 AB 9D 20 9A AB 9D A4 C9 AA 96 22 C0 AD	2325
0B20:	9E A4 B5 AB 96 A2 B4 AD BF 21 C0 AF B6 A1 D4 AE	2467
0B30:	7E A0 DC AD 76 3F 5C B0 4E 3E 9C B1 46 21 B0 B2	2022
0B40:	45 A0 DC B2 45 A0 EC B5 76 A0 D8 B7 7F 3F D4 B8	2536
0B50:	7F A0 E8 D9 7F 20 E8 D9 7F 20 D4 D9 7F 20 E4 B9	2504
0B60:	7F BF E8 B8 7F 20 E8 B5 76 BF E9 B0 76 3F E8 AE	2611
0B70:	76 3F F4 AB 6D 3F E9 A9 65 3F E5 AB 64 BF 95 C7	2370
0B80:	64 BF 90 A6 6B BE A2 A7 62 B0 40 A3 91 30 54 AD	2178
0B90:	60 10 00 7C 29 36 09 F9 F8 10 09 F9 BC B0 59 F9	1813
0BA0:	AC B0 59 FA BC B0 19 FA 9C B0 69 FA AC B0 D4 DB	2792
0BB0:	5D A0 D4 DA 6E 20 E8 DA 7E A0 E8 D9 7F 3F E8 D9	2649
0BC0:	7F BF E9 B8 B7 BF E8 B7 BF 3F E8 D6 BE BF 98 B4	2793
0BD0:	BE 3E 68 B4 65 30 AA B2 B0 65 B1 B4 B0 AB B0	2104
0BE0:	A5 B0 AC B0 AD BF A9 B0 BF 20 AD B0 BF 3F AB B0	2568
0BF0:	C7 3F E8 D0 C7 3F E8 D0 C7 20 A4 D0 C6 BF 64 D0	2704
0C00:	CE 3F 18 LD DC B0 41 97 F3 30 00 F6 F6 30 00 7C	2125
0C10:	2C 37 00 99 F8 10 00 DA EF B0 D4 D3 BF A0 F4 B2	2280
0C20:	A7 A0 F5 B0 AF 3F F1 A7 9E BF B5 AD BE 20 AD AE	2610
0C30:	7C 20 4B B2 BF A0 4B B6 A6 A0 44 B4 AE A0 4B B0	2120

0C40:	AF BF 49 AB 76 20 5C AB 6D BF 4C A9 75 A0 1C AA	2042
0C50:	65 3E 1C AC 6C BF 23 AD 63 3D 20 AF 6D A0 B9 AF	1865
0C60:	5C A0 EC AF 65 24 F4 CD 6C A0 F4 CC 6C A0 F4 AC	2633
0C70:	6D 20 F4 AB 75 20 F1 CA 6C BF E1 A9 6C 3F F5 AB	2409
0C80:	5B BE 50 C8 52 30 00 C5 00 10 00 D0 27 2C 00 99	1348
0C90:	F8 10 CC B8 4B A0 F4 DB 4C 20 F8 DB 45 20 F4 DB	2473
0CA0:	3D 22 E4 DB 3D A0 E4 DB 3D A1 F4 DB 3D A1 E0 BA	2527
0CB0:	46 24 D9 E7 4F 24 C9 B6 57 23 B9 AE 5F 22 4A A4	1801
0CC0:	67 20 8A A5 6F 21 8A AC 67 21 8A B3 67 A0 B9 B6	1927
0CD0:	5F 3D 4C B9 76 A0 4C BA 6E BF 8D B4 4E BE D4 DA	2277
0CE0:	5F 3E D4 DA 5E A0 F5 BA 4E BE D4 B9 4D BF 94 B7	2520
0CF0:	3C B0 10 B6 FC B0 11 B7 FD 3D 41 B8 FD B0 45 B4	2290
0D00:	FD B0 10 D7 B6 30 41 B9 FE B0 00 D8 CC B0 E4 B0	2570
0D10:	7D 3D E9 B0 97 3F E9 B0 97 3F E8 B0 A7 3F E8 AF	2477
0D20:	A7 3F E8 CF A6 BF E4 AF AF 3F F4 AE AE 20 A9 AD	2633
0D30:	A6 BF A9 AC AE BF FD AC A6 3F A9 AC 9D BF ED AC	2815
0D40:	7D 3E 25 AE 25 A2 64 AE 44 A1 45 AC 44 20 B4 D8	1789
0D50:	3B 3F 16 B1 01 30 01 A9 C0 10 00 A4 27 38 00 D7	1222
0D60:	00 10 D0 B8 2A 30 B0 B9 1A 30 94 B9 4C 20 D0 BA	1720
0D70:	55 A0 90 DB 57 20 90 DB 4F A4 C8 BE 47 20 BC BA	2149
0D80:	34 BE B4 B7 2A 30 C4 BA 21 30 B8 B7 50 B0 01 B8	1870
0D90:	BE 30 01 B6 34 30 4B B6 08 30 4B B6 08 30 4B B6	1395
0DA0:	08 30 4B B6 08 30 4B B6 08 30 4B B6 08 30 05 D6	1205
0DB0:	7A 30 00 BA E5 30 E8 D6 7D BE D4 D5 BE B0 D8 D4	2578
0DC0:	96 BE E8 D3 9E A0 E8 D2 9E A0 FC D2 A6 A0 F8 D1	3106
0DD0:	A6 BF E8 D1 A6 BE EC D1 A6 3F F8 D0 A5 BF F8 D0	3096
0DE0:	A5 BF F8 D0 A5 BF E4 CF A5 BF E4 CF A5 3F E4 CE	3056
0DF0:	9C BF A4 CC 94 3E 71 AC 5A 3D 20 AC 00 10 00 B4	1700
0E00:	28 30 00 B8 B0 10 28 A0 7B 20 9C A2 B3 A0 59 A5	1589
0E10:	64 20 E9 AA 64 A3 F9 AE 66 22 EC B2 66 A1 C8 B9	2419
0E20:	67 A2 E0 DB 5F A1 F2 DB 56 20 E8 DA 4E 3F 09 B9	2328
0E30:	4D B0 21 BA 4D 30 64 FA 4E 30 14 FA 4E 30 24 F9	1754
0E40:	4E 30 04 F4 4C 30 04 B4 48 30 04 B4 48 70 00 B5	1351
0E50:	4C 50 04 DB 4F B0 04 DC 4E 30 E0 DC 56 3F E0 DB	2020
0E60:	5F 3E E4 DB 5F BE E4 DB 67 3F E4 DB 67 3F F0 DC	2575
0E70:	67 3E F4 DB 66 BE F0 DB 65 BE A4 DB 2C B0 B1 B7	2585
0E80:	D8 30 00 A4 2A 2D C4 BA 68 10 E0 B8 58 A0 F8 DB	2146
0E90:	5C A0 F4 DB 55 20 F4 F8 55 A0 FC F8 55 A1 F4 F8	2816
0EA0:	55 A1 F4 DB 66 A1 E4 DA 7F 23 D8 B7 97 A0 ED B4	2707
0EB0:	9F A0 EC B1 9F 20 EC CF 96 20 E8 AE 85 20 E8 AD	2524
0EC0:	74 BF 98 AE 64 A0 4C A7 74 BF 4C A8 7D 20 9C AB	2171
0ED0:	5C BF AC AB 54 3F 48 AA 32 B0 08 A1 6A 30 48 AE	1810
0EE0:	80 30 48 AE 80 30 48 AE 80 30 48 AE 80 30 48 AE	1688
0EF0:	80 30 48 AE 80 30 00 DB FC B0 04 DB E5 B0 08 D5	2094
0F00:	FD B0 D9 B8 4D 21 E5 B8 56 20 F0 DB 5D BE F0 DB	2678
0F10:	55 BF E0 DB 5D BF E1 DB 55 A0 E4 DB 4C BF E1 DC	2851
0F20:	38 3F 45 B3 D0 30 00 78 28 28 B0 D6 30 10 D2 DB	1658
0F30:	4C 20 EC DB 4C A0 E4 DB 45 B0 A8 DB 46 30 01 DB	2217
0F40:	26 B0 02 DB 26 B0 9D B8 57 28 94 B7 5F 25 94 B4	1905
0F50:	5F 20 48 CE 5E A1 48 AC 5E A0 48 C7 66 20 4C A5	1804
0F60:	66 20 98 A9 4D A0 58 A9 3C 3E B4 AC 3C 3F 58 B0	1762
0F70:	3D 3E A8 D1 4D A0 A8 D0 55 BF 94 B0 66 20 E8 AC	2254
0F80:	66 20 E9 C8 5D BF EC CA 5D BF E8 C8 54 BF A4 C6	2645
0F90:	54 3F 90 A6 64 3F 14 C4 28 30 24 C6 00 10 00 08	1185
0FA0:	2C 2C 16 B7 E8 50 00 08 2C 2C 16 B7 E8 70 00 00	1250
0FB0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FC0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FD0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FE0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FF0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0

— VOCA I, VOCA II, VOCA III. Ces trois blocs de 4 Ko environ chacun représentent à peu près 150 éléments de vocabulaire français, dont les chiffres permettent d'énoncer par composition tout nombre compris entre 0 et 999 999, les lettres de l'alphabet et une centaine de mots.

VOCA I

No	Ad.Deb	Ad.Fin	Texte
00	006A	0081	A
01	0082	008D	ABSENT
02	00BE	0121	ALARME
03	0122	015D	ALLUMER
04	015E	01B1	ANORMAL
05	01B2	01FD	APPEL
06	01FE	022D	ARRET
07	022E	0291	ATTENTION
08	0292	02E1	AU REVOIR
09	02E2	035D	AUTOMATIQUE
10	035E	03A1	BATTERIE
11	03A2	03C5	BIEN
12	03C6	0415	BONJOUR
13	0416	0441	BONNE
14	0442	04A5	CARBURANT
15	04A6	04FD	CEINTURE
16	04FE	0571	CODE D'ACCES
17	0572	05B5	COMMANDE
18	05B6	0619	COMPOSER
19	061A	067D	CONTROLER
20	067E	06ED	CORRECT
21	06EE	0769	CORRESPONDANT
22	076A	07AD	DANGER
23	07AE	07C9	DE
24	07CA	081D	DEMANDE
25	081E	083D	EAU
26	083E	0891	EGALE
27	0892	08AD	EN
28	08AE	0909	ENTRETIEN
29	090A	092D	ET
30	092E	0985	EXCUSER
31	0986	09B1	FAIRE
32	09B2	09F9	FAITES
33	09FA	0A51	FORMULER

34	0A52	0A99	HEURE
35	0A9A	0AED	ICI
36	0AEE	0B61	IMPOSSIBLE
37	0B62	0BED	INCORRECT
38	0BEE	0C5D	INFORMATION
39	0C5E	0C7D	L'
40	0C7E	0CA1	LA
41	0CA2	0CF5	LANTERNE
42	0CF6	0D2D	LE
43	0D2E	0D51	LES
44	0D52	0D99	MARCHE
45	0D9A	0DE1	MERCI
46	0DE2	0E31	MESURE
47	0E32	0E89	MINIMUM
48	0E8A	0EE1	MINUTE
49	0EE2	0F29	MOINS
50	0F2A	0F71	MOTEUR
51	0F72	0FA9	MILLE

VOLUME 1

0000:	00 6A 00 02 00 0E 01 22 01 5E 01 B2 01 FE 07 2E	1038
0010:	02 92 02 E2 03 5F 03 A2 03 C6 04 16 04 42 04 A6	1105
0020:	04 FE 05 72 05 B6 06 1A 06 7E 06 EE 07 6A 07 AC	1266
0030:	07 CA 08 1E 08 3E 08 92 08 AE 09 0A 09 2E 09 B6	870
0040:	09 B2 09 FA 0A 52 0A 9A 0A EE 0B 62 0B EE 0C 5E	1414
0050:	0C 7E 0C A2 0C F6 0D 2E 0D 52 0D 9A 0D E2 0E 32	1194
0060:	0E BA 0E E2 0F 2A 0F 72 FF 00 00 18 00 30 B4 B4	1265
0070:	08 00 B1 B3 D7 44 B1 93 CF 44 B4 92 CE DD B0 91	2517
0080:	00 5B 00 3C 00 30 A9 D5 80 20 F8 D4 97 26 B4 B4	1942
0090:	97 23 A8 B3 96 40 E0 B3 8A D5 D0 B4 92 38 1A D6	2331
00A0:	9D D0 75 D6 8D 70 25 D6 84 D0 25 D6 84 E0 A1 D4	2392
00E0:	7D 20 75 B1 96 4A 61 AE A4 FF 00 B0 B0 61 00 64	1866
00C0:	00 35 60 B5 F0 00 71 93 D6 42 71 93 DF 41 71 94	1919
00D0:	BF 42 B5 B4 97 40 B5 B4 76 C0 B4 B3 6E 71 B8 B3	2369
00E0:	7E A4 74 B4 A7 A0 B4 B4 CF C0 B1 B3 D7 40 B1 93	2631
0010:	CF 41 B4 92 CE C0 B0 91 C6 5F 73 90 9D 40 B2 AF	2443
0100:	84 DF 92 8D 73 C2 94 AE 63 A2 58 B3 64 40 94 B1	2290
0110:	5C 20 50 82 5C A0 B1 B0 75 3F A5 B3 5E DF A0 D3	2192
0120:	58 5C 00 3C 00 30 50 D6 78 00 FA B5 77 C6 FA 95	1849
0130:	6F 40 E2 D5 56 A1 A2 D6 5F 3D A2 B8 5F 41 A1 B7	2243
0140:	5E B0 D1 94 34 5E D9 92 34 5F AA 96 57 5F E8 B8	2217
0150:	77 DF C5 B9 55 FF 40 B6 42 E1 04 D7 48 40 00 54	2040
0160:	00 30 91 B6 60 20 61 B4 B6 24 F6 B4 96 C1 B0 B3	2074
0170:	96 DF D0 B1 53 C0 D1 B1 4C 21 B5 93 5D A0 B6 B2	2469
0180:	66 C0 B6 CF 66 40 75 CE 8D A0 A5 C0 74 D0 D6 AD	2554
0190:	64 40 D1 AE 3B BF D6 DA 34 3F 96 90 4D 3E 76 B1	2072
01A0:	A7 5D 76 B2 B6 DF F0 B2 73 FE D1 B3 5A 7F E5 B4	2762
01E0:	60 49 00 4C 00 30 B6 B5 8E 00 F2 B3 97 A7 F2 B3	1958
01C0:	97 C0 E0 B4 8C 00 D0 B4 8A C0 00 B4 79 5F 45 B4	2250
01D0:	8B B0 05 B5 28 10 16 B5 85 23 F6 95 87 3F F6 B6	1952
01E0:	8F DF F6 B6 8E DD 90 B6 7E 5E F2 B4 5C DE F2 B4	2861
01F0:	56 40 F2 B4 63 3F B6 B4 52 C3 AA B5 60 24 00 30	1904
0200:	00 43 20 D3 AF C0 60 B3 DF 7F 60 93 E6 BE 51 B2	2224
0210:	96 5D 65 B2 8E 3D 55 B3 95 B0 A5 B4 95 B0 B5 B4	2371
0220:	9E B0 AA B5 9E DD E9 97 8D 5D D0 B8 80 FF 00 64	2570
0230:	00 30 11 95 C0 00 66 B4 BE 20 76 B4 AF A1 76 B4	1842
0240:	9F C2 A5 B3 B5 5B E5 94 B4 37 F1 94 BA B9 91 96	2492
0250:	95 30 A1 95 B5 E0 A6 B0 97 CF 67 CE BF 41 A2 CE	2401
0260:	7E DF 56 D4 5E 5E 16 F5 5D 80 16 F7 55 F0 16 F9	2236
0270:	56 70 65 DA 56 AC E2 D6 6E C2 A2 D1 66 40 71 CE	2375
0280:	66 40 62 CD 6D C1 62 CC 7D 5F 52 CD 74 5C 52 CF	2077
0290:	78 59 00 50 00 30 A5 B0 90 00 E6 AE 96 3C E8 AC	1843
02A0:	97 5E E8 CC 96 BF EE CB 8E DE 96 CD 8D 5E 51 B2	2679
02B0:	85 A1 10 B3 7E 30 11 B2 7E 50 90 82 75 24 E8 AC	1946
02C0:	75 C1 BF B0 97 48 7A B2 B7 C0 79 B3 BF DF 74 B3	2574
02D0:	BF 41 60 D1 7E C2 60 B2 0C E1 20 D1 0C C3 30 D1	2097
02E0:	08 3C 00 7C 00 30 AD B3 70 00 E9 B3 87 42 E9 B2	1728
02F0:	7F 41 D0 B4 73 40 C1 B5 B1 DF 41 B6 8E 30 45 B5	2172
0300:	8F 90 9A B4 77 3F 9A B3 8F B0 96 B0 4D 41 5D 92	2175
0310:	26 40 B8 B3 A7 C0 EA B5 B7 5E E4 B4 4A DC 54 B5	2438
0320:	59 C0 04 B5 64 10 44 B6 6E B0 D9 B7 40 A2 ED B8	2178
0330:	4E 20 EA B9 55 DD FA B9 55 DE 05 B9 3A D0 05 B8	2222
0340:	40 50 09 B8 4F B0 09 B7 56 B0 09 B7 64 30 58 98	1629
0350:	6E 2B E8 96 7D C1 BE B6 85 C2 64 B5 B8 40 00 44	2104
0360:	00 30 E4 AE 38 00 E4 AE 3D 24 25 B1 76 3F B8 B2	1765
0370:	8F BF F7 B4 9F DE F0 B4 B5 C0 E1 B4 B3 41 E4 B4	2896
0380:	7A 42 D1 B4 B9 22 01 B6 B0 B0 B1 D4 7D D0 F1 D4	2423
0390:	76 46 D1 B9 56 E2 D5 DA 55 E1 E0 DA 4D DD D1 98	2736
03A0:	60 54 00 24 00 30 95 D7 38 20 FA D7 45 DB E5 B7	1881
03B0:	66 DE F5 B6 BE DE B6 B5 8E C0 A1 B5 75 C2 B0 D6	2807
03C0:	39 E3 41 B5 58 40 00 50 00 30 00 B5 70 00 40 B4	1347
03D0:	45 A0 E9 AD 2B A2 C0 AF 2C 23 E5 AA 7E 40 E6 AC	2277
03E0:	8E 5F B9 AE 7E 40 90 92 5E 5F 50 B4 56 50 40 B5	1936
03F0:	56 30 E8 B3 5E 3F EA 91 5E 41 AA AF 5E 63 E5 AF	2182

0400:	6E 62 F1 AE 66 44 A1 AD 5D C0 55 AF 52 F0 00 D5	2191
0410:	51 7A 41 B6 50 3A 00 2C 00 30 00 D6 28 00 80 B5	1244
0420:	2A E0 95 AF 28 3F E9 AE 2B C3 FF AF 76 A4 B8 80	2381
0430:	87 A3 F0 B2 7F 43 90 B1 34 C4 D0 B1 3C 91 B0 B3	2376
0440:	2B 31 00 64 00 30 D0 B6 58 00 45 B5 5E 30 05 B9	1297
0450:	65 2E E2 E7 66 A2 F6 B6 B7 A3 7A B3 AF DD A5 B0	2584
0460:	94 DE E6 AE 7D 5D 90 AD 3A 41 D5 AF 3A 21 E1 B0	2314
0470:	5A 23 A1 93 4C 21 F6 94 54 A4 FE 94 55 3E FE 94	2103
0480:	65 C1 FA B0 7D C0 F6 CE 75 20 97 CE 6C 21 A2 CC	2486
0490:	85 C0 62 CC 7D A7 72 CD B5 A2 72 97 7D C4 72 92	2374
04A0:	7A 31 72 92 78 31 00 58 00 30 1A F7 50 10 5A F7	1442
04B0:	64 D0 5A F7 64 90 5A F7 64 B0 EB B4 76 20 6F B4	2358
04C0:	8F 44 6F B4 87 5F E9 B4 4D 58 D8 B3 43 21 E4 B4	2216
04D0:	69 41 95 B4 51 30 55 B7 4B B0 07 B8 45 50 EB B8	1906
04E0:	4D DF EB 97 4E 7F E6 97 4C E1 D2 B7 55 62 EA B6	2565
04F0:	66 C4 D5 94 65 A3 E4 B2 65 24 70 CF 48 7B 00 74	2096
0500:	00 30 11 B3 68 00 31 B3 6D A9 A5 B3 7E A4 AA B2	1836
0510:	9F 43 AA B2 97 A1 DA B3 BF 24 A6 B3 B6 A3 9B B3	2438
0520:	5D B9 F7 B1 43 3F E5 B2 31 DA C0 B5 29 7F E5 B5	2456
0530:	21 7E 69 B5 55 3F FF B5 7E AF AB B5 Bf 28 AA B5	2216
0540:	8F 3F 97 B5 B7 3E AB 94 5D B9 56 D2 51 DE 06 F6	2183
0550:	55 30 06 F6 4D F0 06 F6 3D F0 16 B6 3D A6 A9 B5	2036
0560:	6E 20 99 B5 B6 BF EA B5 B7 8F AA B6 B7 5E E9 B5	2409
0570:	68 5E 00 44 00 30 05 95 B8 10 05 95 8F 90 AB 91	1377
0580:	96 A0 66 B0 9F C6 EA AE B5 40 98 AF 64 40 94 B3	2368
0590:	4C C0 76 CE 7E DF 76 CE BE C0 76 CE BE E0 76 CE	2613
05A0:	8E C1 FA CF 7C DD 90 CD 4A DF D1 D3 32 DE D4 D7	2902
05B0:	2C C0 D4 D6 30 5F 00 64 00 30 45 90 B8 10 45 90	1579
05C0:	8E 90 A5 BE 75 A0 C3 AD B6 CD B2 CB 96 C4 91 A9	2618
05D0:	85 DF 55 A6 B6 5D 60 A4 B2 41 B0 CD 7A 3F B2 99	2063
05E0:	76 B0 D1 71 6D 3D D4 A9 64 BF D5 CB 6E 41 E6 AD	2452
05F0:	6E 40 F1 AF 6E DF E5 D4 5E 5D D5 D6 6D C0 F1 B5	2701
0600:	5D C2 E1 97 5F 42 E5 97 6F A2 D5 97 67 40 E2 B7	2417
0610:	5E DD C4 98 5C 40 11 D6 F8 7A 00 64 00 31 E0 99	1946
0620:	48 00 80 B2 5D B0 10 AE 5D 04 60 AE 6C A2 C1 CC	1871
0630:	7E 41 B1 C8 B5 A2 41 C8 A6 42 44 CA A6 02 54 CD	2039
0640:	86 22 84 AF 54 BE 94 CB 3B 3C 40 D0 79 5D 12 B3	1902
0650:	6D B0 93 B1 66 30 A5 CF 74 BE 84 CB 74 DF C5 CA	2510
0660:	7D C2 B5 CC 76 41 A4 BF 75 C1 AB 92 5E 60 96 B5	2291
0670:	4F 5F EB B7 67 41 EA B8 67 42 A9 B7 68 5A 00 70	2005
0680:	00 30 80 B3 68 10 B0 B3 6D 10 91 93 65 90 18 92	1614
0690:	6C B0 39 94 75 2F F6 94 7D A5 B2 B2 96 41 B2 B0	2214
06A0:	96 5C 76 AF 8D 5A A6 D0 94 5C 96 B2 BC 3C A9 94	2225
06B0:	9E 40 A9 B6 96 5D A6 98 7D 5F CD 97 62 41 50 B9	2138
06C0:	70 60 09 98 75 30 11 99 73 B0 20 B9 79 30 08 97	1540
06D0:	80 70 01 B5 B4 30 74 D6 B4 30 A0 B5 72 B0 F5 95	2137
06E0:	7C 27 F5 B6 B4 44 B6 B6 B8 30 00 B6 B0 50 00 7C	1855
06F0:	00 30 E1 B4 7B 00 E1 B4 7D 00 E1 B4 7D 21 EA B3	2079
0700:	76 20 A6 B2 B7 2E A5 B0 97 5F 66 AE B6 5E A6 B2	2110
0710:	8F 41 E6 B4 8F BF B5 B5 7F BF F4 D5 6E 3F EA D6	2710
0720:	75 BE 19 D7 7E D0 96 D7 7B D0 18 D7 60 50 11 B2	2187
0730:	73 C2 A7 AC 74 BF D3 AC 7D BF B3 CC B0 A1 B3 CC	2626
0740:	7E 40 59 CD 66 40 DD AF 3C B0 E4 B1 33 DF E0 B2	2376
0750:	2A BF A1 B4 5D A2 FA B2 B7 42 A6 B1 97 40 75 AF	2308
0760:	86 E0 65 AE B5 DF 65 AE 60 7E 00 44 00 2E C0 B6	1974
0770:	40 00 B0 B5 57 29 AA B3 BF AB 6A B2 A7 C1 27 AF	2019
0780:	9F C2 15 CE 97 41 18 CD 97 DF 19 B1 BF 5F 09 B5	2029
0790:	87 5E 0A B7 B7 90 06 B7 7F D0 16 B5 77 B7 AA B6	2082
07A0:	7F DE DA B7 76 DF E9 B8 6D 40 D9 B8 6D 40 00 1C	2270
07B0:	00 30 99 AD 58 40 56 D4 5E 2F 9B B4 5E AF 95 B2	1896
07C0:	7E C5 96 B1 53 5A E6 B1 60 31 00 54 00 30 55 B3	1771
07D0:	48 00 B9 B4 66 A5 B5 B4 6F A2 B6 B4 77 DF E9 B2	2453
07E0:	44 DD 94 B2 4C DE 72 B1 6E 3E 72 B0 A6 5D 71 CF	2245
07F0:	A5 C0 71 CF A5 3F B2 CE A6 40 72 CE 9D DF B3 AE	2572
0800:	B5 40 F9 B0 4E 5F C0 B2 41 C0 D1 B3 63 72 F5 B4	2349
0810:	B5 21 F1 B3 B4 C1 F7 B5 B2 C0 00 B5 B0 46 00 70	2051

0820:	00 30 40 10 78 00 FE AE 66 06 FE AC 76 23 1E AF	1875
0830:	65 C1 1A 6C 65 10 E2 CA 64 DE F3 AF 60 3C 00 54	2361
0840:	00 36 5A 00 90 00 A1 E7 86 22 E2 BE 77 C1 E2 B9	2155
0850:	6F C1 D1 98 54 00 CF 95 3A 40 20 98 55 C1 E6 B9	2341
0860:	51 A6 A6 E7 97 C0 76 B5 E7 40 E2 B4 C2 41 E6 B3	2401
0870:	E7 40 B5 B3 66 C0 1E B3 76 60 F8 B3 6D A2 F0 B3	2635
0880:	65 3F 18 B3 5C BF 14 B4 98 BD D4 14 50 5D D4 E4	2519
0890:	48 1C 00 1C 00 31 63 CD 96 40 67 CD 96 2D 12 AC	1388
08A0:	8F 41 23 AC 76 A2 44 AF 74 5D 44 B4 5D DF 00 5C	1803
08B0:	00 30 BA 10 88 00 FA E0 BF C1 B2 AE 76 C1 A8 CE	2273
08C0:	5C C0 90 AD 53 22 A1 AE 7A 20 01 E0 71 30 18 F0	1823
08D0:	75 D0 25 D0 76 23 F1 B1 7F 40 F1 D5 66 C1 D0 D4	2501
08E0:	63 DC 91 B4 6E 50 49 D7 65 50 09 DF 5E 50 99 D8	2068
08F0:	5D 20 D2 B8 67 21 84 B6 6F 40 95 D4 5F 40 A4 D4	2040
0900:	76 A1 D0 B3 43 40 80 D5 40 5D 00 24 00 30 A5 B8	1728
0910:	80 00 A5 B8 7E 23 E4 98 74 A3 E5 B0 77 C0 F5 B6	2450
0920:	66 C0 E4 B8 65 3E D5 B7 5B 3C 61 B7 70 20 00 58	1928
0930:	00 30 96 B5 58 20 99 B5 6F 43 96 B8 57 41 C5 96	1844
0940:	54 40 45 75 56 21 51 96 56 B0 41 BA 4E B0 42 DA	1735
0950:	4E D0 02 D9 4D 50 02 99 4D B0 06 98 47 B0 16 98	1649
0960:	47 B7 96 98 47 3C 96 B8 47 5F 92 D8 3E DF 02 D8	2052
0970:	3D DF 9A B5 66 DF DA B6 76 DF 9A B8 77 40 AB B8	2561
0980:	7D C5 EA B7 70 24 00 2C 00 38 A5 B7 58 10 20 B6	1653
0990:	66 70 16 B5 6E 50 A2 B5 6D D0 EA B6 77 C0 FA B5	2425
09A0:	87 FD EE B3 BF DE A6 B1 BC 7F A6 D0 B6 20 A6 CF	2694
09B0:	68 46 00 48 00 49 55 B6 F8 10 29 B5 EE 50 09 B4	1579
09C0:	FD D0 05 B6 FE D0 B9 B4 87 C0 BA B5 97 C1 EA B6	2929
09D0:	8F A2 D8 B6 6F C1 E4 B6 55 A0 D0 B7 74 5F D0 B6	2654
09E0:	5A C0 80 B6 60 5D 0A B6 60 30 45 B6 65 B0 F9 B5	2075
09F0:	66 5B DF B4 63 5C E5 B5 60 5D 00 58 00 38 21 B3	1742
0A00:	80 10 21 B3 86 90 21 B3 86 B0 22 B4 85 B0 A2 B2	2019
0A10:	85 20 A7 B0 97 44 92 AE A7 DE 92 AE 9F 5F 52 AE	2266
0A20:	85 DF 50 AE 7D 41 11 B1 6D C0 16 B4 6E A0 96 B6	2099
0A30:	66 A0 56 97 66 C0 D8 97 3C 5E A1 B7 4D A1 A6 B7	2245
0A40:	66 A3 E2 B8 66 C2 E5 97 66 41 D5 94 64 C0 40 D1	2444
0A50:	68 7C 00 48 00 30 66 B5 90 00 66 B5 95 08 77 B4	1514
0A60:	97 29 77 B4 97 C0 B7 B4 97 5F 77 B4 97 5E F7 B5	2415
0A70:	96 DE B7 B5 8E D0 F6 B5 8E 5E FA 95 B5 5E F6 95	2783
0A80:	7C DF FA 96 7C BE FA 95 7C FF FA 95 B4 C1 F1 94	2952
0A90:	84 21 50 95 B8 40 90 D6 90 48 00 54 00 30 B6 D8	1653
0AA0:	70 00 B6 D8 74 88 B6 D9 57 27 A6 D9 4F 41 A6 D9	2149
0AB0:	47 5F B7 D9 47 20 A0 D8 5E D0 11 D8 4E D0 25 D7	2131
0AC0:	56 D0 15 D8 56 30 0A D7 5E 30 0A D7 5D D0 E6 D8	2004
0AD0:	65 C0 F2 B8 4F 44 F0 D9 4F 42 E1 D9 47 40 E4 DA	2491
0AE0:	47 40 E8 DA 46 40 D8 DA 4C C0 C0 B8 40 57 00 74	2064
0AF0:	00 30 40 B3 98 00 74 B2 BE 2A 77 B3 BF 43 AB B3	1827
0B00:	AF 44 9A 91 76 38 95 B1 3A DA 05 B3 44 20 05 B4	1787
0B10:	4D 8A 1A B1 56 20 6A B0 56 B0 9A B3 7E C0 AB B4	2034
0B20:	66 5D 03 F8 6D D0 03 F8 5D F0 96 B8 55 B7 D9 B8	2350
0B30:	4D DE C6 B8 4D 3D D6 98 54 DE D6 92 32 5E 51 AF	2251
0B40:	19 C0 68 B1 12 C0 AB B2 18 3F 3C B3 24 20 2C B3	1674
0B50:	4C C0 AC B2 7D 40 BC B3 BC C0 6C B2 BC 50 7B B3	2311
0B60:	88 2F 00 BC 00 30 61 B6 98 00 61 B6 9D 80 61 B6	1645
0B70:	9D AF 62 B6 9F 45 52 B6 7F E1 B1 97 76 A3 B5 93	2297
0B80:	65 3A D0 B6 68 42 80 B3 6C B0 91 93 66 90 18 92	2021
0B90:	6D B2 39 94 76 21 F6 94 7E A0 B2 B2 97 5B B2 B0	2227
0BA0:	97 5C 76 AF 8E 5A A6 D0 94 DC 96 B2 B0 3C A9 94	2356
0BB0:	9F 40 A9 B6 97 5D A6 98 7E 5F CD 97 63 41 50 B9	2142
0BC0:	70 60 09 98 76 30 11 99 74 B0 20 B9 7A 30 08 97	1543
0BD0:	80 70 01 B5 85 30 74 D6 85 30 A0 B5 73 B0 F5 95	2140
0BE0:	7D 27 F5 B6 85 44 B6 B6 BC 30 00 B6 80 50 00 70	1846
0BF0:	00 30 76 B4 98 00 76 B4 9F 0C 76 B4 9F 26 76 B4	1760
0C00:	5F E1 A2 B4 66 D1 95 B5 5C A1 21 E6 75 10 25 D5	2328
0C10:	85 70 55 B4 8D 22 B8 B0 0E DF 67 AF 9F 5F 63 AD	2277
0C20:	A6 C0 93 AD 9E 20 56 AD B4 A0 59 B1 54 BF 95 B2	2287
0C30:	55 A0 62 B1 66 BF 76 B3 7F 5F B4 B4 7F 40 99 D5	2248

0C40:	7C	C0	1A	F5	7C	90	1A	F7	75	70	A4	F0	6D	70	A6	95	2367
0C50:	76	45	61	D0	5D	C1	61	CE	54	7F	00	D1	50	7F	00	70	1069
0C60:	00	31	20	E5	00	00	20	E5	00	1E	74	E5	3A	30	74	F5	1205
0C70:	54	26	74	E6	21	5D	74	E5	26	21	28	E5	50	5F	00	24	1360
0C80:	00	30	A6	C1	68	00	66	D0	61	3E	E6	E1	EF	E1	E7	E3	2064
0C90:	97	5E	E6	E3	9E	C0	E9	E2	96	60	E6	E2	8D	25	A0	E3	2447
0CA0:	80	23	00	54	00	30	96	D7	48	00	A6	D5	4E	27	F6	D4	1687
0CB0:	4E	A2	E7	E4	E7	A1	77	D2	9F	C1	77	D0	A7	C1	76	CF	2592
0CC0:	8F	22	91	AF	5C	C1	90	CF	3E	21	55	AF	71	C1	41	E2	2034
0CD0:	71	04	41	E5	76	30	61	D6	6D	30	E6	E6	7E	24	F2	E5	1995
0CE0:	8F	C1	72	E3	BF	5F	91	E0	7D	42	90	E1	5C	DF	C1	D1	2417
0CF0:	4E	D6	41	D3	28	37	00	38	00	31	20	E5	00	00	20	E5	1191
0D00:	00	1E	74	E5	3A	30	74	E5	54	26	74	E6	2E	5D	74	E5	1586
0D10:	26	23	28	E5	56	42	69	E4	76	A0	6A	E4	76	C0	6E	E4	1892
0D20:	77	5E	AE	E4	6E	DE	EE	E4	5D	A1	42	D7	50	40	00	24	2026
0D30:	00	30	51	D6	70	10	51	D6	74	10	F7	D5	45	20	AE	D5	1827
0D40:	46	20	A3	E8	5F	20	A7	E8	5F	A0	E2	E8	5F	A0	E2	E8	2257
0D50:	50	60	00	48	00	3A	44	E4	5D	C0	54	E2	66	BF	62	E0	1668
0D60:	77	04	72	E0	BF	1C	71	E1	E7	E0	E1	E1	CF	5B	71	E0	2187
0D70:	E6	BE	61	AF	E5	D0	22	E0	DD	D0	22	D0	EC	D0	91	D9	2720
0D80:	FE	50	E1	D9	FD	D0	61	E9	FD	70	15	96	C3	E0	DA	94	2744
0D90:	E3	A5	90	E4	73	3D	95	E4	78	3D	00	48	00	30	95	93	1722
0DA0:	38	00	ED	92	28	E0	ED	93	34	04	EE	93	34	E0	DA	93	1980
0DB0:	3C	E2	99	95	46	21	96	E7	67	A1	9A	E7	6F	A0	96	E5	2131
0DC0:	56	DF	A1	E3	56	40	27	F3	56	30	17	F7	40	F0	E2	D7	2243
0DD0:	4D	30	D5	E8	56	59	D5	E9	4D	DE	E6	E9	45	C1	D9	E9	2473
0DE0:	38	46	00	50	00	27	40	E5	40	00	ED	92	35	2C	EA	93	1367
0DF0:	56	26	E6	E4	7F	23	E6	E5	7F	C1	A6	E6	67	C1	79	E6	2278
0E00:	5E	DC	ED	E6	56	21	99	D7	55	EE	99	E7	55	60	E5	E7	2373
0E10:	5E	C4	75	E7	5F	23	A1	E7	57	22	A1	E7	57	7F	62	E6	2023
0E20:	6F	41	A2	E5	77	20	65	D3	E6	BC	E5	D0	7C	5E	EC	E1	2324
0E30:	30	5F	00	58	00	30	C9	8D	48	00	CD	8E	4C	2F	E5	92	1442
0E40:	4C	A6	E6	E0	56	C1	E6	D2	4D	C2	E1	E0	3C	22	EA	E6	2173
0E50:	45	C2	E5	D8	4E	21	C2	DA	56	DF	C9	91	4C	DD	DD	91	2453
0E60:	4C	BE	D9	E1	44	EE	A2	E1	56	5D	76	E2	8F	DE	E5	E1	2452
0E70:	8D	DD	E7	E1	E5	BF	AA	E1	54	3D	D8	E2	3B	EE	E8	E2	2591
0E80:	38	DD	C4	E4	3A	43	C4	E4	38	41	00	58	00	2F	DC	95	1782
0E90:	38	A0	DD	95	34	27	A2	E8	4F	28	E6	D8	57	A0	E6	D9	2237
0EA0:	4F	C1	E2	D9	4F	21	D7	97	45	A0	D9	96	45	5D	D9	E7	2351
0EB0:	45	BE	A0	E8	47	3D	A0	E8	57	5C	E1	E8	5F	5B	F1	E8	2230
0EC0:	54	5D	A4	F8	50	E0	00	E6	60	63	00	E6	60	43	02	D6	1831
0ED0:	6A	E0	52	D6	75	30	FA	E5	E5	DF	F6	E5	95	5D	D0	E5	2588
0EE0:	90	7E	00	48	00	30	44	E4	58	00	4C	E4	5C	41	4A	E4	1393
0EF0:	5D	C1	E9	E2	36	38	49	E2	26	3D	45	E1	46	5F	45	E0	1717
0F00:	4E	23	44	E0	5E	21	94	E3	E6	C1	94	93	8E	A1	E5	97	2020
0F10:	D7	40	E1	98	F7	5F	E1	94	FF	21	62	D1	D7	5F	72	D2	2408
0F20:	A6	C0	21	E2	66	41	11	94	58	42	00	48	00	30	98	E2	1505
0F30:	28	00	75	E1	5E	A1	E5	E3	E7	E0	E1	E3	65	5E	E0	E3	2262
0F40:	82	DE	E4	E5	71	42	41	D8	78	C1	00	E7	79	E0	11	96	2085
0F50:	7D	E0	66	E5	7E	E0	E5	E5	7F	3E	FA	E5	7F	DE	FA	95	2664
0F60:	7F	7F	FA	95	7E	7F	A9	94	75	7F	74	F2	51	FF	10	D2	2387
0F70:	20	4B	00	38	00	30	8A	95	40	00	8A	94	4D	2F	8A	94	1258
0F80:	4E	27	51	E8	5F	3B	EA	E8	66	E9	EC	E8	55	FD	E8	E8	2415
0F90:	64	DF	E0	E5	5C	FD	8C	E4	64	DC	FD	E3	5C	DE	BD	E3	2840
0FA0:	5C	A1	BD	E4	94	C0	AD	E3	98	4D	FF	E7	E8	70	00	00	2325
0FB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
0FC0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
0FD0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
0FE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
0FF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0

VOCAB II

No	Ad.Deb	Ad.Fin	Texte
00	0070	00A7	NIVEAU
01	00A8	00EB	NORMAL
02	00EC	0113	NOUS
03	0114	0167	NUMERO
04	0168	019B	OUVERT
05	019C	01E7	OUVERTE
06	01E8	0233	OUVERTURE
07	0234	0277	PANNE
08	0278	02C7	PAROLE
09	02C8	02F3	PLU(S)
10	02F4	0327	PLUS
11	0328	035B	PORTE
12	035C	039F	POSER
13	03A0	03FF	POSSIBLE
14	0400	042B	POUR
15	042C	049F	PRESENTE
16	04A0	04E7	PRESSION
17	04E8	0523	PREVU
18	0524	059B	RADIATEUR
19	059C	05F3	RAPPELER
20	05F4	066F	RECLAMATION
21	0670	06BF	REVISION
22	06C0	0723	SECONDE
23	0724	0783	SELECTION
24	0784	07DF	S'IL VOUS PLAIT
25	07E0	0853	SYNTHETISEUR
26	0854	08AB	SYSTEME
27	08AC	0913	TEMPERATURE
28	0914	0967	TENSION
29	0968	09BF	TERMINER
30	09C0	09FB	TEST
31	09FC	0A1B	TOUT
32	0A1C	0AB8	TRANSMISSION
33	0ABC	0AB7	TROP
34	0AB8	0AFF	TYPE
35	0B00	0B7B	ULTERIEUREMENT
36	0B7C	0B9B	UNE
37	0B9C	0BF3	VALABLE
38	0BF4	0C3F	VEUILLEZ

39	0C40	0C73	VOTR(E)
40	0C74	0CB7	VOTRE
41	0CB8	0CDF	VOUS
42	0CE0	0D17	ZERO
43	0D18	0D3B	UN
44	0D3C	0D6F	DEUX
45	0D70	0DAB	TROIS
46	0DAC	0DEB	QUATRE
47	0DEC	0E43	CINQ
48	0E44	0E8B	SIX
49	0E8C	0EDB	SEPT
50	0EDC	0F23	HUIT
51	0F24	0F5B	NEUF
52	0F5C	0F63	SILENCE 1
53	0F64	0F6F	SILENCE 2
54	0F70	0F83	SILENCE 3

VOLUME II

0000:	00 70 00 A8 00 1C 01 14 01 68 01 9C 01 E8 02 34	1086
0010:	02 70 02 C8 02 F4 03 28 03 5C 03 A0 04 00 04 2C	923
0020:	04 A0 04 E8 05 24 05 9C 05 F4 06 70 06 C0 07 24	1210
0030:	07 84 07 E0 08 54 08 AC 09 14 09 68 09 C0 09 FC	1246
0040:	0A 1C 0A 8C 0A B8 0B 00 0B 7C 0B 9C 0B F4 0C 40	1026
0050:	0C 74 0C B8 0C E0 0D 18 0D 3C 0D 70 0D AC 0D EC	1229
0060:	0E 44 0E 8C 0F 0C 0F 24 0F 5C 0F 64 0F 70 FF 00	1125
0070:	00 38 00 30 09 95 30 00 C9 93 34 A1 C2 19 3E C0	1696
0080:	05 09 3E C0 F2 B8 4E C0 E2 B4 55 5E 94 E3 64 DF	2615
0090:	EA AD 65 C5 EB AC 66 41 E7 AF 66 C1 E6 E1 5D 3E	2494
00A0:	D5 AE 4E BF D2 B0 40 3A 00 44 00 30 D0 E1 50 00	1742
00B0:	D1 E1 4C 25 B5 93 5D A0 B6 E2 66 C0 B6 CF 66 40	2289
00C0:	75 CE 85 A0 A5 CD 74 40 D6 AD 64 40 D9 AE 3E BF	2358
00D0:	D6 B0 34 3F 96 90 2C 3E 76 B1 A7 5D 76 B2 B6 DF	2161
00E0:	F0 B2 73 FE D1 B3 5A 7F A8 B4 60 49 00 20 00 30	1997
00F0:	98 B5 48 00 D8 B3 3C AB C8 B3 45 A5 D9 B3 45 A1	2270
0100:	95 91 4D DF B5 B3 4E DD 96 B5 4C DE 96 B5 4E BE	2435
0110:	96 B5 58 3D 00 54 00 30 00 00 00 00 80 B5 62 BC	1207
0120:	C4 94 43 20 C5 94 44 40 B1 98 4E C4 A2 98 47 41	1973
0130:	99 95 3D DE DD 93 2B DF 98 96 4E C3 A2 B8 5F 40	2302
0140:	A6 B8 67 C0 62 B4 7F 5F 63 AF B5 DD 96 CC 7D 41	2317
0150:	D2 CC 7D C2 06 CC 7D C1 D7 C9 75 42 E6 A9 75 40	2648
0160:	C6 C8 73 DF C0 B0 68 7E 00 34 00 30 EB B2 60 40	2007
0170:	D6 B1 5E 62 9A B0 56 3F 56 91 65 5E FE 94 76 DE	2230
0180:	EB 96 7F BF EB 96 87 DD FB B5 BF FF EA B3 B6 5F	2916
0190:	65 B2 8D A3 E4 B4 B6 C1 B5 B4 80 42 00 4C 00 30	1954
01A0:	EB B2 60 40 D6 B1 5E 62 9A B0 56 3F 56 91 65 5E	2061
01B0:	FE 94 76 DE EB 96 7F BF EB 96 87 DD FB B5 BF DE	2983
01C0:	EA B3 B6 5F 65 B2 8D A3 E4 B4 B6 A1 B5 B4 B1 42	2344
01D0:	44 B5 79 25 00 B5 6C 30 01 D5 6D 50 79 B5 6D 50	1638
01E0:	29 B5 68 D0 21 D5 68 30 00 4C 00 30 EB B2 60 40	1632
01F0:	D6 B1 5E 62 9A B0 56 3F 56 91 65 5E FE 94 76 DE	2230
0200:	EB 96 7F BF EB 96 87 BA FB B5 BF FF EA B3 B6 5F	2881
0210:	65 B2 8D C1 07 B8 45 D0 EB B8 4D C4 EB 97 4E 61	2334
0220:	D2 B7 55 45 EA B6 66 C1 D5 94 65 A4 E4 B2 65 24	2427
0230:	70 CF 48 7B 00 44 00 30 40 B3 70 00 29 B2 95 B0	1529
0240:	A9 B2 A6 B0 F0 B1 B6 DF AD B2 B6 DF ED B3 AE BC	3071
0250:	FC B2 B4 BE EC B2 3C 3D FC B2 34 3F FC B2 34 3E	2376
0260:	B9 B3 33 BF B0 B2 3B A0 A9 B3 33 A0 69 B3 33 A0	2246
0270:	B9 B2 2B A0 79 B2 70 4A 00 50 00 52 80 B3 60 00	1616
0280:	15 B5 B6 B0 55 B4 B0 9C 26 B1 CF DE 2B B0 CF DF	2367
0290:	6A B0 A6 D7 6A AF 9E 5D 6A AF BF 42 6E B0 A7 40	2250
02A0:	69 B1 9F 40 A8 B2 B6 41 FC B3 5D 62 FC B3 54 BC	2375
02B0:	FC B3 5D 23 B0 B3 BE C0 FD B3 9E C0 B3 B3 96 5F	2657
02C0:	99 B3 92 5A D4 B3 90 41 00 2C 00 35 60 53 60 00	1540
02D0:	A1 54 5D 25 71 95 4D A1 B6 B5 5E A7 A2 B7 4F DF	2146
02E0:	E1 D7 4F 41 E1 B7 4F 40 E5 B7 4E C0 F5 B7 4D 40	2386
02F0:	96 D7 50 36 00 34 00 35 60 53 60 00 A1 54 5D 25	1254
0300:	71 95 4D A1 B6 B5 5E A7 A2 B7 4F DF E1 D7 4F 41	2355
0310:	E1 B7 4F 40 F5 B7 4D 40 16 F7 5D D0 16 F7 B5 F0	2332
0320:	16 F7 9D F0 16 F7 B8 70 00 34 00 30 00 B3 58 00	1550
0330:	11 B3 76 30 FA 91 7E 2F FC AF B7 C3 B5 AE 97 DE	2415
0340:	6B CD 7E 40 41 B4 9C F0 C1 D7 98 21 D9 D8 9E 30	2378
0350:	E1 B8 6D 3D E6 B7 68 B8 EA B7 58 58 00 44 00 30	1998
0360:	B2 99 70 C0 B2 99 76 90 3C 71 6D 20 D4 B3 64 AF	2112
0370:	D5 CB 7D 45 E6 AD 6E 40 F1 AF 6E DF E5 D4 5E 5D	2564
0380:	D5 D6 6D C0 F1 B5 5D C2 E1 97 5F 42 E5 97 6F A2	2627
0390:	D5 97 67 40 E2 B7 5E D0 C4 98 5C 55 11 D6 F8 60	2355
03A0:	00 60 00 30 A6 91 40 00 05 B3 44 08 05 B4 4D 8A	1179
03B0:	1A B1 56 20 6A B0 56 B0 9A B3 7E C0 AB B4 66 5D	2014
03C0:	03 F8 6D D0 03 F8 5D F0 96 B8 55 B7 D9 B8 4D DE	2454
03D0:	C6 B8 4D 3D D6 98 54 DE D6 92 32 5E 51 AF 19 C0	2169
03E0:	6B B1 12 C0 AB B2 1B 3F 3C B3 24 20 2C B3 4C C0	1725
03F0:	AC B2 7D 40 BC B3 8C C0 6C B2 8C 50 78 B3 B8 2F	2226

0400:	00 2C 00 36 44 B6 78 10 65 B4 7E B0 55 B1 7D A0	1614
0410:	95 E1 7C C5 E9 B1 7C C1 D1 B2 86 5F DA B1 B6 3C	2503
0420:	67 B1 8D 59 15 B1 96 59 41 B5 78 4B 00 74 00 3F	1564
0430:	15 E4 60 10 05 B4 7E 30 51 B4 95 B0 5B AF 96 20	1706
0440:	2F B3 9F C0 AF B5 9F C0 AA B5 76 DD D1 05 5C DB	2720
0450:	53 05 4C A1 66 D5 3D 20 AA B4 65 11 AA B2 7E DF	2106
0460:	27 B0 A6 DF 23 AE A6 C1 37 AF 9D E1 52 AF 9E 42	2264
0470:	76 D0 8D A7 A9 B1 6C A5 90 B1 70 C0 91 B1 70 C0	2504
0480:	16 B2 68 60 16 B3 63 B0 F6 B4 65 20 E6 B3 6D 20	1985
0490:	B1 B4 75 C0 B9 B4 7D DD E4 B4 5A 5E D0 B5 60 5D	2547
04A0:	00 4B 00 30 B0 B6 B0 00 21 92 87 30 22 B0 7C AE	1428
04B0:	AA B5 86 C1 AA B6 7F C1 E4 B7 66 DF A7 D8 76 D0	2795
04C0:	25 F9 76 B0 41 B9 66 70 09 98 5D B0 E1 B7 5D B5	2156
04D0:	A5 B7 76 DF 91 AF 5D BF 61 CE 66 5E 65 CE 5E 20	2220
04E0:	D4 B1 2F 21 D4 B1 28 21 00 3C 00 30 50 B5 38 00	1352
04F0:	55 B4 4F B0 62 B2 6C C0 EA B4 7F C7 EA B5 77 A2	2532
0500:	DA B5 67 C0 D4 B5 5D DD D6 B6 5D DB FB 95 4D 5D	2679
0510:	F8 96 54 FC D1 98 54 C1 C5 99 4C 20 E4 97 52 C0	2467
0520:	10 D6 40 4A 00 78 00 30 E3 AF 60 00 E5 AE 76 21	1588
0530:	B5 AF 7D A1 B5 AF 7D B2 B0 8F 21 B4 B3 8F 41	2350
0540:	B4 B4 76 A2 A1 94 7B 41 D0 D8 4C DF E1 D8 45 20	2402
0550:	E2 B8 46 3F E5 B8 67 5F F4 B6 7F 5F E4 B5 64 40	2375
0560:	E0 B5 73 3F 90 B5 61 7F 21 D6 65 30 11 D5 6E 30	1916
0570:	FA B5 6E 20 FA B5 7E DF F9 B5 7E DF F6 B5 7E 7F	2812
0580:	F5 B5 7E C0 A5 B4 7E 40 A1 D4 7E 3F 61 D4 75 BF	2458
0590:	65 D2 54 DF 50 D2 4A C5 14 D2 00 24 00 58 00 32	1583
05A0:	3A B3 80 10 3A E3 85 10 36 B3 8D B0 79 B2 96 A0	1926
05B0:	6E B1 C7 C1 6E B2 BF A5 BA B1 6D C0 A5 B1 80 7F	2584
05C0:	00 B5 9B 10 00 B5 BA 10 34 B2 7E 10 28 B2 6F 10	1401
05D0:	68 B4 4E A3 A9 B5 56 A0 E6 B5 3E 5F E9 B7 67 41	2273
05E0:	D4 B7 6E C2 D9 B8 5D C1 D4 B9 5D 40 C0 B9 5A BA	2593
05F0:	C0 B9 58 38 00 7C 00 3D 15 B0 80 00 28 B1 8D C3	1571
0600:	B0 B3 8E 43 BE B4 8F 3E AE B6 7F 20 99 B6 54 DB	2305
0610:	E4 B4 41 5D A5 B5 50 A0 19 B4 5D 30 19 B5 60 BF	2004
0620:	A9 B4 76 23 FE B4 8E 22 AE B3 8E 5E DA 91 3C DD	2345
0630:	9E 92 3D 41 AB B3 86 5F A9 B4 6D 5F 07 D5 55 D0	2075
0640:	07 D7 4E 70 07 DB 4E 50 4A B8 4D 24 C9 B6 4D 20	1656
0650:	86 94 55 20 55 90 6D 40 59 8C 75 40 41 92 7D C3	1742
0660:	50 B2 5D C3 D0 D3 3B C3 40 D2 49 DD 40 B1 70 5F	2235
0670:	00 50 00 30 B6 B2 88 00 A5 B3 85 A1 A9 B5 87 C1	1940
0680:	E6 B8 5F E0 D9 B6 56 5F D1 B7 5D 5D D6 B8 40 C0	2558
0690:	D5 99 4E 25 E6 B9 4F 22 E9 B9 46 7F D6 B8 4D DF	2322
06A0:	C6 DA 55 5F 97 D9 56 B0 96 D9 56 B0 D5 B9 45 C1	2515
06B0:	A1 B3 76 E3 A1 CE 5E 41 91 AF 7D C1 71 BE 68 5A	2298
06C0:	00 64 00 3E 1E F7 A8 10 1E F7 AC 10 1E F7 AC 30	1585
06D0:	1A F7 8D 70 1A B5 7E 40 FE B4 6F 22 FE B4 6F 40	2111
06E0:	AE B5 6E A0 D7 95 4B DF 63 90 5D DC 47 91 76 21	2207
06F0:	47 B0 A6 DB 13 AB AD DD 13 A4 AD 5F 43 A9 A4 C0	2259
0700:	53 A9 93 C1 53 AA 8B 41 C4 B2 40 E2 B0 B5 4A 42	2162
0710:	62 B4 74 21 F2 B4 84 A2 F2 B4 8D 43 E2 B5 94 5E	2422
0720:	90 B5 78 5B 00 60 00 30 06 F7 58 00 06 F7 65 30	1423
0730:	06 F6 66 50 06 F7 6E B0 5A D6 77 A0 BA B6 7F CE	2257
0740:	FA B5 6F 3F F6 D5 4E 5A BA B6 6F BE DA B7 67 5C	2497
0750:	E1 97 53 59 54 B6 64 DA 16 F7 66 B0 16 F7 5C F0	2280
0760:	16 F7 55 F0 16 F7 55 B0 EA B8 55 A0 D6 B8 5D BF	2469
0770:	9B B0 75 DE 96 AF B5 C0 A5 AC 7D 3F 90 B1 65 5F	2362
0780:	B0 B5 4B 43 00 5C 00 30 0A F1 40 00 0A F6 44 F0	1467
0790:	0A F6 45 D0 DA DB 4E AF 8B B9 3F 2F EA B7 36 CB	2325
07A0:	EA B6 36 3D AA B5 5F 5A B5 B3 4D D9 E6 AF 55 5D	2256
07B0:	D7 AC 55 41 D7 AE 4D 3F C0 B3 49 FD 25 B4 4C 10	2072
07C0:	25 B3 4D 90 24 B3 56 00 FC B4 4D DE FC B5 6E 5E	2106
07D0:	BC B6 8E 5E FC B6 9D DA FD B7 8B 49 FD B7 8B 40	2696
07E0:	00 74 00 30 56 F7 50 00 16 F6 5E 50 16 F5 6F 70	1509
07F0:	B9 B4 9F 4F 69 B3 A7 4B A5 B4 7F 40 40 B4 71 5F	2117
0800:	54 D1 7B 3F 55 D4 75 10 16 E7 5E 9C 19 B3 66 DF	2000
0810:	B5 B8 66 BF D4 B8 44 EC 99 E7 3B 5F 09 B7 4B 50	2099

0820:	08 08 4D B0 B4 DA 45 DC D5 D9 4D C1 E6 B7 45 08	2514
0830:	E6 B6 44 50 56 B6 3C B0 AA B5 5D BD 99 B5 77 28	2190
0840:	EA B4 BF C3 1A B0 9F DE 15 AE B7 57 6B CE 76 3D	2244
0850:	99 AD 58 5E 00 58 00 30 12 F7 48 00 12 F7 4E 50	1401
0860:	02 F8 57 70 02 F8 57 50 99 B9 4E CC A9 B9 4E C0	2110
0870:	94 B9 46 DE 02 F8 5F 50 02 F7 63 70 02 F6 61 B0	2031
0880:	02 F6 67 30 A9 B5 67 32 E8 B5 B7 A0 EA B5 B7 C0	2352
0890:	EA B4 B7 5F 56 B3 86 38 45 93 3E 42 49 93 36 DE	1939
08A0:	55 93 46 5F C8 B0 44 DF C5 B5 30 62 00 68 00 30	1740
08B0:	00 B4 79 00 20 B5 86 B0 B6 B2 B6 AB A6 CF BF 41	2275
08C0:	51 AC 55 C1 96 AF 44 21 A6 AD 3A A1 51 D1 7B B0	2104
08D0:	15 B4 BE B0 E2 B6 6D A1 B6 B6 7F C0 E5 B5 7E DF	2639
08E0:	A5 D3 7E 3F B6 D2 97 DE F6 D4 97 DF 95 B3 64 5F	2685
08F0:	E1 B3 93 20 58 D9 9D D0 10 D9 7C B0 E1 B7 56 40	2344
0900:	E1 B7 56 60 F5 B5 6E E0 A5 CF 7E 41 50 CF 5C DE	2514
0910:	95 AF 30 3E 00 54 00 30 A4 B5 78 00 11 B3 7B 30	1398
0920:	24 D5 B6 30 A6 B2 BF AF 66 B0 9F A3 A5 CE 6F 40	2239
0930:	AB CE 5E C1 A5 CF 4D DE 45 F1 35 BF 02 D2 3E B0	2336
0940:	41 D9 4E F0 21 B8 56 B0 21 B8 5E 50 61 B8 66 30	1907
0950:	A2 96 66 A1 A6 91 66 C0 52 AF 5E A0 41 B0 B6 5D	2159
0960:	85 B2 5E 20 CC AF 28 5B 00 58 00 30 01 D6 B8 10	1450
0970:	01 D6 BE 90 51 B7 7D A0 A7 B7 76 AF A7 B8 7F C1	2364
0980:	A5 B6 97 DE 55 D5 97 3E 46 B4 86 40 49 92 4C BF	2165
0990:	B8 94 4C A2 41 96 4D A0 61 D8 4E 3D A2 D9 46 5E	1969
09A0:	A3 B8 45 5D D9 95 4B 5E EC 95 4A BE AB B8 65 5E	2243
09B0:	A6 B9 75 5E B2 B9 6C 40 E7 99 5B 5F A9 B8 78 4B	2212
09C0:	00 3C 00 3E 11 D7 78 10 12 B8 7E B0 BA B5 7E 20	1519
09D0:	B6 B6 97 C2 BA B6 B7 41 E4 B6 6D DE AA F7 7C 00	2767
09E0:	1A F7 94 70 1A F7 93 30 01 96 90 50 00 B6 98 D0	1918
09F0:	66 B6 9C B0 20 B5 9C 32 20 B5 B8 40 00 20 00 42	1546
0A00:	11 B3 50 10 14 D6 5E B0 A6 B5 5E 20 FD 90 5D C1	1952
0A10:	E4 AF 5D 40 50 B3 65 DF A1 B2 68 31 00 70 00 31	1796
0A20:	10 B4 00 10 14 B4 07 30 59 B1 76 40 1A AF 9E 22	1308
0A30:	85 B2 AE A4 15 BC C7 43 15 BA 9E C2 42 B2 5E 5C	2017
0A40:	02 D4 4E 30 03 F5 4E 10 03 F9 34 70 C6 98 3B C0	1699
0A50:	B5 B6 56 C3 B5 B7 4E DE 5A B8 3D DE 02 D9 3D 3E	2111
0A60:	03 F8 3E 30 03 FA 46 70 96 DA 46 3E 46 B8 5E C2	1838
0A70:	56 B6 6F 22 46 B4 76 BF 45 BF 7E 5F 95 91 65 5F	1895
0A80:	90 AF 74 C0 B0 AF B3 DF 00 B0 B0 5F 00 2C 00 30	1775
0A90:	15 D6 58 10 15 D6 5F 10 52 D2 6E 20 D2 AC 7C CA	1827
0AA0:	E3 AA 7D A4 C1 AF 7E C2 D1 AC 75 DE D2 AD 65 5C	2670
0AB0:	C2 B3 74 D8 D1 B7 60 3A 00 48 00 30 03 94 B8 00	1661
0AC0:	0B 95 96 30 0B 96 97 B0 0B 96 97 10 15 B8 7E B2	1635
0AD0:	A6 B9 57 23 41 BA 57 C2 51 BA 5F BE 9A B9 4F BE	2165
0AE0:	B9 98 4D 3D CD 98 2A BF 00 B6 28 7E 10 B6 59 D0	1860
0AF0:	11 B5 76 B0 6A B3 BE 3E FE B3 B0 5E ED B4 B8 47	2273
0B00:	00 7C 00 30 51 B8 50 00 D7 B9 54 A5 D7 B8 56 44	1719
0B10:	96 B7 4E DF A6 B6 4E C0 F6 B7 46 DF A7 B6 3D BD	2573
0B20:	51 95 32 DD E0 B7 40 41 14 B5 4A B0 AE B3 55 30	1974
0B30:	D8 B7 5D A6 E9 B6 6F 21 BA B5 7F 5F BA B4 7F 5E	2393
0B40:	DA B8 67 DD DA D9 4E DF F9 B9 4E A0 EA B1 5F 22	2674
0B50:	DA B5 52 25 A9 B2 B6 FF 69 B0 B6 BD 6A AE 7D DD	2484
0B60:	67 AD 6C 5D F9 B2 44 5F 99 B3 3C DF 53 BF 93 C1	224E
0B70:	33 BE 7D 45 71 AE 7C 5D 54 AE 60 48 00 20 00 30	1397
0B80:	61 B7 58 00 61 B7 57 A2 A6 B8 47 A4 EE B7 36 20	1985
0B90:	D4 B5 3C C2 95 B4 42 7F 40 B3 48 5D 00 58 00 30	1713
0BA0:	96 B0 40 00 59 B1 65 00 6D B0 7E C2 AE B2 97 A1	2026
0BB0:	BD B3 A7 DE AE B2 76 DD BD B2 6E 20 BD B2 76 BD	2631
0BC0:	DD B2 A6 DD FE B2 A7 20 F9 B2 AE A0 FF B1 95 5F	2854
0BD0:	60 AF 52 40 A5 AF 39 FF 94 B3 6C C1 AC B3 6D 40	2221
0BE0:	BC B2 64 40 F9 B3 B5 5F FD B3 95 C0 AD B3 93 45	2527
0BF0:	94 B4 B8 45 00 4C 00 30 54 B4 B0 00 54 B4 B4 B4	1577
0C00:	69 B4 65 21 19 B4 55 A0 1D B4 5D A0 15 B6 66 30	1684
0C10:	15 B4 6F 41 AA B3 77 C0 BA B4 77 A1 DB B8 4F C0	2357
0C20:	D1 F8 3D 40 D4 D8 2C A2 D5 D9 35 BF D5 D9 47 5E	2485
0C30:	CA B7 67 E0 CA B7 67 40 CA B7 66 BE 3A B3 B0 22	2337

0C40:	00 34 00 30 00 97 48 00 00 97 4A 1E 51 B1 3E DE	1117
0C50:	66 E0 7D 3F B6 B0 8E 2B B6 B1 BF 42 F5 D2 7E C0	2350
0C60:	9A D0 6B B2 E5 B1 71 D8 02 B8 8E 50 42 96 B5 B0	2315
0C70:	F4 CF 85 CA 00 44 00 30 00 97 48 00 00 97 4A 1E	1380
0C80:	51 B1 3E DE 66 B0 7D 3F B6 B0 8E 2E B6 B1 BF 42	2116
0C90:	F5 D2 7E C0 9A D0 6B B2 E5 B1 71 D8 02 B8 8E 50	2563
0CA0:	42 96 B5 B0 74 CF 85 C6 A5 D2 B6 A9 A2 B3 7E A0	2612
0CE0:	A7 94 66 5F C6 93 78 41 00 28 00 2F AA B5 58 00	1568
0CC0:	85 B3 4A 90 B5 B3 4C CB 85 B3 4D A1 E6 AF 56 5D	2255
0CD0:	D7 AC 56 41 D7 AC 4E 3F 80 B1 34 DD C0 B3 48 5A	2179
0CE0:	00 38 00 30 65 D4 58 00 A5 D4 54 A2 56 D5 56 A2	1675
0CF0:	E6 B5 5E 40 ED B6 6F C2 EA B6 7F C0 AA B0 B7 5F	2604
0D00:	96 AB 8D DD EB A9 7D E1 DA AB 75 60 95 AB 7D 42	2544
0D10:	40 93 8A DD 80 94 88 5F 00 24 00 30 10 B5 00 00	1358
0D20:	70 B3 7E 2F 70 D4 7F 24 70 D4 B7 DF 71 D4 B7 DA	2311
0D30:	74 D4 75 7C B0 D4 71 E2 B0 B4 60 23 00 34 00 30	1883
0D40:	85 D4 28 10 06 B5 35 10 07 D6 3E 90 A9 B6 56 20	1553
0D50:	EF B6 5F BD EB B4 6F C6 D6 B3 6F 43 EA B3 66 DE	2737
0D60:	FA B4 66 DA F6 B4 66 3E EA B4 64 C5 40 B6 80 7E	2551
0D70:	00 3C 00 2E 14 D3 88 10 14 D3 8C 90 24 D0 06 B0	1430
0D80:	55 D2 6D 30 C4 D2 62 A0 CA A7 7B 5F D9 AB 7C 20	2244
0D90:	DA AC 96 40 69 AD B7 C2 7D AE AF C1 B8 AF B7 5E	2562
0DA0:	BD AF A6 5B 76 AF 93 60 40 D4 98 46 00 40 00 3D	1780
0DB0:	0B B6 C8 10 0B B7 D7 30 0B B8 8F 30 AA B6 7F A0	1882
0DC0:	F9 B5 97 BB B4 B4 B7 DE 74 B3 BF DE B8 B2 A7 5E	2864
0DD0:	90 B3 95 5C 51 B3 8A C7 14 B4 8A E0 A5 B2 C4 D0	2470
0DE0:	A0 D1 FC 50 40 D2 FC 50 01 B1 E8 50 00 58 00 36	1939
0DF0:	19 F7 80 10 19 F7 84 10 19 F7 85 B0 19 F7 86 50	1903
0E00:	19 F7 86 D0 19 F7 86 B0 A4 D4 6E A0 78 D4 9F C0	2525
0E10:	74 D3 8F FF 71 B3 77 5E B0 D4 7E DB 91 B5 75 5E	2500
0E20:	95 B6 7C C3 D0 D7 53 41 10 B4 61 61 00 B4 68 70	2007
0E30:	04 B4 70 30 04 B5 7D 30 04 B7 86 30 1B B7 8C 50	1498
0E40:	54 B7 90 30 00 48 00 34 16 F7 80 10 16 F7 83 90	1540
0E50:	16 F7 86 30 16 F7 86 F0 15 F7 86 D0 58 D8 66 A0	2270
0E60:	D8 B9 5F A5 D4 D9 47 C2 D4 DA 47 40 C4 D8 46 5B	2493
0E70:	D9 D8 45 B8 BA D7 56 A7 56 F8 5E A0 16 F7 86 B0	2459
0E80:	16 F7 86 50 16 F7 85 F0 16 F7 80 70 00 50 00 36	1768
0E90:	1A F7 80 10 1A F7 83 10 1A F7 85 50 1A F7 85 D0	1937
0EA0:	1A F7 85 B0 B8 B5 7E 20 B8 B5 8F A5 F8 B6 97 C2	2553
0EB0:	F4 B6 B7 DF FC B6 7F 39 E4 B6 74 A0 E8 B7 73 A0	2778
0EC0:	D1 B7 82 BF E4 B5 59 A0 90 D6 80 DF 40 D7 78 60	2575
0ED0:	01 B5 84 70 A4 B6 9C 70 E4 B6 90 5E 00 48 00 30	1808
0EE0:	95 96 58 00 D9 97 4B 5E E9 B6 43 DA D5 95 44 42	2120
0EF0:	F5 95 4D C4 DA B7 4F 46 D6 D9 4F C4 D0 DA 4D DE	2648
0F00:	C1 D9 33 58 40 B6 40 FE 01 B6 78 E2 00 B6 B8 42	2026
0F10:	00 B6 90 30 14 B5 9B 30 20 B6 A4 B0 74 B7 8C 50	1899
0F20:	40 D8 C8 40 00 38 00 30 B4 B4 40 00 B8 B4 4E 20	1498
0F30:	B9 B4 7F A2 BE B3 97 C5 FE B3 8F DF FA B2 76 DE	2938
0F40:	65 B4 65 5F 65 B5 75 B0 29 B5 AD D0 19 B4 FF D0	2323
0F50:	28 B6 EC F0 25 B6 EC 70 14 B4 D8 50 00 08 00 30	1817
0F60:	00 00 00 60 00 0C 00 30 00 00 00 60 00 00 00 60	348
0F70:	00 14 00 30 00 00 00 60 00 00 00 60 00 00 00 60	356
0F80:	00 00 00 60 FF 3B EA B8 66 B9 EC B8 55 FD E8 B8	2289
0F90:	64 DF BD B5 5C FD BC B4 64 DC FD B3 5C DE 8D B3	2840
0FA0:	5C A1 BD B4 94 C0 AD B3 98 4D FF B7 E8 70 00 00	2325
0FB0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FC0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FD0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FE0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0
0FF0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0

## VOCA III

No	Ad.Deb	Ad.Fin	Texte
00	005A	0089	DIX
01	008A	00D9	ONZE
02	00DA	0125	OUZE
03	0126	017D	TREIZE
04	017E	01E5	QUATORZE
05	01E6	0229	QUINZE
06	022A	0285	SEIZE
07	0286	02B5	VIN(GT)
08	02B6	02F9	TREN(TE)
09	02FA	0351	QUARAN(TE)
10	0352	03BD	CINGUAN(TE)
11	03BE	0411	SOIXAN(TE)
12	0412	0445	CENT
13	0446	047D	MILLE
14	047E	049D	T'
15	049E	04C5	TET
16	04C6	050D	A
17	050E	0549	B
18	054A	05A9	C
19	05AA	05ED	D
20	05EE	0639	E
21	063A	0695	F
22	0696	06F1	G
23	06F2	076D	H
24	076E	07AD	I
25	07AE	0801	J
26	0802	0855	K
27	0856	08B5	L
28	08B6	091D	M
29	091E	0985	N
30	0986	09DD	O
31	09DE	0A2D	P
32	0A2E	0A7D	Q
33	0A7E	0AF9	R
34	0AFA	0B65	S
35	0B66	0BB1	T
36	0BB2	0C01	U
37	0C02	0C5D	V
38	0C5E	0CF5	W

39	0CF6	0D81	X
40	0D82	0E21	Y
41	0E22	0E99	Z
42	0E9A	0EE9	TERMINAL=
43	0EEA	0F39	TERMINAL=

VOCA III

0000:	00 5A 00 BA 00 DA 01 26 01 7E 01 E6 02 2A 02 86	1023
0010:	02 B6 02 FA 03 52 03 BE 04 12 04 46 04 7E 04 9E	1102
0020:	04 C6 05 0E 05 4A 05 AA 05 EE 06 3A 06 96 06 F2	1186
0030:	07 6E 07 AE 08 02 08 56 08 B6 09 1E 09 86 09 DE	1005
0040:	0A 2E 0A 7E 0A FA 0B 66 0B B2 0C 02 0C 5E 0C F6	1132
0050:	0D B2 0E 22 0E 9A 0E EA FF 00 00 30 00 30 90 D8	1318
0060:	40 00 90 D8 46 00 86 D8 5E 26 F6 D9 57 C6 E5 D9	2218
0070:	57 43 F4 DA 4E BF F5 09 4E 5A E5 B9 4D B9 1A F7	2464
0080:	55 50 1A F7 55 70 1A F7 54 70 00 50 00 35 00 00	1237
0090:	00 20 41 BA FD A0 12 CA DF 24 42 CA B7 43 42 CC	1899
00A0:	87 41 42 CC BF 7F 42 CA AE DE 42 AA 9E DD 82 6A	2255
00B0:	8D 7D 81 AA 64 DE 81 AE 94 7E 58 CD 7C 40 D6 D1	2365
00C0:	61 40 E9 F5 5C 40 E1 15 5C 40 E2 03 6C 40 B2 B3	2398
00D0:	85 40 61 D4 63 C0 01 B6 90 46 00 4C 00 27 00 00	1309
00E0:	00 20 D4 B5 4C 80 54 B6 4E 0C 66 B6 57 2A 52 B6	1662
00F0:	5F AB 53 B1 57 44 52 AE 57 41 92 AD 66 DE 92 AC	2047
0100:	5D 7C D2 AD 53 7C D2 AE 52 7E 94 AE 51 DF D5 B4	2418
0110:	48 40 A6 B6 54 40 AA D6 5C C0 E6 D5 64 E0 B1 D5	2460
0120:	5C C0 11 B3 18 65 00 58 00 30 00 00 00 20 10 D2	999
0130:	9F 10 10 D2 97 90 61 B3 8F 20 21 B4 96 BF AA 94	2018
0140:	97 40 72 B4 AF C7 76 B5 9F C3 B6 B7 97 FC B7 B7	2670
0150:	8F DC F6 B7 87 DD B7 B7 7E FF E6 B7 6D 61 D9 96	2881
0160:	5C C3 D2 B7 54 5F A1 B6 54 40 51 D7 4C C0 61 D6	2225
0170:	6D 7F E2 B4 85 41 70 D5 75 5F 30 D5 60 46 00 68	1908
0180:	00 3B 00 00 00 20 05 97 97 10 05 97 97 B0 B8 96	1234
0190:	97 A0 AB 94 97 A3 85 94 7E 40 EA B4 B8 40 94 B4	2360
01A0:	79 C1 90 B4 79 A1 80 B4 81 90 11 D7 B7 B0 A1 D4	2417
01B0:	87 84 A6 D2 9F DF 66 B1 9F DE A6 B1 9F 5E A6 AF	2622
01C0:	96 7D B2 AD 94 7E 96 AC 82 FF 96 AE 8C 40 92 AF	2456
01D0:	7C C0 55 D5 64 E0 84 D6 74 40 E6 B3 84 40 91 94	2362
01E0:	73 40 50 B4 58 47 00 44 00 43 00 00 00 20 55 DA	1068
01F0:	77 B0 F6 B7 7F C0 67 B6 8F C0 62 B6 6F DF 22 B6	2493
0200:	6F FC 22 B5 66 FD 36 B5 5E 5F 75 B5 6D 7F B4 B5	2252
0210:	6C C0 F1 95 B8 C0 E2 D5 53 E0 E1 05 58 E0 A4 B5	2833
0220:	84 C0 60 D5 7C 40 50 D5 68 48 00 5C 00 3D 00 00	1443
0230:	00 20 1F F6 7C D0 1B F6 7D F0 B5 B5 8F C0 F6 B6	2404
0240:	9F C3 B6 B6 97 DE 66 B7 97 DD F6 B7 97 DE F6 B7	2979
0250:	8F 5D F5 97 87 5D F2 B7 86 5E E6 B7 75 5E F1 B6	2560
0260:	6C C0 A1 B6 5C 40 E8 D6 4C C0 55 D6 54 C0 61 D7	2400
0270:	5C 40 54 F7 5C 22 A5 D6 64 20 E4 B5 5C 20 A4 D6	2035
0280:	68 C0 34 D5 60 46 00 30 00 30 01 B5 68 00 55 B3	1376
0290:	74 41 51 D5 6C C1 14 D5 64 C2 25 D6 6C C1 66 B3	2136
02A0:	8D C2 76 B3 A7 40 62 B4 A7 41 61 B4 AF 7D 61 94	2195
02B0:	AD 5E 90 D5 78 60 00 44 00 30 11 B7 90 10 11 B7	1516
02C0:	95 90 52 96 96 10 11 96 96 10 21 B3 8D A0 56 93	1770
02D0:	8D A0 27 90 8E C3 13 8D 7F C5 23 8E 7F DF 23 8D	2008
02E0:	77 5C 23 8C 75 DA 12 8B 75 D9 53 8C 74 5F 56 AB	1903
02F0:	72 C0 51 AD 41 C1 40 AF 48 4A 00 58 00 30 05 96	1494
0300:	88 10 05 96 8C 10 09 97 8D 90 75 97 86 A0 B8 96	1807
0310:	97 2B 7B 95 A7 A0 66 93 8F C0 24 93 BF 5F 10 93	2057
0320:	9E 3C 25 92 95 3E 21 91 8D 06 32 B0 66 C4 22 AF	1670
0330:	57 3F 33 AF 56 DE 33 AE 5E 5C 23 8E 5D DD 23 8D	1762
0340:	5D 5C 32 8D 64 DF 21 AD 64 40 75 AE 68 41 40 B5	1777
0350:	60 63 00 6C 00 40 1E F7 80 10 1E F7 83 90 1E F7	1617
0360:	83 B0 1E F6 8D 70 B6 B4 9F 20 B6 B4 9F 21 66 B5	2226
0370:	7F DF 12 B6 57 C1 62 B7 57 3F 96 94 54 DF 92 98	2164
0380:	53 A0 D1 96 5A BF D1 97 48 DF 91 96 6B A0 A1 95	2410
0390:	85 1E A1 95 8D 80 AB 94 8E BF 6B 92 9F 20 23 B1	2050
03A0:	9F 3F 23 B0 86 DF 22 B0 6E 5C 32 AF 65 DE 22 BF	1927
03B0:	65 5C 71 8E 5C 5F 61 8E 5B 5F 51 AF 48 64 00 54	1572
03C0:	00 30 1A F7 F8 10 1A F7 FC 10 1A F7 FC 30 1A F7	1972
03D0:	FC F0 92 B0 7B C0 DA 8E 6B 20 67 B1 8F 48 B8 92	2456
03E0:	9F A4 BA 93 9F A0 A0 B4 86 5D 2A F7 9D D0 1A F7	2469
03F0:	9C F0 15 D6 8D 50 B6 B3 7E 22 B2 B3 97 3E 36 D1	2206

0400:	9F 5D 31 CF 8E 7C 32 AE 94 FE 70 8E 53 C2 71 8D	2025
0410:	4B 63 00 34 00 30 1A F7 C8 10 1A F7 C8 90 1A F7	1653
0420:	CB B0 1A F7 AD 70 1A F4 8D F0 B5 D2 86 A0 66 D1	2584
0430:	9F AE 22 CF BF 40 12 CD 8E DC 52 AA 84 D0 52 A0	2226
0440:	83 3E 10 AE 78 41 00 38 00 30 8A 95 40 00 8A 94	1309
0450:	4D 2F 8A 94 4E 27 51 B8 5F 3B LA B8 66 B9 EC B8	2071
0460:	55 FD E8 B8 64 DF B0 B5 5C FD BC B4 64 DC FD B3	2912
0470:	5C DE BD B3 5C A1 B0 B4 94 C0 AD B3 98 4D 00 20	2257
0480:	00 30 01 B6 78 10 01 B6 78 90 14 07 85 B0 B4 D6	1755
0490:	85 30 E5 B5 7C A0 A5 B5 8C B0 60 D6 90 70 00 28	2143
04A0:	00 30 01 B6 78 10 01 B6 7D 90 14 07 86 B0 D6 B8	1762
04B0:	6E 20 B3 B8 66 A3 B3 D8 5F 5F F3 09 6E DE F3 D9	2607
04C0:	6D A0 F3 D9 68 00 00 48 35 30 00 99 F8 10 6D B2	1710
04D0:	CE 20 69 B1 BF 20 2A B1 B7 A0 2A B1 D7 A4 2A B1	2122
04E0:	D7 A2 25 B1 D7 A1 25 B0 D7 A0 25 B1 CF 3E 2A B1	2257
04F0:	C6 BC 2A B0 CE B9 7F B0 BE 3C BF B0 B5 20 FE B0	2558
0500:	B4 2F 6A 8E B3 B0 7E AF A2 30 5E B1 E0 B0 00 3C	2072
0510:	10 2D C0 B4 30 10 05 B7 7D B0 55 96 6F 20 99 B6	1712
0520:	74 A1 9A B7 76 A2 99 B8 6E A2 99 B8 6E A1 9A B8	2451
0530:	6E A1 AA B8 66 3E EA B8 65 39 D9 B8 54 3D D9 B7	2311
0540:	5B B8 EF B7 5B 3F D9 B7 60 3B 00 60 1D 34 05 B7	1774
0550:	60 10 02 F6 59 B0 02 F6 5B 30 02 F6 5D 30 02 F5	1648
0560:	5D 30 02 F5 5D B0 02 F5 5C B0 02 F5 5C B0 02 F4	1933
0570:	5D 30 01 D4 5D B0 01 B4 5D B0 02 B4 5C B0 EA 94	1905
0580:	65 20 AA 95 76 20 9A 96 7E A0 97 97 7E BE 9B 97	2116
0590:	6E 3C DB 97 6D BC D6 97 65 3B D9 97 5C BE C7 B7	2394
05A0:	64 3C 96 98 62 BD D6 97 50 21 00 44 1C 2F C0 B4	1742
05B0:	28 10 04 B5 53 30 59 B6 75 30 69 B6 55 A0 AD B7	1696
05C0:	5D A4 ED B7 66 20 DA B7 66 20 DA B7 66 20 9A B8	2219
05D0:	66 3E 9A B8 66 3C EA B8 5D BC EA B8 5D 3C EB B8	2353
05E0:	5C 20 EA B8 5B B8 EA B8 5B 3E EF B8 51 BE 00 4C	2161
05F0:	23 2B 00 99 F8 10 1A B2 95 A0 6A 92 9E A0 FA 92	1974
0600:	8E 20 AB 92 96 A1 AB B2 97 21 AA B3 97 21 AB B2	2217
0610:	96 A1 6B B2 97 20 9B B2 8F 3F 5B B3 8F 3C AB 92	2108
0620:	86 3F AB 92 86 3D 96 93 85 BC F6 92 84 3C AA 93	2228
0630:	84 25 EA 93 82 B0 AA 93 68 30 00 5C 2C 2E 21 B3	1719
0640:	38 10 6E B4 AE A0 EF B5 A7 20 EE B5 A7 22 EE B6	2355
0650:	9F 22 EE B6 9F 20 DA B6 97 21 EB B6 8F 20 EB B5	2396
0660:	86 A0 ED B5 B5 BA AB B3 7C 3F 95 B5 7B B0 65 B6	2477
0670:	8B 30 05 B6 AC B0 14 B4 C5 B0 06 B7 E5 B0 05 B3	2073
0680:	F5 B0 25 B4 FD B0 15 B5 ED 30 14 B4 FD 30 04 B5	2240
0690:	F5 30 25 B3 B2 B0 00 5C 1E 2C 04 B0 58 00 45 B8	1502
06A0:	54 20 61 D9 3D 20 21 BA 26 A0 09 B6 26 A0 10 D9	1562
06B0:	2E A0 05 B7 2E B0 01 97 2E B0 00 B9 37 30 01 B9	1464
06C0:	37 30 11 B9 3E 30 45 B7 66 23 99 96 65 A5 97 98	1676
06D0:	66 A1 9B 98 66 A0 97 98 66 BD AB B8 5E 3C E7 B8	2350
06E0:	66 3B D7 98 65 3D EB 98 65 3C EA 98 64 BC D6 97	2277
06F0:	5B 20 00 7C 24 2F 00 75 F0 10 6F B2 CE A0 7F B2	1663
0700:	CF 22 2A B2 DF A0 6E B2 DF A0 6E B3 D7 A0 7D B3	2483
0710:	CF 20 7E B3 BE A0 BE B3 AE BE A9 B2 A5 BA 6D B3	2613
0720:	85 3F 29 B4 54 BC 1A B3 03 B0 87 B8 FD B0 45 B8	2077
0730:	FD B0 A1 DA FD B0 91 DA FE 30 91 DA FE 30 64 DA	2885
0740:	FD B0 61 DA FD B0 A4 DA FE 30 71 DA FE 30 65 BA	2777
0750:	FD B0 64 D9 FD B0 74 D9 FD B0 95 B9 FD 30 65 BA	2859
0760:	FD 30 A1 BA FC B0 61 DA FB B0 71 DA FA 30 00 40	2511
0770:	23 30 11 BA 40 01 B6 DA 3F 20 B6 DA 37 20 A6 DA	1717
0780:	3F A0 A2 DA 3F 20 A2 DA 47 20 B2 DA 3F 20 B2 DA	2068
0790:	47 3F F2 DA 46 BF E3 DA 46 BE F2 DA 3D BF F2 DA	2732
07A0:	3D 3A E4 DA 34 3E C0 D9 3B 30 05 B0 01 B0 00 54	1637
07B0:	1B 37 09 B9 78 00 DA B8 74 A0 21 DA 6E 20 22 BA	1479
07C0:	67 20 06 B8 5F B0 06 B8 5F B0 12 D9 57 B0 56 BA	1827
07D0:	57 B0 96 B9 4E 20 A6 B9 4E A0 E6 B9 46 A1 E6 B9	2358
07E0:	46 A0 E6 B9 46 A0 E6 B9 3E BE F7 B9 46 3D EB B9	2525
07F0:	3D 3B E5 B9 3C BD E1 B9 3C BA E1 B9 3B A0 EB B9	2488
0800:	3A 36 00 54 20 34 00 B1 EA 10 94 B5 FD 10 54 B4	1569
0810:	FE B0 24 B4 DE 00 25 B4 96 00 AA 94 7F 20 AA B3	2013

0820:	9F 20 6A B3 AF 3F 6A B2 B7 A0 6A B2 BF A0 67 B2	2259
0830:	B7 20 6A B1 B7 3E B6 B1 B6 BE FA 91 A5 BD BF 90	2510
0840:	AD B8 B7 B0 AD B0 AF 90 B5 3C 6C B0 B3 B0 59 AF	2542
0850:	AB 30 69 B1 B8 30 00 60 2A 2C B1 77 F8 10 6E B5	1715
0860:	A6 A0 AB B6 A7 22 FA 96 A7 22 AA B5 9F 21 D6 B6	2404
0870:	9E A2 EA B5 97 22 EA B6 B7 20 EF B5 B6 BF FE B5	2682
0880:	7L BE AB B4 76 BF A7 B4 6E BD A7 B4 66 BF A7 B4	2606
0890:	66 3E 97 B4 65 BE D7 B4 65 3D D2 B4 64 BD C7 B6	2390
08A0:	6C 3F D2 B4 6C BE E7 B3 64 A0 F7 B3 64 26 E3 B3	2499
08B0:	58 A7 E6 B2 59 A0 00 68 22 30 80 88 78 10 AF B5	1905
08C0:	97 20 FF B5 9F 20 AD B5 9F A0 EB B5 9F A0 EE B6	2622
08D0:	97 21 DA B5 97 A2 EE B5 B7 20 AB 94 7F 20 5E 93	2201
08E0:	65 BF 5E 92 46 3E 5E 92 26 20 2E 92 06 20 6E 92	1457
08F0:	25 BE 6E 92 3D 3D AF 91 3D 3E AE 91 34 BF 99 91	1908
0900:	2C 3C BE 91 2C 3F 6D 91 34 3E 6F 91 34 20 6E 91	1509
0910:	44 3F 6E 91 2C 3F AA 91 33 A6 6A 90 00 20 00 68	1411
0920:	1E 28 20 B3 00 10 99 B6 A7 20 E9 B6 A7 23 AD B6	1803
0930:	9F A0 E9 B6 9F A2 EC B6 9F 21 D8 B6 97 20 E9 B6	2661
0940:	9F 22 E9 B6 BF 20 E8 B5 B6 A0 E8 B5 76 20 99 B5	2387
0950:	55 3C 88 B6 45 3F 88 B7 45 3F C9 B5 3C BE C8 B6	2060
0960:	34 BE C9 B5 34 A0 89 B4 3C BD 89 B5 3C BF 88 B5	2288
0970:	44 3C 89 B5 43 A0 C8 B5 3E A0 C8 B5 33 20 EC B3	2152
0980:	2A A0 AE B3 30 30 00 58 31 30 44 B1 B9 10 8C AE	1596
0990:	7C A0 49 AA 8D A0 49 AB 96 A0 45 AB 9E A0 45 AC	2178
09A0:	96 A0 85 B0 7E A0 85 B1 76 A1 86 AE 7E 21 B6 AC	2267
09B0:	7E 20 C6 AC 76 20 C6 AC 76 20 C6 AC 76 20 C6 AC	2088
09C0:	76 20 86 AB 74 3E 85 AA 7E 3E C5 A7 76 3E C5 A5	2032
09D0:	75 BD 85 A4 74 BF 85 A2 7B BF 04 C2 62 AA 00 50	2065
09E0:	25 3B 00 B5 00 10 00 B3 EA 10 04 B2 05 10 4B B6	1179
09F0:	5D B1 D9 94 6E 13 D9 96 7F 3F D9 B6 7F 20 99 B7	2167
0A00:	7F A1 DA B7 77 A1 9A B8 77 A0 99 B8 77 3D 9A B7	2440
0A10:	6F 3E DA B7 6E BC DA B7 65 BE ED B7 65 BB D9 B7	2672
0A20:	65 3E CA 94 5B A1 C9 B5 52 A0 C8 B7 50 30 00 50	1980
0A30:	1E 30 4B 97 48 10 50 B8 FE 10 50 B8 FF 90 10 B7	1785
0A40:	CE B0 10 97 BD 30 64 97 45 A0 95 97 4E 20 DA 96	1996
0A50:	4E 22 97 96 4E 21 98 96 46 A1 9B 96 46 20 9B 96	1772
0A60:	3E 3F 97 97 46 3C DB 96 4D 3D D7 97 3C BC DB 96	2047
0A70:	3C 3F CB 96 38 38 E2 98 32 BF D0 D7 32 30 00 7C	1855
0A80:	22 2D 60 B5 40 10 55 B5 B6 B0 AB B6 A7 A2 99 96	2045
0A90:	9F A1 EA B6 97 21 D6 B6 9F 22 DB B5 97 20 D5 96	2455
0AA0:	97 21 DA 96 BF 20 D9 B6 BF 3F D9 96 BF 3E DA 97	2273
0AB0:	BF 3E DA B7 BF 3F DA B7 87 3D DA B7 86 BE EA 96	2518
0AC0:	B6 3F 96 97 85 BD 9A B7 85 BC C5 95 85 3E D6 96	2383
0AD0:	85 20 EA 95 84 BF DA 96 8C BF DF 96 84 BF DB 96	2635
0AE0:	8D 3F EF 94 95 BF AF B3 95 22 AA B2 9C A3 FE B2	2567
0AF0:	9C A3 FE B1 93 20 BE B0 90 30 00 6C 29 2F 00 93	1830
0B00:	F8 10 F9 B5 9E A0 EA B5 9F A0 AE B5 9F A1 EE B5	2840
0B10:	9F A2 E9 B5 97 A1 E9 95 97 20 EA B5 B6 A0 D6 B5	2716
0B20:	7E 20 D9 B6 7E 3E D9 B7 75 B8 CA B6 64 A0 51 F7	2418
0B30:	64 B0 11 F7 9C B0 11 F7 9C B0 01 F8 D4 B0 01 F8	2354
0B40:	DC B0 01 F8 EC B0 01 F8 FC B0 01 F8 EC B0 01 F8	2644
0B50:	F4 B0 01 F8 FC B0 01 F8 FC B0 01 F8 FC B0 01 F8	2700
0B60:	F8 30 01 F8 F9 B0 00 4C 1D 37 00 93 00 10 45 99	1518
0B70:	F6 30 4E B5 75 A0 AD B6 5E 3E DB B6 66 3E D9 B7	2303
0B80:	6E A0 9A B7 66 BF A9 B8 66 A0 99 B8 66 A1 AA B8	2469
0B90:	66 BE AA B8 66 BC DA B8 5E 3D EE B8 4D 3C E8 B8	2468
0BA0:	54 3E DA B7 53 3C CB 97 4A BD CA 96 4A 20 CA B7	2150
0BB0:	50 B0 00 50 1F 22 44 B6 38 10 D9 96 4C A0 96 98	1628
0BC0:	55 AF 5A 97 46 23 9A 97 4E A0 96 97 4E A0 EA 96	2072
0BD0:	4D A0 96 97 46 20 96 97 46 21 96 97 3E 20 96 97	1740
0BE0:	46 3F D6 97 45 BE 96 97 45 BD D6 97 45 3B D9 96	2176
0BF0:	3C 3C D1 96 43 20 C0 97 42 B9 C4 97 39 A0 C4 96	2082
0C00:	39 30 00 5C 24 1E 00 99 F8 10 40 F4 73 60 95 B3	1527
0C10:	4F A6 19 94 04 A0 08 15 04 A0 09 93 04 A0 19 93	1417
0C20:	05 21 08 92 55 21 19 93 05 A7 JA 95 66 24 5A 95	1206
0C30:	71 25 9A 97 7F A1 DA B7 7F 21 D9 B7 7F 21 DA B7	2279

0C40:	7F 20 DA B7 7F 3F EA B7 76 BB DA B7 65 BF DA B8	2567
0C50:	65 3B EA B8 65 3D D9 B7 5C 20 C9 B7 4B 30 00 9B	1920
0C60:	2B 24 B5 B8 2B 00 02 95 FE A0 22 B5 06 AF 5A 94	1635
0C70:	46 A4 B6 92 46 71 BA BE 3D 20 D5 90 33 A0 C1 96	1901
0C80:	32 A0 80 97 3A 3A 80 A0 2A 3F 90 AE 2A 30 65 90	1664
0C90:	2C A0 6B 92 4D A3 69 93 46 20 6C 93 46 A2 9D 93	1839
0CA0:	67 20 AE 93 76 BF 9B 93 76 BE 5A 93 66 BE 6B 92	2157
0CB0:	65 BD AA 92 65 3D 69 92 5C 3E 55 B3 2B A0 55 B4	1905
0CC0:	3B 20 04 B5 4B 20 59 94 64 20 AA 94 6D A2 DA 95	1708
0CD0:	6E 20 DA 96 6E 20 DA 97 6E 20 DA 96 6E 3E DA 97	2072
0CE0:	65 BD D6 B7 65 20 DA B7 65 3C DA B7 64 23 DA B8	2320
0CF0:	5B 26 CA B4 59 B0 00 BC 29 2C 95 B9 50 10 FA B9	1868
0D00:	4D A0 75 B9 4E A2 75 B9 5F 20 70 BA 56 A1 60 BA	2035
0D10:	57 22 A2 BA 4F 21 F7 BA 3D A1 F2 BA 2B BC D0 DA	2321
0D20:	2A 3B D1 BA 29 30 B0 B8 30 30 B4 B7 30 30 40 B6	1647
0D30:	30 30 01 F4 32 B0 01 F4 36 30 01 F4 35 B0 01 F4	1633
0D40:	35 B0 01 F4 3D 30 01 F4 3D 30 01 F4 3D 30 01 F4	1536
0D50:	45 30 01 F4 45 30 01 F4 45 30 01 F4 44 B0 01 F4	1575
0D60:	4C B0 01 F4 4C B0 01 F4 4C B0 01 F4 4C B0 01 F4	1988
0D70:	54 B0 01 F4 54 B0 01 F4 54 30 01 F4 5A 30 01 F4	1770
0D80:	5B 30 00 A0 26 39 16 B9 68 10 F7 D9 3E 20 F7 B9	1708
0D90:	35 A0 F3 B9 35 20 F1 99 34 BF CC 99 3B 3D E1 B9	2250
0DA0:	2A 3B 50 B9 24 B0 0B 93 07 30 09 96 05 30 59 96	1252
0DB0:	6E 20 9D B5 7E A2 6C B3 BE A0 5E B3 B6 BA 6E B2	2238
0DC0:	86 BF E9 B2 BE A2 ED B3 96 A0 ED B4 9E A0 DA B4	2899
0DD0:	9E A0 FE B5 96 BF DE B5 BE A0 96 97 7E A0 97 9B	2689
0DE0:	5D BF CA 97 4B BE C4 93 59 B0 D4 B5 59 30 90 D5	2397
0DF0:	4B 30 90 D6 4B 30 90 D6 4B 30 90 D6 4B 30 90 D6	1912
0E00:	4B 30 90 D6 4B 30 04 B7 AB 30 08 97 FD B0 0C B4	1784
0E10:	ED 30 0B 9A CB B0 3B B8 A3 B0 19 B8 B3 30 A5 B7	2141
0E20:	70 30 00 7B 2B 21 05 B3 4B 00 00 B8 52 A0 05 96	1193
0E30:	5C 20 00 97 5D A3 02 B4 66 A4 19 B4 67 2A BC B5	1698
0E40:	77 24 E8 B4 87 A1 FD B4 87 BF FD B5 7F 3E FD B5	2679
0E50:	7F 20 FE B5 7F 3F E9 B5 76 B9 FD B4 56 20 D5 B4	2445
0E60:	54 3C D0 B4 3A B0 D0 D5 39 B0 90 B5 39 B0 D0 B5	2367
0E70:	29 30 D4 B5 29 B0 C0 B4 39 30 FA B4 5D 23 AE B4	2088
0E80:	7D 20 AE B4 7D 20 EA B4 85 A0 EE B4 85 30 FF B3	2408
0E90:	8D 30 EE B3 8B B0 B9 B3 6A B0 00 50 00 2F 01 D6	1909
0EA0:	8B 10 01 D6 BE 90 51 B7 7D A0 A7 B7 76 BE A7 B8	2211
0EB0:	7F DD A5 B6 97 DF 55 D5 97 20 46 B4 B6 5F 49 92	2248
0EC0:	4C A0 B8 94 4C BF 41 96 4D BF 61 D8 4E 3F A2 D9	2103
0ED0:	46 41 A3 B8 45 41 D9 95 4C 41 76 B2 B6 E0 F0 B2	2243
0EE0:	74 E1 D1 B3 5A 61 A8 B4 60 04 00 50 00 30 01 D6	1707
0EF0:	8B 10 01 D6 BE 90 51 B7 7D A0 A7 B7 76 AF A7 B8	2196
0F00:	7F C1 A5 B6 97 DE 55 D5 97 3E 46 B4 B6 40 49 92	2218
0F10:	4C BF B8 94 4C A2 41 96 4D A0 61 D8 4E 3D A2 D9	2072
0F20:	46 5E A3 B8 45 5D D9 95 4C 5E 76 B2 B6 FF F0 B2	2360
0F30:	74 FE D1 B3 5A 7F A8 B4 60 4A 6E DF FA B2 76 DE	2594

— BRUITS + DEMO. Ce bloc d'environ 2,5 Ko fournit, outre trois phrases de démonstration dont deux par une voix féminine, divers bruits pouvant trouver leur application dans des jeux, et un exemple de musique simple.

No	Ad.Deb	Ad.Fin	Texte
00	00E4	0363	ICI LE SYNTHETISEUR DE PAROLE MEA 8000
01	0054	005F	BRUIT 1 (TUT)
02	0060	006B	BRUIT 2 (PIP)
03	006C	009B	BRUIT 3 (COUP DE FEU)
04	009C	00B3	BRUIT 4 (TLUUIIT)
05	0020	002B	BRUIT 5 (SIFFLET)
06	002C	0053	BRUIT 6 (WOODWOODWOOD)
07	0394	03AB	LOCO 1 (LONG)
08	0364	037B	LOCO 2 (MOYEN)
09	037C	0393	LOCO 3 (COURT)
10	03AC	065F	MUSIQUE (MICHAEL)
11	0660	084B	VOTRE CORRESPONDANT EST ABSENT
12	084C	0A2B	VEUILLEZ RAPPELER ULTERIEUREMENT

BRUITS+DEMO

0040:	00 84 00 54 00 60 00 6C 00 9C 00 70 00 2C 03 94	851
0010:	03 64 03 7C 03 AC 06 60 08 4C FF FF FF FF FF 00	1866
0020:	00 0C 00 10 F0 0F FC 60 F0 01 FC 60 00 28 00 40	1338
0030:	00 00 85 60 00 00 85 6F 00 00 85 71 00 00 85 6F	963
0040:	00 00 85 71 00 00 85 6F 00 00 85 71 00 00 85 6F	980
0050:	00 00 85 60 00 0C 00 80 E0 00 40 E0 E0 00 40 E0	1393
0060:	00 0C 00 F6 3C BE FA 60 3C BE FA 60 00 30 00 80	1620
0070:	FF 00 07 90 FF 00 07 F0 FF 00 00 70 FF 00 06 70	1648
0080:	FF 00 00 70 FF 00 04 70 FF 00 03 70 FF 00 02 70	1477
0090:	1F 00 01 70 FF 00 00 70 FF 00 00 70 00 18 00 40	1190
00A0:	FC 00 05 70 FC 24 26 70 FC 6C 67 70 FC 84 A7 76	2093
00B0:	FC FC E4 70 02 B0 1F 31 86 D8 70 01 86 D8 74 88	2167
00C0:	B6 D9 57 27 A6 D9 4F 41 A6 D9 47 5F B7 D9 47 3D	2133
00D0:	A0 D8 5E DF 05 F8 56 F0 05 F8 5D F0 E6 D8 65 C0	2397
00E0:	F1 B8 4E 40 F0 D9 4E C1 E1 D9 47 41 E4 DA 47 41	2455
00F0:	E8 DA 46 40 D8 DA 4C 5F 61 B6 30 5E 74 B5 3A 30	2013
0100:	74 B5 54 20 74 B6 7E 5E 74 B5 26 21 28 B5 56 42	1592
0110:	69 B4 76 A0 6A B4 76 C0 68 B4 77 40 AB B4 6E DE	2312
0120:	EB B4 5A BD 42 D7 55 42 23 F8 55 10 06 F7 56 10	1865
0130:	06 F6 5E 30 02 F6 66 D0 02 D6 6F 50 65 B5 6F 2F	1799
0140:	B9 B4 9F 2F B8 B3 AF 2A 69 B3 A7 25 AB D4 9F 3F	2241
0150:	A5 B4 7F 20 40 B4 71 5F 54 D3 78 3F 55 F4 75 10	1896
0160:	16 D7 5E 90 E9 B8 66 DF 85 B8 66 BF D4 B8 44 DE	2513
0170:	99 B7 38 3F C5 B7 4A BF 81 B7 4A 9E 09 F7 4B 10	1994
0180:	4C F8 40 10 08 F8 4D 90 8D D9 4D 9B 84 DA 45 DF	2126
0190:	D5 D9 4D C1 D9 B8 5D BA E6 B7 45 D8 5A F6 44 B0	2661
01A0:	56 F5 4D 30 AA F5 5D 9A 99 B5 77 28 EA B4 BF C3	2363
01B0:	5A B2 87 BE 1A B0 9F BF 15 AE 87 57 68 CE 76 3C	2053
01C0:	95 CE 53 BC 99 AD 5A A9 56 D4 5C AF 9B B4 5E AD	2378
01D0:	8A B4 7E A4 95 B2 7E BF 96 B1 53 5F E6 B1 62 3F	2325
01E0:	80 B3 60 5E 80 B3 60 02 15 B5 76 90 15 B5 86 90	1846
01F0:	55 B4 8D 9C 6A B1 AF 3E 26 B1 CF DF 26 B1 CF 9E	2307
0200:	2E B0 CF 80 57 B0 BF 36 6A B0 A6 B8 69 B0 9E 3C	2193
0210:	6A AF 9E 3D 6A AF BF 42 6E B0 A7 40 69 B1 9F 40	2060
0220:	AB B2 86 41 FC B3 5D 62 FC B3 54 FF FC B3 5D 23	2496
0230:	8D B3 8E C0 FD B3 9E C0 BE B3 96 5F 7E B3 95 39	2609
0240:	99 B3 92 5E D4 B3 91 22 D5 B3 98 7C D5 B3 98 60	2450
0250:	D5 B3 98 60 D5 B3 98 60 D5 B3 98 60 D5 B3 98 60	2560
0260:	D5 B3 98 60 65 B5 B4 CF B6 B6 A7 2F AA B6 9F AF	2573
0270:	AA B6 8F A0 BA B6 87 BF BA B6 97 BF AA B6 97 BF	2753
0280:	AB B4 7F 3E 9A 94 55 BF 6E B3 35 B0 6E B3 4D BE	2205
0290:	6E B3 45 BD 6E B2 45 BE AE B2 45 DD AE B2 44 5E	2250
02A0:	AB B1 54 3D FA B2 6A 5F 65 B2 71 30 65 B2 70 10	1969
02B0:	65 B2 70 70 00 B3 80 30 66 B3 AC 30 68 B3 9E 3C	1863
02C0:	AF B3 97 5F 6E B3 87 7E B8 B3 8E A2 FB B3 88 5E	2480
02D0:	D6 B2 81 A0 C6 B1 79 9E 29 B0 90 62 2D B1 BD A4	2369
02E0:	6F B1 D6 C1 AE B1 CE DF 69 B1 D7 3F 6E B1 CF 5F	2624
02F0:	BF B1 BE FF 6A B1 AC C0 A5 AF 99 C0 A0 B0 90 27	2664
0300:	A0 B0 90 60 A0 B0 90 60 A0 B0 90 60 56 B7 59 7D	2211
0310:	D7 96 43 A4 E7 95 44 5F B6 B5 45 C5 A7 97 46 2B	2199
0320:	57 B9 3E AE 4A B8 2F 25 5A B8 47 AA A6 B8 36 DE	1991
0330:	90 99 45 3A 8A 95 3C 8A 8A 94 4D 59 51 B8 5F 3B	1828
0340:	EA B8 66 B9 EC B8 55 D8 E8 B8 64 DE BD B5 5C FD	2882
0350:	8C B4 64 FE FD B3 5C FF BD B3 5C A1 BD B4 94 C0	2831
0360:	AD B3 98 45 00 18 80 80 6F 00 05 D0 6F 00 05 50	1373
0370:	6F 00 03 50 6F 00 02 50 6F 00 01 50 00 18 80 80	859
0380:	6F 00 05 B0 6F 00 05 30 6F 00 03 30 6F 00 02 30	779
0390:	6F 00 01 30 00 18 80 80 6F 00 05 F0 6F 00 05 70	1024
03A0:	6F 00 03 70 6F 00 02 70 6F 00 01 70 02 B4 82 82	1117
03B0:	00 03 9D 60 00 03 9D 60 00 03 9D 60 00 03 9D 60	1024
03C0:	00 00 00 2F 00 00 00 49 00 07 BD 60 00 07 BD 60	704
03D0:	00 07 BD 60 00 07 BD 60 00 00 00 2F 00 00 00 49	704
03E0:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
03F0:	00 0A D5 60 00 0A D5 60 00 00 00 31 00 00 00 58	775

0400:	00 07 BD 60 00 07 BD 60 00 00 00 2E 00 00 00 48	702
0410:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0420:	00 00 00 2F 00 00 00 2A 00 0C E5 60 00 0C E5 60	763
0430:	00 0C E5 60 00 0C E5 60 00 00 00 31 00 00 00 37	778
0440:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0450:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0460:	00 00 00 35 00 00 00 7B 00 07 BD 60 00 07 BD 60	760
0470:	00 07 BD 60 00 07 BD 60 00 00 00 2E 00 00 00 48	702
0480:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0490:	00 00 00 2F 00 00 00 2A 00 0C E5 60 00 0C E5 60	763
04A0:	00 0C E5 60 00 0C E5 60 00 0C E5 60 00 0C E5 60	1348
04B0:	00 0C E5 60 00 0C E5 60 00 0C E5 60 00 0C E5 60	1348
04C0:	00 0C E5 60 00 0C E5 60 00 0C E5 60 00 0C E5 60	1348
04D0:	00 0C E5 60 00 0C E5 60 00 00 31 00 00 00 7E	849
04E0:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
04F0:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0500:	00 00 00 34 00 00 00 7E 00 07 BD 60 00 07 BD 60	759
0510:	00 07 BD 60 00 07 BD 60 00 00 00 2E 00 00 00 48	702
0520:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0530:	00 00 00 36 00 00 00 2B 00 0A D5 60 00 0A D5 60	735
0540:	00 0A D5 60 00 0A D5 60 00 0A D5 60 00 0A D5 60	1276
0550:	00 00 00 31 00 00 00 5B 00 07 BD 60 00 07 BD 60	721
0560:	00 00 00 29 00 00 08 C5 60 00 08 C5 60 00 08 C5 60	944
0570:	00 08 C5 60 00 00 00 36 00 07 BD 60 00 07 BD 60	939
0580:	00 07 BD 60 00 07 BD 60 00 00 00 31 00 00 00 3D	694
0590:	00 05 AD 60 00 05 AD 60 00 05 AD 60 00 05 AD 60	1096
05A0:	00 05 AD 60 00 05 AD 60 00 05 AD 60 00 05 AD 60	1096
05B0:	00 00 00 2F 00 00 00 66 00 00 73 00 03 9D 60	520
05C0:	00 03 9D 60 00 03 9D 60 00 03 9D 60 00 00 00 2C	812
05D0:	00 09 00 21 00 05 AD 60 00 05 AD 60 00 05 AD 60	855
05E0:	00 05 AD 60 00 00 00 2F 00 00 00 23 00 07 BD 60	648
05F0:	00 07 BD 60 00 07 BD 60 00 07 BD 60 00 07 BD 60	1168
0600:	00 07 BD 60 00 07 BD 60 00 07 BD 60 00 00 00 31	925
0610:	00 00 00 3D 00 05 AD 60 00 05 AD 60 00 05 AD 60	883
0620:	00 05 AD 60 00 05 AD 60 00 05 AD 60 00 05 AD 60	1096
0630:	00 05 AD 60 00 00 00 31 00 00 00 3F 00 03 9D 60	642
0640:	00 03 9D 60 00 03 9D 60 00 03 9D 60 00 03 9D 60	1024
0650:	00 03 9D 60 00 03 9D 60 00 03 9D 60 00 03 98 20	955
0660:	01 EC 85 67 00 99 F8 10 15 B3 0A B0 01 B0 03 20	1488
0670:	59 B2 64 20 A9 B0 7D 28 F9 B1 96 28 A9 B1 9F 22	2064
0680:	A5 B1 9F 20 A8 D2 97 20 E8 D1 8D A1 54 B3 89 30	2285
0690:	54 B3 88 50 00 B3 8D B0 40 F9 8F B0 12 01 8E B0	2152
06A0:	12 D1 8D B0 95 B1 96 22 D9 B1 9E A2 D5 B3 96 BF	2501
06B0:	04 B4 96 BF 98 B3 8D BC 54 B1 7A B0 00 8D 98 50	2325
06C0:	00 8D 98 30 91 B4 FC 30 61 82 E5 B0 98 AE 8D 2F	2160
06D0:	A8 AF 9D A3 A4 AE A6 22 A5 AC A6 22 BA AC 9E 20	2286
06E0:	A9 AC 9D BE 96 CE 95 C2 A6 D0 9D A3 A5 B2 A6 A7	2623
06F0:	E4 D5 AF 21 A8 D6 AF 3F 68 D7 9D B8 A4 D7 5C B4	2583
0700:	12 D8 54 B0 12 D9 65 30 12 F6 7D D0 02 D5 93 D0	2045
0710:	00 B7 BA 30 00 B8 E3 B0 14 D6 EE B0 18 D5 A5 B0	2222
0720:	45 B1 85 A6 6A AB 95 A2 BA AB 8D A0 71 AB 95 20	2256
0730:	72 B8 95 20 62 6C A5 A1 51 8E A6 20 60 CF 75 8E	1997
0740:	A6 B0 4D 3A 81 B2 4C BE 82 D6 4C 3E B4 D7 5B 31	2019
0750:	51 B7 B4 30 91 D5 B5 AE E8 D3 9E A7 D4 D3 A6 A0	2626
0760:	E8 B1 A6 BE 99 B0 AE BF A9 AE AE A0 A9 AE AE 3F	2716
0770:	B9 AE AE 3F B2 AE AD A0 A2 AE AD A1 A3 8D AD 22	2462
0780:	53 BC B5 A1 53 8D 8D BE 52 8E AD 22 B5 AE A4 20	2150
0790:	45 B0 B0 40 45 B0 B0 55 98 B9 80 35 D4 D9 7D BB	2263
07A0:	E8 DA B5 AF EC DA 85 24 E8 D9 85 3F 90 D8 7B 3E	2571
07B0:	80 B4 78 50 80 B4 78 50 44 D9 8C 30 94 D7 95 27	2040
07C0:	98 D5 AE 20 E2 B5 C5 8D E8 B5 C6 3F E4 B5 C6 3F	2708
07D0:	94 B3 AD 20 50 B3 58 50 50 B3 58 50 01 D6 5C B0	1869
07E0:	01 D7 5D B0 01 D8 65 30 02 F7 65 30 02 FA 6D 30	1658
07F0:	12 F8 6D 30 02 F8 75 30 02 F7 74 B0 02 F6 7D 30	1800
0800:	01 F6 7D 30 01 F7 65 B0 11 F5 85 30 F4 D4 B0 A0	2161
0810:	9D B3 A6 3L 94 B2 B6 3D 98 F0 B5 BE 94 AF B5 3F	2394

0820:	A0 BE AD 20 50 AD A4 A3 A2 8C 9C A2 92 8C 9C 22	2183
0830:	90 8C A4 24 50 A9 E4 26 61 A9 94 26 64 AA 93 A4	1984
0840:	98 CF 93 24 54 CB 89 3F 25 E4 70 70 01 F0 5D 5E	1882
0850:	25 E4 74 70 69 E4 7C A0 99 E4 7C 20 56 E5 74 20	1918
0860:	05 E4 5B BF 05 E3 64 3D 15 E4 7C A7 59 E3 85 21	1738
0870:	A9 E4 95 2D E9 E4 96 2C AE E5 9E A7 EA E6 9F 23	2392
0880:	5D B7 BF 22 55 B8 6F 20 A4 D9 56 3B A4 FA 45 ED	2063
0890:	E4 FA 45 20 64 FA 45 20 E8 FA 4D A0 E4 DA 56 2A	2179
08A0:	74 DA 6F 22 34 DA BF 20 28 DA 9F 20 68 DA A7 20	1894
08B0:	64 D9 A7 20 94 D8 A6 BF D8 D7 A6 BF A8 D6 A6 3D	2634
08C0:	54 D6 A5 E1 65 D5 A5 30 A5 E4 E5 E0 98 E4 CE A9	2576
08D0:	98 B3 D6 A0 9C E3 D6 BF 9C E3 CE BF 9C E1 A5 EC	2863
08E0:	54 E2 62 E0 10 D4 90 50 40 E5 CB 30 45 D5 CB E0	2145
08F0:	40 D9 FC 30 00 E5 74 A0 98 E2 95 A0 9C E4 96 23	2198
0900:	FD E5 96 A0 BD E5 96 3F 65 96 75 3C 50 97 4D 3C	2123
0910:	91 E6 44 BF 99 E6 44 BF A4 D8 7D A1 A8 D8 BE A5	2537
0920:	AB D8 96 A2 98 D9 96 A0 94 D9 96 A2 94 E9 96 A0	2695
0930:	94 D9 9E A2 94 D9 9E A2 A4 D9 96 A3 A4 D9 96 A3	2758
0940:	64 DA 96 A3 68 DA 9E A2 68 D9 BE 3E F8 E9 75 BE	2538
0950:	F9 E9 74 BE BD E8 E4 30 E9 E8 68 30 AD E7 64 30	2321
0960:	6C E7 65 38 74 E7 5D A1 A5 E7 55 A0 90 E9 4C BE	2189
0970:	E1 E9 3C EA A2 D9 3C BB 20 D9 04 A0 21 B8 13 30	1979
0980:	00 E8 F8 30 00 E8 F8 50 40 D8 C3 E0 48 D8 9C A3	2250
0990:	A3 D8 9D A9 E8 D8 9E 24 D8 D7 A6 A0 98 E6 BE BF	2819
09A0:	94 D5 BE A0 98 E3 AE 3D A4 E5 9E 3F D8 E6 96 21	2424
09B0:	D8 E8 BE 20 A8 D9 75 BE A8 D9 65 BF 94 DA 6E 20	2451
09C0:	94 D9 7E A0 98 E8 BE A0 E8 E6 97 20 E8 E6 9F 3F	2522
09D0:	98 E4 AF 3F 58 E1 E6 BF A9 AF A6 3D 5A E1 95 37	2250
09E0:	05 D0 9D 30 E4 D1 7C 30 E0 94 7C 30 05 BE 04 23	1565
09F0:	14 BE 04 A1 11 EC 04 BF 02 EC 04 BE 02 68 04 3E	1190
0A00:	44 E0 94 ED 90 ED EC BF 91 AF A4 BE 98 AF AC ED	2559
0A10:	96 AF AC BF 94 AE A4 BF 91 BE 9C 3F 91 BF E6 A0	2460
0A20:	00 AF E2 BF 21 49 C2 30 41 97 C0 30 00 00 00 00	1300

— CHIFFRES ANGLAIS. On trouvera dans ce bloc d'environ 2 Ko les éléments permettant de composer tout nombre entre 0 et 999 en anglais, ce qui est beaucoup plus facile qu'en français étant donné l'absence de liaison entre les mots. En outre, 4 mots pouvant être utilisés avec ces chiffres sont fournis.

No	Ad.Deb	Ad.Fin	Texte
00	0032	006D	0 (ZERO)
01	006E	00B1	ONE
02	00B2	00F5	TWO
03	00F6	0141	THREE
04	0142	0189	FOUR
05	018A	01E1	FIVE
06	01E2	0241	SIX
07	077A	07E5	SEVEN
08	0242	0285	EIGHT
09	0286	02CD	NINE
10	02CE	031D	TEN
11	031E	0399	ELEVEN
12	039A	03F5	TWELVE
13	03F6	043D	THIR-
14	043E	0475	FOR-
15	0476	04BD	FIF-
16	04BE	0521	-TEEN
17	0522	0565	TWEN-
18	0566	05A5	-TY
19	05A6	0605	HUNDRED
20	0606	0685	TELEPHONE
21	0686	06E9	NUMBER
22	06EA	0739	CODE
23	073A	0779	AND

CHIFFRES ANGLAIS

0000:	00 32 00 6E 00 B2 00 16 01 42 01 8A 01 E2 07 7A	1146
0010:	02 42 02 86 02 CE 03 1E 03 9A 03 F6 04 3E 04 76	1039
0020:	04 BE 05 22 05 66 05 A6 06 06 06 86 06 EA 07 3A	968
0030:	FF FF 00 3C 29 36 89 91 88 00 95 91 83 20 E4 90	1912
0040:	95 A2 E4 90 96 25 F8 90 96 C1 ED 90 97 5F E8 91	2609
0050:	8E FF E9 90 86 DE DA 90 7D 7F DE 8F 7D 20 C5 8E	2605
0060:	74 FE D5 8D 73 E0 D9 8D 7B F0 D4 80 70 40 00 44	2400
0070:	2E 36 20 A0 70 00 95 85 6E 40 9A 89 7D 41 9A 8C	1629
0080:	96 C2 A9 8E 9F 41 AA 8F 9F 5E AB 8F 97 3F AB 8F	2281
0090:	96 DE 80 91 64 DD 80 6C 6C 40 C0 8D 6E 7F 80 8E	2211
00A0:	6A E0 80 B3 62 40 95 B3 84 20 F4 B3 7A C0 50 B1	2269
00B0:	91 E0 00 44 2C 44 11 D7 C7 30 01 D8 97 30 42 B9	1695
00C0:	FF 50 8A B7 FE B0 36 B5 25 B0 ED 94 76 A0 EA 94	2579
00D0:	76 DE EA 94 6E DE EF 93 66 5E EA 92 65 DE EE 92	2723
00E0:	6C DE D5 90 63 C0 D4 HE 63 5E D9 8E 63 60 D5 8D	7433
00F0:	63 40 D5 AE 6A C0 00 4C 19 3E 40 B9 F8 10 45 B9	1778
0100:	FA 50 40 B9 FC D0 45 B9 F6 50 55 B9 C6 B0 56 B6	2531
0110:	9E 90 A6 93 84 20 E4 93 6D 3F A0 93 75 DE A5 94	2285
0120:	6E E0 E5 B7 6F 7F E4 B8 66 7E E4 B9 65 40 E4 B9	2615
0130:	5C 7F E4 B9 5C 40 E1 D9 53 40 E0 BA 62 FC 84 B7	2452
0140:	71 40 00 48 2D 3E 00 B8 F8 10 10 BA FC 10 05 B5	1460
0150:	FD 00 84 B5 FD D0 85 B3 E5 30 90 AB 8C A0 90 AA	2785
0160:	8D C0 90 AA 8D C0 90 A9 8D FF A5 A9 8D DE AA A9	2725
0170:	8D DF E9 A9 8D 7F E9 AA 8C DF F8 AA 84 40 B5 AA	2765
0180:	7A C0 A4 AA 79 40 54 AB 80 41 00 58 29 3E 55 B6	1739
0190:	78 10 09 B7 FD 50 15 B4 FE 70 45 B6 FE 70 55 B5	2111
01A0:	FE 30 99 8F 9E C0 AE 8E 9F 5F AD AF 9F 40 AC 80	2437
01B0:	9F 7F AC B0 9E DF AB B3 9E 7E E8 B4 9D DF E8 B4	2850
01C0:	8D C0 E4 B5 7C 7F E8 B5 74 40 69 B5 7B B0 89 B5	2489
01D0:	6B D0 E4 B2 7B 50 E4 B2 7B B0 D4 B2 7A A1 91 B4	2627
01E0:	71 40 00 60 32 46 01 D8 F8 10 05 D8 FC 30 06 D8	1626
01F0:	FD 50 05 F8 FD F0 01 F8 FE F0 87 B8 FF 30 48 B8	2715
0200:	FE 30 E8 B5 76 A0 E9 B5 77 BE F8 B6 77 BC E5 B6	2864
0210:	77 38 E9 B7 7E B8 D5 98 6C 5E D0 B7 71 C0 C0 06	2576
0220:	68 40 40 B4 60 40 40 D6 30 40 40 B5 80 20 26 B4	1585
0230:	BE 30 01 D7 FE 50 04 FC FE 50 08 F9 F8 70 05 FA	2253
0240:	F8 D0 00 44 17 3E E8 B3 90 00 E8 B4 95 20 E8 B4	2169
0250:	97 5F E8 B5 87 5F E8 B7 7E DF E8 B8 76 5D E0 B9	2689
0260:	5B 5B E0 D8 6C 39 C0 D6 71 5D 40 D6 70 7F 15 B6	2119
0270:	C0 70 56 D7 DF B0 12 DA FE D0 7B B5 B4 F0 40 B4	2670
0280:	9A 70 40 B4 98 50 00 48 00 37 40 B5 78 00 D0 AA	1612
0290:	5B C0 C0 AC 5C C1 C0 D0 65 C0 E9 B2 7E A1 E8 B2	2733
02A0:	86 A1 F8 B2 96 C0 E8 B3 97 5E F9 B3 96 DF F9 B4	2949
02B0:	96 DF E8 B5 8E 5E E8 B6 7D DF D4 B7 7C DF B4 AB	2829
02C0:	73 21 80 AC 62 E0 C5 AE 62 E0 00 00 38 00 50	1602
02D0:	1F 41 00 B6 90 10 11 D7 D6 B0 45 D7 F7 B0 0A B7	1960
02E0:	FF 50 2A B5 C6 30 A9 B5 8D A0 E8 B5 87 40 F8 B5	2496
02F0:	97 5E F8 B5 97 5E E8 B5 96 DE E8 B5 96 3E E8 B4	2741
0300:	7D 3D 90 AC 63 BF 80 8D 5C 5F C0 AD 64 40 C0 AD	2142
0310:	6B A0 C0 AD 6A A0 C4 AD 69 A0 C0 96 70 59 00 7C	2199
0320:	1A 38 10 D8 58 00 E2 B8 65 40 E8 B8 6E C0 E8 B6	2109
0330:	6F C0 E8 B6 6F 20 E1 D4 5D C0 E0 D4 65 C0 E4 D3	2750
0340:	5D C0 E8 B3 76 A1 E8 B3 87 A4 E8 B3 8F C1 E8 B3	2843
0350:	8F 5F E8 B3 86 BF E4 B2 7D 3F 94 B5 6C 5E A5 B5	2445
0360:	6C BE E4 B1 6D 20 E8 B1 7E 3F E8 B1 86 3F E8 B1	2457
0370:	8D BE EC B2 85 5E E8 B1 7D 3F C0 8E 63 8F C0 8E	2527
0380:	5B 5F C0 B2 5B 60 D6 B1 6C 2D D4 B3 7C 40 E4 B3	2260
0390:	7B 8D 80 B5 61 5B A9 B4 70 3E 00 5C 1A 40 00 F9	1763
03A0:	F8 10 01 99 FF 30 86 98 FE B0 81 B7 FE 30 55 B2	2314
03B0:	B4 D0 50 B1 BB B0 E0 8B 73 A0 EA 8D 7E 40 E8 91	2588
03C0:	97 40 E9 B2 9F FE E8 AF 8E FE E8 AC B5 7F E4 AB	2905
03D0:	7B FF E8 AB 72 C0 E5 AC 6A DD E5 AC 6A DD 40 B6	2789
03E0:	75 20 04 B3 C5 50 24 B1 B5 50 26 B1 93 B0 A4 B2	1963
03F0:	7B 5F 60 B3 81 C0 00 48 31 3F 05 B9 E8 10 04 B7	1623

0400:	FC	B0	04	D9	B0	30	05	B8	A5	B0	05	B8	AE	30	05	B8	1936
0410:	AE	30	09	B7	A5	B0	5A	B7	B5	30	E9	B3	77	40	FD	B3	2706
0420:	B7	5F	E9	B3	8F	7F	F9	B3	87	3F	E9	B3	B6	DE	E5	B3	2714
0430:	7E	20	B0	B3	7A	5E	D1	B5	79	A1	15	B7	B0	B0	00	38	1978
0440:	2B	3E	10	BA	F8	10	05	B5	FE	70	B4	B5	FD	D0	B5	B3	2206
0450:	E5	30	90	AB	BC	A0	90	AA	B0	C0	90	AA	B0	C0	90	A9	2499
0460:	B0	FF	AA	A9	B0	DD	F9	A9	B0	7F	E9	AA	8C	DF	F8	AA	2951
0470:	B4	40	B5	AA	7A	C0	00	48	3A	3A	25	B3	F8	10	12	B4	1727
0480:	FB	50	16	B3	FB	D0	05	B4	FC	50	05	B5	FC	D0	1A	B5	2361
0490:	FE	50	1A	B5	FE	30	1E	B5	CC	B0	F9	94	76	20	F9	94	2346
04A0:	76	20	E9	94	76	C0	E1	B6	6D	A0	16	B6	FE	D0	06	B9	2374
04B0:	FF	50	05	B5	FD	F0	05	B8	FD	F0	00	B4	D0	70	00	44	2295
04C0:	2D	2A	01	D8	F8	10	01	D8	FE	10	11	D9	FE	B0	46	D9	2006
04D0:	FF	30	45	D8	FF	30	2A	B7	F7	30	AE	B7	7D	B0	E8	F8	2485
04E0:	65	C0	E4	B9	64	F0	E4	B9	64	A0	E1	BA	74	EF	E0	BA	2831
04F0:	74	DF	E4	BA	6C	A0	E0	DA	6C	C0	B0	B9	5A	DE	C0	98	2732
0500:	62	DF	B0	76	62	C0	B0	95	62	C0	B0	D5	63	21	EA	B4	2311
0510:	75	20	D4	B4	7B	A0	A0	D4	7A	C0	A4	B4	91	43	60	B4	2342
0520:	90	25	00	44	30	41	50	B6	A0	10	05	B6	BE	90	01	B5	1503
0530:	D7	B0	29	B4	C6	B0	B0	96	B5	D0	24	8F	BC	B0	A4	8F	2503
0540:	B3	A0	F9	8F	7E	20	E9	91	87	3F	E8	93	8F	3E	E8	94	2381
0550:	8F	3E	E8	B5	87	3E	D8	B5	6E	BE	D0	AF	54	5E	C0	B1	2442
0560:	53	FF	50	D7	68	E0	00	40	26	32	50	D7	68	00	16	D8	1753
0570:	ED	D0	8A	B7	FE	50	06	B6	B6	30	9E	B6	9C	B0	C4	B5	2567
0580:	6B	BE	E8	B7	6D	DE	E8	B8	6D	DE	D4	B9	6B	DF	C4	99	2866
0590:	6B	C0	D9	B8	62	C0	94	B8	7B	50	B4	B6	92	50	55	B6	2332
05A0:	B0	D0	55	B6	B0	30	00	60	1F	36	14	B3	60	00	64	B2	1533
05B0:	90	B0	64	B2	AD	B0	64	B2	95	D0	91	D0	B5	42	A4	CF	2518
05C0:	9E	A2	A4	D0	9F	3F	B1	91	75	DE	B0	B1	5D	5F	C0	B0	2388
05D0:	5D	5F	C0	AD	5B	5F	D0	B5	52	A0	14	B6	87	30	14	B7	1958
05E0:	6F	30	E5	92	76	41	E5	93	7E	E0	E5	93	76	22	E8	B4	2383
05F0:	5E	20	C0	B5	6A	5E	05	B7	9D	B0	14	D8	E6	30	66	B4	2019
0600:	7C	50	B1	B4	62	57	00	B0	00	3E	01	B7	E0	10	01	B7	1544
0610:	DE	30	42	B7	A6	B0	16	B6	7E	B0	65	B5	96	30	A0	B4	2187
0620:	B0	A0	E0	B4	8F	3F	E1	B3	8F	C0	E2	92	5D	40	E2	D3	2616
0630:	55	21	E5	B4	7F	40	E4	B4	76	A1	D0	B3	64	DE	94	B5	2443
0640:	6C	BF	15	B5	6D	B0	2A	B4	8C	F0	EA	B2	8E	DE	F8	B1	2589
0650:	96	C0	E8	B1	96	C0	E8	B1	8E	7F	E4	B2	7E	7F	C4	B0	2802
0660:	5C	C0	C4	B0	5C	40	C0	AE	5B	C0	D0	AC	5A	A1	B0	D1	2429
0670:	64	A1	D0	D3	7C	22	C0	B0	73	22	B0	AF	72	A1	B6	B3	2294
0680:	71	A0	17	B4	60	60	00	64	00	35	17	B4	60	60	40	B0	1421
0690:	7A	21	D0	B8	53	20	C4	AF	5B	C0	E9	B2	8E	C0	E5	B1	2422
06A0:	96	C0	E6	AF	8E	41	91	8E	5D	40	D5	8F	54	E0	C0	91	2399
06B0:	5C	A0	D0	8F	53	DF	C0	8E	53	21	C0	AE	52	A1	A4	B1	2309
06C0:	7E	21	E5	B0	8E	A1	E5	B1	95	FF	E5	B1	95	A2	E5	B1	2800
06D0:	8C	C0	E5	B0	8C	20	D0	B0	83	A1	94	B2	73	3F	95	B2	2416
06E0:	8A	5E	95	B2	89	40	95	B2	88	40	00	50	2F	36	10	97	1635
06F0:	D0	10	38	97	AF	30	75	B7	E6	D0	26	B6	FE	30	66	B3	2195
0700:	B5	20	E6	92	8E	A9	E5	92	96	C0	FE	92	7E	7F	FA	92	2618
0710:	6E	60	EA	92	66	60	EE	93	65	E0	D0	B3	62	5D	11	B6	2271
0720:	CE	B0	15	D6	D7	90	06	B4	E6	30	BA	B4	5C	B0	E4	B4	2482
0730:	5B	5D	00	B4	59	60	00	00	00	70	00	40	1A	36	00	B7	988
0740:	60	00	90	B2	A3	40	A8	B3	A6	40	E8	B3	A6	FF	A8	B3	2401
0750:	9E	5F	EC	B2	96	40	B1	F1	64	C0	94	AD	5C	41	D4	AC	2341
0760:	54	40	C0	F0	4B	DF	B5	D8	54	20	45	BA	9E	30	46	DA	2028
0770:	76	B0	ED	B2	64	D8	40	D7	61	DD	00	6C	00	40	01	D8	2014
0780:	F8	10	05	DB	FC	30	06	DB	FD	50	05	FB	FD	F0	01	FB	2347
0790:	FE	F0	87	BB	FF	30	E8	B3	8F	C1	E8	B3	8F	5F	E8	B3	2926
07A0:	86	8F	E4	B2	7D	3F	94	B5	6C	5E	A5	B5	6C	BE	E4	B1	2499
07B0:	6D	20	E8	B1	7E	3F	E8	B1	86	3F	E8	B1	8D	8E	EC	B2	2499
07C0:	85	5E	E8	B1	7D	3F	C0	8E	63	8F	C0	8E	5B	5F	C0	B2	2338
07D0:	5B	60	D6	B1	6C	20	D4	B3	7C	40	E4	B3	7B	B0	B0	B5	2325
07E0:	61	5B	A9	B4	70	3E	50	D8	8A	D0	04	B5	ED	B0	15	B4	2152
07F0:	DE	90	42	B8	FC	B0	A5	B4	93	50	00	00	00	00	00	00	1616

— VOC'HORLOGE. Ce bloc de vocabulaire contient tous les éléments nécessaires à la réalisation d'une horloge parlante de qualité. Un exemple d'utilisation, facilement transposable à une autre machine, est donné pour le SHARP MZ-700. Ce bloc occupe 2 Ko de mémoire.

No	Ad.Deb	Ad.Fin	Texte
00	0048	007F	ZERO
01	0770	0797	UNE
02	0080	00B3	DEUX
03	00B4	00EF	TROIS
04	04F8	0543	QUATRE
05	00F0	0147	CINQ
06	0560	058F	SI(X)
07	0148	0197	SEPT
08	0198	01DF	HUIT
09	01E0	0217	NEUF
10	0590	05B3	DI(X)
11	0218	0267	ONZE
12	0268	02B3	DOUZE
13	02B4	030B	TREIZE
14	030C	0373	QUATORZE
15	0374	03B7	QUINZE
16	03B8	0413	SEIZE
17	05B4	05E3	VIN(GT)
18	05E4	0627	TREN(TE)
19	0414	046B	QUARAN(TE)
20	046C	04D7	CINQUAN(TE)
21	0628	066B	HEURE
22	066C	06C3	MINUTE
23	06C4	071F	SECONDE
24	04D8	04F7	T'
25	0720	073B	TE
26	073C	0767	TET
27	0544	0553	S'
28	0554	055F	Z'
29	0768	076F	SILENCE 1
30	07EC	07F7	SILENCE 2
31	0798	07EB	BONJOUR

VOC 'HORLOGE'

0000:	00 48 07 70 00 80 00 F4 04 F8 00 F0 05 60 01 48	1165
0010:	01 98 01 E0 05 90 02 18 02 68 02 E4 03 0C 03 74	975
0020:	03 88 05 B4 05 E4 04 14 04 6C 06 28 06 6C 06 L4	1103
0030:	04 08 07 20 07 3C 05 44 05 54 07 68 07 FC 07 98	1001
0040:	FF FF FF FF FF FF 00 00 38 00 30 65 D4 58 00	2290
0050:	A5 D4 54 A2 56 D5 56 A2 E6 B5 5E 40 ED B6 6F C2	2463
0060:	EA B6 7F C0 AA B0 87 5F 96 AB 8D DD EB A9 7D E1	2748
0070:	DA A8 75 60 95 AB 7D 42 40 93 8A DD 80 94 88 5F	2184
0080:	00 34 00 30 85 04 28 10 06 B5 35 10 07 D6 3E 90	1184
0090:	A9 B6 56 20 EF B6 5F F0 EF B4 6F C6 D6 B3 6F 43	2469
00A0:	EA B3 66 DE FA B4 66 DA F6 B4 66 3E EA B4 64 C5	2788
00E0:	40 B6 80 7E 00 3C 00 2E 14 D3 88 10 14 D3 8C 90	1504
00C0:	24 D0 06 B0 55 D2 6D 30 C4 D2 62 A0 CA A7 7B 5F	2129
00D0:	D9 AB 7C 20 DA AC 96 40 69 AD B7 C2 7D AE AF C1	2467
00E0:	88 AF B7 5E 80 AF A6 5B 76 AF 93 60 40 D4 98 46	2291
00F0:	00 58 00 36 19 F7 80 10 19 F7 84 10 19 F7 85 80	1559
0100:	19 F7 86 50 19 F7 86 D0 19 F7 86 B0 A4 D4 6E A0	2328
0110:	78 D4 9F C0 74 D3 8F FF 71 B3 77 5E B0 D4 7E DB	2646
0120:	91 B5 75 5E 95 B6 7C C3 D0 D7 53 41 10 B4 61 61	2148
0130:	00 B4 68 70 04 B4 70 30 04 B5 7D 30 04 B7 86 30	1467
0140:	18 B7 8C 50 54 B7 90 30 00 50 00 30 1A F7 80 00	1415
0150:	1A F7 83 10 1A F7 85 50 1A F7 85 D0 1A F7 85 B0	2102
0160:	B8 B5 7E 26 B8 B5 8F A5 F8 B6 97 C2 F4 B6 87 DF	2761
0170:	FC B4 7F 39 E4 B6 74 A0 E8 B7 73 A0 D1 B7 82 BF	2707
0180:	E4 B5 59 A0 90 D6 80 DF 40 D7 78 60 01 B5 84 70	2288
0190:	A4 B6 9C 70 E4 B6 90 5E 00 48 00 30 95 96 58 00	1769
01A0:	D9 97 4B 5E E9 B6 43 DA D5 95 44 42 F5 95 4D C4	2400
01E0:	DA B7 4F 46 D6 D9 4F C4 D0 DA 4D DE C1 D9 33 58	2530
01C0:	40 B6 40 FE 01 B6 78 E2 00 B6 88 42 00 B6 90 30	1851
01D0:	14 B5 98 30 20 B6 A4 B0 74 B7 8C 50 40 D8 C8 40	2069
01E0:	00 38 00 30 84 B4 40 00 88 B4 4E 20 B9 B4 7F A2	1608
01F0:	BE B3 97 C5 FE B3 8F DF FA B2 76 DE 65 B4 65 5F	2761
0200:	65 B5 75 80 29 B5 AD D0 19 B4 D4 D0 28 B6 EC F0	2501
0210:	25 B6 EC 70 14 B4 D8 50 00 50 00 35 00 00 00 20	1228
0220:	41 BA ED A0 12 CA DF 24 42 CA B7 43 42 CC 87 41	2067
0230:	42 CC 8F 7F 42 CA AE DE 42 AA 9E D0 82 6A 8D 7D	2321
0240:	81 AA 64 DE 81 AB 94 7E 58 CD 7C 40 D6 D1 6C 40	2271
0250:	E9 F5 5C 40 E1 F5 5C 40 E2 D3 6C 40 B2 B3 85 40	2423
0260:	61 D4 63 C0 01 B6 90 46 00 4C 00 27 00 00 00 20	1144
0270:	D4 B5 4C 80 54 B6 4E 0C 66 B6 57 2A 52 B6 5F A8	1893
0280:	53 B1 57 44 52 AE 57 41 92 A0 66 DE 92 AC 5D 7C	2001
0290:	D2 AD 53 7C D2 AE 52 7E 94 AE 51 DF D5 B4 4B 40	2340
02A0:	A6 B6 54 40 AA D6 5C C0 E6 D5 64 E0 B1 D5 5C C0	2605
02B0:	11 B3 18 65 00 58 00 30 00 00 00 20 10 D2 9E 10	889
02C0:	10 D2 97 90 61 B3 8F 20 21 B4 96 BF AA 94 97 40	2059
02D0:	72 B4 AF C7 76 B5 9F C3 B6 B7 97 FC B7 B7 8F DC	2818
02E0:	F6 B7 87 DD B7 B7 7E FF E6 B7 60 61 D9 96 5C C3	2805
02F0:	D2 B7 54 5F A1 B6 54 40 51 D7 4C C0 61 D6 6D 7F	2174
0300:	E2 B4 85 41 70 D5 75 5F 30 D5 60 46 00 68 00 3E	1731
0310:	00 00 00 20 05 97 97 10 05 97 97 B0 B8 96 97 A0	1486
0320:	AB 94 97 A3 85 94 7E 40 EA B4 88 40 94 B4 79 C1	2363
0330:	90 B4 79 A1 80 B4 81 90 11 D7 87 B0 A1 D4 87 84	2370
0340:	A6 D2 9F DF 66 B1 9F DE A6 B1 9F 5E A6 AF 96 7D	2630
0350:	B2 AD 94 7E 96 AC 82 FF 96 AE 8C 40 92 AF 7C C0	2497
0360:	55 D5 64 E0 84 D6 74 40 E6 B3 84 40 91 94 73 40	2225
0370:	50 B4 58 47 00 44 00 43 00 00 00 20 55 DA 77 B0	1184
0380:	F6 B7 7F C0 67 B6 8F C0 62 B6 6F DF 22 B6 6F FC	2561
0390:	22 B5 66 FD 36 B5 5E 5F 75 B5 6D 7F B4 B5 6C C0	2189
03A0:	F1 95 68 C0 E2 D5 53 E0 E1 D5 58 E0 A4 B5 84 C0	2857
03B0:	60 D5 7C 40 50 D5 68 48 00 5C 00 3D 00 00 00 20	1151
03C0:	1F F6 7C D0 18 F6 7D F0 B5 B5 8F C0 F6 B6 9F C3	2726
03D0:	B6 B6 97 DE 66 B7 97 DD F6 B7 97 DE F6 B7 8F 5D	2861
03E0:	F5 97 87 5D F2 B7 B6 5E E6 B7 75 5E F1 B6 6C C0	2624
03F0:	A1 B6 5C 40 EB D6 4C C0 55 D6 54 C0 61 D7 5C 40	2256

0400:	54 F7 5C 22 A5 D6 64 20 E4 B5 5C 70 A4 D6 6B C0	2178
0410:	34 D5 60 46 00 58 00 30 05 96 88 10 05 96 8C 10	1185
0420:	09 97 8D 90 75 97 86 A0 B8 96 97 2E 7B 95 A7 A0	2137
0430:	66 93 8F C0 24 93 8F 5F 10 93 9E 3C 25 92 95 3E	1876
0440:	21 91 8D 06 32 E0 66 C4 22 AF 57 3F 33 AF 56 DE	1742
0450:	33 AF 5E 5C 23 8E 5D DD 23 8D 5D 5C 32 8D 64 DF	1777
0460:	21 AD 64 40 75 AF 6E 41 40 E5 60 63 00 6C 00 40	1445
0470:	1E F7 80 10 1E F7 83 90 1E F7 83 80 1E F6 8D 70	2086
0480:	B6 E4 9F 20 B6 E4 9F 21 66 E5 7F DF 12 B6 57 C1	2220
0490:	62 E7 57 3F 96 94 54 DF 92 98 53 A0 D1 96 5A EF	2217
04A0:	D1 97 48 DF 91 96 6E A0 A1 95 85 1E A1 95 8D 80	2269
04B0:	AB 94 8E 8F 6E 92 9F 20 23 E1 9F 3F 23 E0 86 DF	2098
04C0:	22 80 6E 5C 32 AF 65 DE 22 8F 65 5C 71 8E 5C 5F	1772
04D0:	61 8E 5E 5F 51 AF 48 64 00 70 00 30 01 B6 78 10	1252
04E0:	01 B6 7B 90 14 D7 85 80 B4 D6 85 30 E5 85 7C A0	2263
04F0:	A5 B5 8C B0 60 D6 90 70 00 4C 30 3D 08 B6 C8 10	1819
0500:	08 B7 D7 30 08 E8 8F 30 AA B6 7F A0 F9 B5 97 B6	2244
0510:	B4 B4 B7 DE 74 B3 8F DE B8 B2 A7 5E 90 B3 95 5C	2660
0520:	51 B3 8A C7 14 B4 8A E0 A5 B2 C4 D0 A0 D1 FC 50	2607
0530:	40 D2 FC 50 FA 95 84 D8 F1 94 84 40 50 95 8B 42	2372
0540:	90 D6 90 45 00 10 34 34 1A F7 55 50 1A F7 55 70	1599
0550:	1A F7 54 70 00 0C 34 34 22 B5 64 60 75 85 6B 60	1497
0560:	00 30 2E 34 16 F7 80 10 16 F7 83 90 16 F7 86 30	1551
0570:	16 F7 86 F0 15 F7 86 D0 58 D8 66 A0 D8 B9 5F A5	2480
0580:	D4 D9 47 C2 D4 DA 47 40 C4 D8 46 5B D9 D8 45 B8	2518
0590:	00 24 34 30 90 D8 40 00 90 D8 46 00 B6 D8 5E 26	1520
05A0:	F6 D9 57 C6 E5 D9 57 43 F4 DA 4E C0 F5 D9 4E 5A	2710
05B0:	E5 B9 4D DC 00 30 30 01 B5 68 00 55 B3 74 C1	1714
05C0:	51 D5 6D 41 14 D5 65 41 25 D6 6D C1 66 B3 8E C1	2036
05D0:	76 B3 A7 C1 62 B4 A7 C0 61 B4 AF FE 61 94 AE 5E	2513
05E0:	90 D5 78 60 00 44 30 30 11 B7 90 10 11 B7 95 90	1590
05F0:	52 96 96 10 11 96 96 10 21 B3 8D A0 56 93 8D A0	1778
0600:	27 90 8E C3 13 8D 7F 45 23 8E 7F 5F 23 8D 76 DC	1789
0610:	23 8C 75 DC 12 8B 75 DD 53 8C 74 5E 56 AB 72 D0	2032
0620:	51 AD 41 C2 40 AF 4B 47 00 44 2F 2F 66 B5 95 01	1490
0630:	77 B4 97 20 77 B4 97 C0 B7 B4 97 5F 77 B4 97 40	2247
0640:	F7 B5 96 DF B7 B5 8E C0 F6 B5 8E 5F FA 95 85 5E	2789
0650:	F6 95 7C DF FA 96 7C BE FA 95 7C FF FA 95 84 C0	2957
0660:	F1 94 84 22 50 95 88 40 90 D6 90 48 00 58 20 2F	1728
0670:	DC 95 3B A0 DD 95 34 27 A2 B8 4F 21 B6 D8 57 A0	2152
0680:	E6 D9 4F C1 E2 D9 4F 20 D7 97 45 A0 D9 96 45 40	2368
0690:	D9 B7 45 A0 A0 B8 47 3D A0 B8 57 5C B1 B8 5F 5B	2175
06A0:	F1 8B 54 5D A4 F8 50 E0 00 B6 60 63 00 B6 60 43	2040
06B0:	02 D6 6A B0 52 D6 75 30 FA B5 85 DF F6 B5 95 5D	2415
06C0:	D0 B5 90 7E 00 5C 1F 39 1E F7 AC 30 1A F7 8D 70	1862
06D0:	1A B5 7E 41 FE B4 6F 21 FE B4 6F 40 AE B5 6E A1	2211
06E0:	D7 95 48 C0 63 90 5D C0 47 91 76 3F 47 B0 A6 DF	2189
06F0:	13 AE AD DE 13 A4 AD 5F 43 A9 A4 DE 53 A9 93 DF	2280
0700:	53 AA 8B 5F C4 B2 40 FF 80 B5 4A 5F 62 B4 74 3E	2114
0710:	F2 B4 84 8F F2 B4 8D 40 E2 B5 94 5F 90 B5 78 5F	2562
0720:	00 1C 28 30 00 B6 60 40 02 D6 6A B0 52 D6 75 30	1417
0730:	FA B5 85 DF F6 B5 95 5F D0 B5 90 7F 00 2C 2D 30	2255
0740:	01 B6 78 10 01 B6 7D 90 14 D7 B6 B0 D6 B8 6E 20	1856
0750:	B3 B8 66 A3 B3 D8 5F 5F F3 D9 6E DE F3 D9 6D A0	2734
0760:	F3 D9 6B 40 F3 D9 6B 40 00 08 32 32 00 B4 6B 70	1763
0770:	00 2B 30 32 A2 B7 57 E1 F6 B8 57 E0 D0 B7 4D 41	2069
0780:	C0 B6 5C C0 94 D6 5C DF 94 D5 54 40 E4 D6 5D DF	2602
0790:	72 B5 4C DF 71 D5 4B 5F 00 54 32 30 00 B5 70 00	1562
07A0:	40 B4 45 A0 E9 AD 2B A2 C0 AF 2C 23 E5 AA 7D C0	2246
07B0:	E6 AC 8E 5F B9 AE 7E 40 90 92 5D DF 50 B4 55 50	2219
07C0:	40 B5 55 30 E8 B3 5D BF EA 91 5D C1 AA AF 5E E3	2404
07D0:	E5 AF 6E 62 F1 AE 65 C3 A1 AD 5D 40 55 AF 53 C0	2349
07E0:	55 AF 53 5F 00 D5 51 DA 00 D5 50 5A 00 0C 32 32	1445
07F0:	00 B4 6B 70 00 B4 6B 70 FF FF FF FF FF FF FF FF	2832

— PAROLE ET MICROS. Ce bloc est le plus volumineux, puisqu'il occupe 6 Ko. Il se compose des phrases utilisées dans la cassette de présentation fournie avec l'interface vocale "PAROLE" de CEDIC-NATHAN pour les ordinateurs THOMSON (MO5, TO7, TO9). On pourra naturellement réutiliser tout ou partie de ce vocabulaire sur d'autres machines, décrites ou non dans ce livre.

No	Ad.Deb	Ad.Fin	Texte
01	0C78	0D2F	Bonjour!
02	0030	0223	C'est moi votre ordinateur Prefere.
03	0224	0443	CEDIC-NATHAN m'a appris a Parler.
04	0444	0513	Je sais compter...
05	0C34	0C77	un...
06	0EF0	0C33	deux...
07	1134	118F	trois...
08	1190	1207	cinq.
09	0514	059F	Euh...pardon!
10	1208	127B	quatre...
11	05A0	065F	Je suis poli.
12	0660	0707	Bonjour!
13	1768	1807	Au revoir!
14	0708	079F	Bienvenue!
15	07A0	081F	Merci!
16	0908	0BEF	Je peux prendre une voix feminine.....
17	0D30	0F43	Votre correspondant est absent
18	0F44	1133	veuillez rappeler ulterieurement.
19	127C	1767	Et bien d'autres phrases .....
20	0820	0907	Programmez moi bien!

PAROLE ET MICROS

0060:	0C 78 00 30 02 24 04 44 0C 34 0F F0 11 34 11 90	035
0010:	05 14 12 08 05 A0 06 60 17 68 07 08 07 A0 09 00	644
0020:	0D 30 0F 44 12 7L 08 20 FF FF FF FF FF FF FF 00	2111
0030:	01 F4 29 4A 02 D9 F8 10 02 F8 F4 E0 02 F8 F5 30	2056
0040:	02 F8 F5 30 02 FF F5 30 02 F8 F5 30 02 F8 F6 30	2173
0050:	16 B7 C6 B0 A9 B7 77 A0 A4 D7 7F BE BB E7 71 ED	2564
0060:	EB E8 77 20 56 B6 52 3C CA 94 44 3F D8 B5 44 A0	2090
0070:	CC B6 44 A0 8L B7 44 A0 8C B8 44 BF 0C B4 55 3E	2085
0080:	4L B0 54 B0 5C B6 4C BF 69 B0 55 3F 25 B0 3L BF	1842
0090:	25 BF 9E B0 61 B0 BE BE 65 B1 C6 BE BA B1 LL 3E	2465
00A0:	75 B2 E6 3D E4 B2 ED 3L 55 B2 A4 B0 69 B2 9C E0	2310
00E0:	AF B1 7C 30 2L B1 3E B0 26 B1 23 B0 12 B1 03 B0	1773
00C0:	15 B1 0B 30 10 B1 54 30 74 B0 75 2D EE A1 B6 22	1574
00D0:	EA AF B6 20 BE B0 B6 A0 B8 B1 7E 3F 69 B1 54 A0	226L
00L0:	45 B0 52 30 91 L2 6A 30 44 D7 E5 B0 27 B1 CD B0	2137
00F0:	26 B1 ED 30 B5 B2 A4 30 6A AF 96 AF 6A AF 96 BA	2198
0100:	7L AC 96 3D 6A AL B0 B0 5B AE 9L 3D 6A AF 96 21	2069
0110:	6A AF BE 21 AA AF 43 3C D0 B1 3A A6 95 B5 44 A3	2090
0120:	AA B7 4D A2 E6 D8 4F 79 D6 D9 4E 21 D5 B8 3D FF	2371
0130:	C8 B5 3C A0 C8 B5 3C BE E9 B5 45 A0 A9 B6 67 A0	2465
0140:	6A B6 B7 A0 6A B6 9F A0 69 B6 97 20 6C B6 B6 BF	2275
0150:	AC B5 63 B0 A4 B5 72 E0 94 D5 71 B0 A4 D5 61 30	2435
0160:	B4 B5 58 30 50 F7 B5 30 55 D5 B6 B0 68 B5 77 AB	2188
0170:	6B B5 7F BF 2B B4 7F BF 1A B3 B7 B0 1E B1 7E B0	2198
0180:	56 B0 75 BC 22 CE 45 30 11 D2 A4 30 01 B3 62 30	1689
0190:	44 B3 50 30 44 B3 50 30 44 B3 50 30 44 B3 50 30	1500
01A0:	14 91 02 B0 14 B0 06 30 11 D1 A5 20 32 D2 05 20	1313
01B0:	1A B5 6E AC 6A B7 77 A2 AA B8 77 A1 D4 B8 5E 3D	2244
01C0:	A9 B7 24 BE 16 B7 05 B0 16 B7 15 B0 16 B7 25 B0	1784
01D0:	16 B7 2D B0 16 B7 3D B0 16 B7 4D B0 95 B8 5D A2	1914
01E0:	AB B8 67 21 AB B9 67 3F EE B8 6F 3D EB B8 6F 3E	2196
01F0:	9B B7 6F 3D 49 D7 66 ED 59 D6 6E B5 59 D6 6F 23	2132
0200:	6B B7 77 A4 DB B8 77 3F EB B8 6F 3D EB B9 6E BF	2470
0210:	EB B9 66 BC EB B9 66 3D CC B9 75 20 54 DA 42 A0	2359
0220:	20 B8 40 1F 02 20 26 43 02 F8 E0 10 02 F8 D4 E0	1581
0230:	02 F8 BC B0 02 F7 AC B0 02 F7 95 30 02 F7 B4 B0	2214
0240:	02 F7 6D B0 1D B7 5C A0 A9 B7 5F 22 A9 B8 5F A0	2087
0250:	EA B8 57 20 EA B8 4C 20 D4 D7 23 B0 D4 D8 1B 30	2206
0260:	B1 B4 23 B0 A4 F9 3D 30 02 D9 3E 2F E2 DA 3E AF	2275
0270:	E2 DA 36 24 B5 B5 35 3F B1 BA 3A 30 B2 B6 31 30	1794
0280:	45 B4 39 B0 04 B4 36 30 04 B4 35 30 B8 B6 32 B0	1597
0290:	B8 B7 35 31 98 B6 3D B1 9C B6 3D BC AA B5 7F 3D	2119
02A0:	A6 B4 C7 BF A5 B4 BF BE A5 B4 AF 3E 68 B5 BE EE	2461
02E0:	AB B5 6C 3D A4 B6 62 30 61 B5 61 30 95 B5 60 30	1907
02C0:	95 B5 60 30 95 B5 60 30 95 B5 60 30 95 B5 60 30	1896
02D0:	95 B5 60 30 56 B6 E5 E0 A5 F5 76 30 AB B4 7F 22	2235
02E0:	66 B3 B7 20 75 B1 77 20 26 B0 77 20 61 AE B7 22	1698
02F0:	61 AC 97 23 22 AC 97 24 22 6B 96 23 22 B8 95 A3	1659
0300:	22 BA 95 24 23 AB BC A3 12 AA BB A3 22 AA BA A1	1859
0310:	30 AA BA 30 10 AA B9 30 01 B9 00 30 00 AC 00 30	1181
0320:	00 AC 00 30 00 AC 00 30 00 AC 00 30 00 AC 00 30	880
0330:	00 AC 00 30 00 AC 00 30 00 AC 00 30 00 AC 00 30	880
0340:	00 AC 00 30 00 AC 00 30 6C B5 33 30 08 B3 3B B1	1459
0350:	ED B4 3D 3F 9D B4 3D A0 AE B3 46 20 66 B3 BF 3F	2041
0360:	65 B4 AF A0 A5 B4 C7 A0 A5 B4 C7 A0 A9 B4 CF A0	2740
0370:	A9 B4 CF A0 A9 B4 C7 A0 AA B4 BF BF A9 B4 A7 20	2704
0380:	6A B3 7D BF 90 B3 7B 30 A0 B4 6A 30 A0 B4 B9 30	2114
0390:	11 B2 B0 30 10 B1 A0 30 29 B2 C2 B0 12 B6 D6 30	1823
03A0:	12 B6 9E 30 AB BA 5E BF A7 DA 56 BF E7 D9 4E BE	2426
03B0:	EB D9 4F 3F FE D9 67 3F A6 B8 B7 B0 69 B7 9F BF	2540
03C0:	69 B5 AF BE 69 B3 A6 3E 50 B4 6B B0 94 B4 B2 30	2212
03D0:	91 B5 91 B0 90 B5 90 30 90 B5 90 30 59 B1 96 30	2145
03E0:	AC B2 A6 A2 B9 B2 B7 3F F9 B2 C7 3E A4 B2 CE A0	2683
03F0:	A5 B7 C6 B1 A9 B1 BE 3E 69 B0 9D 3C 65 B0 94 3C	2300

0400:	59 E2 95 21 58 B7 C6 21 09 E0 05 A7 0A E4 4D A1	1726
0410:	48 B8 55 22 A4 B7 5D 21 A8 E9 65 3F A9 E9 65 20	1852
0420:	A9 E9 65 3F A5 BA 65 3F 94 BA 5C A0 B6 BA 54 3F	2086
0430:	E9 EA 5C 20 A4 B8 4C 30 71 B8 2B B0 04 B8 52 30	1858
0440:	45 E6 48 10 00 00 80 31 08 E7 18 00 99 E9 3C 20	1369
0450:	09 E8 36 20 1D D9 46 B0 1C D7 57 30 66 B6 67 A3	1699
0460:	66 E6 6F BF 56 B7 6E BD 11 F6 75 BC 01 F6 6L B0	2255
0470:	01 F7 6E 30 01 F8 6D 30 01 F8 6D 30 01 E9 6D 30	1561
0480:	2E B8 6E 30 D6 B8 6D A1 AA B9 66 3F AA E9 66 20	2065
0490:	A6 E9 55 B1 D6 B5 4D 3L B6 B9 2F BF 94 E8 41 A1	2275
04A0:	10 B7 49 30 10 B7 48 30 51 B3 EC 30 21 B1 D8 B0	1788
04B0:	18 B1 93 B0 18 B8 75 2C 48 A9 6D 22 48 AA 5C A1	1727
04C0:	45 AE 4D A2 49 AE 4D A1 49 AC 55 A1 46 AF 4D BE	1961
04D0:	CA B0 33 3E B0 B3 31 30 40 B5 40 70 40 B5 40 50	1765
04E0:	01 E7 E6 B0 05 B6 53 B0 AA B6 66 AF AB B7 67 2F	2169
04F0:	AE B7 6F 25 EE B8 77 25 EE B8 76 AB CA B9 6E 29	2320
0500:	CA B8 45 A6 D4 EA 44 A6 D8 EA 44 27 C4 BA 4A 30	2266
0510:	C4 B6 40 B0 00 BC 27 39 00 99 F8 10 16 B6 FC B0	1903
0520:	65 B6 6E 20 96 B5 6F 20 55 B6 7F BF 66 B6 7F 3D	1956
0530:	65 96 76 BA A6 94 6C BE 90 B5 42 B0 90 97 7A 30	2199
0540:	54 B5 61 30 54 B5 60 30 54 B5 60 30 11 B4 66 B0	1703
0550:	A5 B4 BE AF 69 B3 A7 33 64 B3 BF BE B4 B3 BF 3D	2435
0560:	60 B2 CE BF 74 D1 B6 3F 7A B1 AE 20 74 B0 A5 22	2237
0570:	B1 B0 95 A3 A5 B0 5C 3D D0 B1 32 30 05 B5 1D BA	2043
0580:	BC B3 7D A3 E9 B4 7E 20 69 B3 B6 3D 2A B1 95 A1	2186
0590:	28 AE 34 A1 43 97 C3 21 0C B7 02 A0 4C B3 20 00	1517
05A0:	00 C0 24 36 29 BA B8 10 29 BA B0 30 29 BA BE 30	1494
05B0:	04 B7 BF 30 29 B5 BF 21 5A B5 5F 22 56 B5 5F 22	1572
05C0:	56 B5 67 20 56 B5 56 BD 02 B7 4D 37 05 F8 4E 30	1640
05D0:	05 F8 4D B0 04 F8 4E 30 04 F8 4E 30 04 F8 4E B0	1768
05E0:	02 D8 4E B0 29 B8 4E 30 05 97 4D 30 11 B8 4B B0	1556
05F0:	AB B8 4D AA E7 B9 46 A0 E8 D9 3E BE EA B9 3E BD	2622
0600:	D5 B7 3C 3E C5 B8 22 30 41 B6 38 F0 00 B4 01 B0	1881
0610:	15 B2 6E B0 65 B0 7F 2F 76 B1 BF 2F 92 B2 96 AF	2070
0620:	A6 B3 BF 2F E6 B4 B6 A4 A4 B6 4D BA E9 B7 3D 20	2361
0630:	CD B8 35 37 CA B8 3D 31 E6 D9 4D B1 E7 DA 3D B3	2383
0640:	EC FA 35 37 DB FA 34 B9 E2 FA 2C 3E D2 DB 2B 3F	2414
0650:	E2 DB 2A BF C6 B7 32 A0 41 D9 29 B0 41 B9 2B 10	2074
0660:	00 AB 8A 24 B8 91 2A A0 55 B1 3C A0 44 96 3D 25	1623
0670:	14 BA 3D 21 19 AA 44 A1 19 B6 44 A1 18 B6 44 A1	1451
0680:	18 B6 4D 20 14 B8 4D 20 14 BF 4D 20 11 92 55 20	1084
0690:	15 B3 55 3E 10 B3 55 BF 10 B4 55 BC 09 B5 5E B0	1747
06A0:	05 B7 5F 30 19 B4 5D A7 65 B4 65 20 A9 92 5C A1	1778
06B0:	E6 B1 54 23 D2 B0 54 22 97 BF 54 A2 E3 AF 55 24	2093
06C0:	E6 BE 5C A4 A7 AE 5D 24 AB AE 5D 24 97 AF 55 26	2021
06D0:	EB AE 5D 25 DB AE 65 2B EB AE 64 A7 DE AE 5C A6	2406
06E0:	DA AE 54 23 DE AE 4C 27 EA AF 44 27 AA AF 4C A6	2125
06F0:	BE B0 4C A7 BA B0 55 22 B6 B0 5D 23 B9 AF 5C 20	2060
0700:	64 AF 5B 3F 59 D0 50 30 00 98 3F 32 C0 B7 20 10	1542
0710:	99 B6 25 20 D2 D9 3D 24 E4 F8 3D A4 E6 DB 46 23	2180
0720:	E7 D8 4E A6 A7 D8 5F 26 66 B7 6F A3 66 B7 7F A4	2342
0730:	65 B6 7F A5 6A B6 B7 25 6A B5 B7 25 79 B4 5E 20	1921
0740:	66 B3 3D BC 99 B4 2C B5 56 B5 1C B0 15 B5 04 30	1813
0750:	55 B5 1C 30 56 D5 23 B0 B5 B2 2A B0 B8 B8 43 B0	1944
0760:	B4 B4 44 B5 C4 B4 3D 3D C6 B4 3D BF DB B7 45 BE	2350
0770:	DB B8 45 A0 EB B8 46 20 EB B8 46 3F EB B8 46 20	2226
0780:	EB B8 45 A0 EB B8 46 21 EB B8 45 A1 EB B8 45 3F	2370
0790:	DB B8 44 20 C5 97 3B 20 C8 B8 31 30 D4 B8 30 10	1883
07A0:	00 B0 3B 2D 00 94 F8 10 1C B5 03 A0 9C B6 2C 20	1430
07B0:	98 B5 2D 21 A4 B7 26 A3 52 B8 47 A3 95 B8 47 A4	1937
07C0:	A5 B8 57 A7 99 B8 6F A7 99 B7 6F 29 79 B5 76 A0	2286
07D0:	22 B3 6D B0 00 D6 65 30 01 F6 66 30 01 F6 5D B0	1774
07E0:	01 F7 5D B0 01 F7 56 30 01 F7 56 30 01 F8 4E 30	1656
07F0:	31 F8 46 30 1E D9 44 B0 D2 D9 4E 20 E1 DA 4E 20	1996
0800:	E8 FA 4L 20 F8 FA 4D A0 12 BA 4C A0 B9 E9 5C 3C	2416
0810:	C4 B7 33 3B C4 B9 7A B9 CB B5 2A 30 B8 E6 1B 30	1955

0820:	00	E8	59	30	00	99	F8	10	15	B4	D6	30	16	B1	B4	20	1612
0830:	B6	AF	F4	A0	17	AA	04	24	57	AE	A6	A0	57	AB	A6	A1	2115
0840:	B7	AD	9E	20	B9	AF	9E	20	75	90	7D	A0	71	90	54	3D	2044
0850:	R1	91	39	3B	90	F2	4A	BD	20	59	EC	B0	71	B3	7D	30	2009
0860:	66	B2	7E	25	65	B2	A6	A0	66	B1	B7	25	B6	B1	BF	22	2131
0870:	A9	B1	BF	A0	7B	B0	B6	3F	98	F2	3B	A0	DC	93	34	20	2190
0880:	D8	94	34	21	D5	96	35	A2	EA	B7	46	A0	D6	B8	46	A0	2302
0890:	A6	B8	46	A0	AE	94	3E	20	9C	B2	35	3F	CC	B6	34	A3	2047
08A0:	CC	B6	3D	22	8C	B7	45	20	1D	AA	4E	21	6F	AE	67	20	1635
08B0:	3E	E0	7F	A1	7B	93	9F	A2	BB	B4	97	21	BA	B4	7E	A0	2317
08C0:	E5	B4	3E	BE	91	93	3A	A1	80	94	3A	22	91	B8	2D	B0	2087
08D0:	C5	DA	35	E0	98	F9	35	30	93	DA	34	AA	D8	F9	3D	A4	2423
08E0:	91	F9	46	AB	51	D9	57	27	40	B9	67	26	41	B4	57	25	1815
08F0:	55	E4	46	B0	91	B5	3E	3C	02	94	45	3C	C1	B2	43	BA	2003
0900:	D0	B6	39	B0	B0	B5	40	30	02	E8	25	3C	00	99	F8	10	1792
0910:	AB	B6	44	30	6D	B6	4D	B0	29	BA	8E	B0	15	B6	47	20	1861
0920:	64	B6	5F	23	54	B5	6F	24	61	B4	77	21	91	93	65	A0	1806
0930:	A1	B4	33	A0	90	D6	4A	30	90	B4	51	30	91	B5	50	B0	2067
0940:	15	B2	CD	B0	26	92	5D	3F	66	92	76	BF	A6	92	6E	3D	1960
0950:	66	91	66	3D	56	91	2C	3F	81	B3	49	B0	51	B2	71	30	1725
0960:	14	B5	AD	B0	29	B3	B4	B0	41	93	EB	B0	16	AC	7D	30	2116
0970:	27	AC	7E	A9	26	AC	76	A1	26	AC	96	A1	26	AD	9F	20	1918
0980:	26	AD	8F	3F	26	CF	B7	3F	26	B0	77	3F	27	B1	06	B0	1667
0990:	AD	B1	3C	3C	DE	B2	3B	3C	03	D3	05	22	5A	B3	6D	A1	1781
09A0:	5A	B2	66	3F	09	B2	4E	3C	55	B5	5E	21	55	B6	4E	A2	1658
09B0:	65	B7	56	A0	66	B6	56	BF	DA	B6	34	3D	CC	B5	34	3F	2104
09C0:	CC	B5	34	3F	C8	B5	34	3F	55	B5	2D	A0	15	94	05	BF	1832
09D0:	15	92	2D	3F	10	92	4C	BE	00	B0	04	30	14	B4	03	B0	1310
09E0:	55	B4	4B	B0	40	B0	7C	23	62	8E	6C	A0	6A	90	7E	21	1832
09F0:	15	92	86	BF	25	92	B6	BE	A5	93	AE	3F	51	74	9C	BE	2139
0A00:	15	B5	BC	30	15	B6	C5	30	05	B7	CD	30	05	B5	FD	B0	1942
0A10:	16	B6	FD	B0	2A	B7	05	30	AA	B7	66	2E	AB	B7	5E	BF	2051
0A20:	EA	B7	56	3E	96	B7	3C	BC	D9	94	33	A2	D9	94	33	A1	2301
0A30:	D9	93	33	A1	DA	B9	3E	21	C6	D9	3E	3F	C6	D9	36	3F	2146
0A40:	C2	D9	35	A0	C6	D7	35	BF	C6	D6	35	20	D5	B3	2B	A1	2374
0A50:	C4	B2	33	21	C4	B2	33	22	C4	B3	33	21	C1	B8	3C	A1	1974
0A60:	C1	B9	3D	A1	C6	D9	3D	A3	C2	BA	4E	23	B2	DA	56	25	2203
0A70:	91	DA	4E	A6	82	BA	46	27	85	B9	3D	A7	C1	B3	34	24	2038
0A80:	C0	B4	2B	A4	C4	B7	2A	A4	C8	B6	33	25	C8	B4	33	A2	2227
0A90:	C8	B4	33	A4	C8	B4	33	A4	D0	D6	3D	20	A8	D7	45	BE	2344
0AA0:	50	D7	56	3E	51	D6	65	BB	54	D5	64	B5	F0	D6	4A	35	2182
0AB0:	10	97	70	70	00	B6	68	70	10	B7	60	70	10	B7	58	70	1595
0AC0:	10	B7	50	70	10	B7	48	70	00	B6	49	D0	E6	B8	4D	31	1777
0AD0:	D7	B9	5E	B0	D7	B9	5F	3F	EB	B9	5E	BF	EA	B9	66	BF	2658
0AE0:	AA	B8	66	A0	AA	B8	76	A0	A9	B7	7E	3F	55	B6	75	BE	2363
0AF0:	56	B7	6E	20	66	B7	7E	BE	6A	B7	7F	3F	AA	B8	7F	3E	2034
0B00:	A6	B6	76	BF	96	B4	4B	BE	B5	95	42	A0	91	B7	41	B0	2329
0B10:	91	B7	59	30	90	D6	60	B0	15	B2	CC	30	66	B1	B4	B0	2133
0B20:	26	AB	56	2F	11	AB	55	AE	11	AB	5D	A5	11	AB	5D	A0	1667
0B30:	11	AB	66	20	12	AD	66	3D	06	B0	6C	B8	16	D0	6E	3C	1550
0B40:	06	CA	75	B0	46	AB	7C	3E	4A	AB	8E	21	5A	AD	86	A0	1918
0B50:	6A	AF	7E	A0	6A	B1	76	3E	A6	B2	3C	BC	41	D4	43	38	2022
0B60:	B1	AF	68	B0	54	D2	40	30	50	D7	D4	B0	5A	B7	6D	30	2103
0B70:	AA	B8	6F	2F	EA	B8	77	20	EA	B8	77	20	9A	B7	66	3E	2151
0B80:	9E	B7	4D	21	AB	B8	5E	AA	EA	B9	5E	A6	D6	B9	66	A9	2419
0B90:	E4	B9	66	23	B4	B8	44	BF	14	B8	4C	30	15	B6	A5	30	1869
0BA0:	01	99	FD	30	10	B6	FD	30	16	B6	FD	30	16	B6	FD	30	1964
0BB0:	16	B6	FD	30	16	B5	FD	30	55	B1	74	30	BB	B1	96	B1	2126
0BC0:	BB	B0	96	B2	7B	B0	9E	BC	BB	B1	A6	B9	BB	B3	9E	3C	2635
0BD0:	B7	B4	85	BC	A7	B4	5C	BD	CB	B3	34	20	CC	B4	1B	A0	2362
0BE0:	C8	B9	23	20	CC	B7	12	BE	C8	B7	1A	3F	9C	B7	18	30	1934
0BF0:	00	44	6C	39	C0	D5	20	10	80	B6	25	20	00	B6	46	A0	1477
0C00:	DE	B6	5F	20	D7	B5	67	A0	9B	B4	67	21	EA	B4	67	22	2217
0C10:	EA	B4	6F	21	EA	B4	77	25	EB	B5	77	A6	AA	B5	6F	26	2329
0C20:	E6	B5	6E	AB	DA	94	75	A7	D6	B4	4D	27	DA	B4	4C	28	2363
0C30:	C0	B6	58	30	00	44	65	41	00	99	F8	10	04	B3	76	20	1494

0C40:	16 B4 67 A0 16 F4 67 A0 12 B3 67 A2 13 B3 67 A1	1854
0C50:	13 B3 67 A0 15 B3 67 A1 12 B4 67 25 16 B5 7F A6	1759
0C60:	12 05 A7 7A 01 B4 7E 2B 90 B5 4D 20 C0 B5 4E A0	1832
0C70:	01 B6 51 B0 D1 B6 40 10 00 B8 23 3C 44 B6 2B 10	1624
0C80:	C4 AF 32 B0 C5 AE 2B A0 C4 B3 04 A0 55 AD 6E A	2256
0C90:	5D AD 7E A4 5A A4 9E 26 59 61 6C 26 57 A5 9E A	2049
0CA0:	45 AC 96 A7 14 AC 7C AC 55 AC 66 A0 20 B0 56 BE	2040
0CB0:	11 B0 56 B1 1A B0 56 36 45 B5 56 30 45 B7 57 30	1579
0CC0:	05 B6 57 30 15 B3 56 B1 58 B4 5E 31 54 B2 5E 30	1613
0CD0:	96 91 65 B0 96 B0 55 3C 97 B0 55 3E 9A AF 4C BF	2126
0CE0:	9A AE 4C 3F A6 AD 44 3E A5 AC 4B A0 95 AE 4B 30	1954
0CF0:	95 AD 4B 30 91 AE 53 30 50 AC 53 30 80 AF 53 20	1696
0D00:	0B AE 5B 20 A6 AE 5B A0 69 AF 63 A0 AD AE 7C 20	2146
0D10:	69 AE 84 A0 24 AD D4 B0 51 B0 F4 30 32 90 C4 30	2155
0D20:	70 B1 C4 B0 64 B3 B4 F0 62 B3 8B B0 62 B3 8B 30	2349
0D30:	02 14 B5 67 00 99 F8 10 15 B3 0A B0 01 B0 03 20	1273
0D40:	59 B2 64 20 A9 B0 7D 2B F9 B1 96 2B A9 B1 9F 22	2064
0D50:	A5 B1 9F 20 A8 D2 97 20 E8 D1 80 A1 54 B3 89 30	2285
0D60:	54 B3 8B 30 54 B3 8B 30 00 B3 8D B0 40 F9 8F B0	2022
0D70:	12 D1 8E B0 12 D1 8D B0 95 B1 96 22 D9 B1 9E A2	2313
0D80:	05 B3 96 BF D4 B4 96 BF 98 B3 8D BC 54 B1 7A B0	2685
0D90:	00 8D 9B 30 00 8D 9B 30 00 8D 9B 30 91 B4 FC 30	1648
0DA0:	61 B2 E5 B0 9B AE 8D 2F AB AF 9D A3 A4 AE A6 22	2395
0DB0:	A5 AC A6 22 BA AC 9E 20 A9 AC 9D BE 96 CE 95 3C	2338
0DC0:	A6 D0 9D A3 A5 B2 A6 A7 E4 D5 AF 21 AB D6 AF 3F	2639
0DD0:	68 D7 9D B8 A4 D7 5C B4 12 D8 54 B0 12 D9 65 30	2192
0DE0:	12 D7 6D B0 12 F6 7D B0 12 F7 85 B0 02 D5 93 B0	2195
0DF0:	00 B7 BA 30 00 B8 E3 B0 14 D6 EE B0 10 D5 A5 B0	2222
0E00:	45 B1 85 A6 6A AB 95 A2 BA AB 8D A0 71 AB 95 20	2256
0E10:	72 8B 95 20 62 6C A5 A1 51 8E A6 20 60 CF 75 BE	1997
0E20:	A6 B0 4D 3A B1 B2 4C BE 82 D6 4C 3E 84 D7 5B 31	2019
0E30:	51 B7 B4 30 91 D5 B5 AE EB D3 9E A7 D4 D3 A6 A0	2626
0E40:	E8 B1 A6 BE 99 B0 AE BF A9 AE AE A0 A9 AE AE 3F	2716
0E50:	B9 AE AE 3F B2 AE AD A0 A2 AE AD A1 A3 8D AD 22	2462
0E60:	53 8C B5 A1 53 8D 8D RE 52 8E AD 22 B5 AE A4 20	2150
0E70:	45 B0 B1 20 45 B0 B0 20 45 B0 B0 34 45 B0 B0 35	1854
0E80:	98 B9 8D 36 D4 D9 7D B8 EB DA 85 AF EC DA 85 24	2654
0E90:	E8 D9 85 3F 90 D8 7B 3E 80 B4 78 B0 80 B4 78 30	2270
0EA0:	80 B4 78 30 80 B4 78 30 44 D9 8C 30 94 D7 95 27	1976
0EB0:	98 D5 AE 20 E2 B5 C5 BD EB B5 C6 3F E4 B5 C6 3F	2708
0EC0:	94 B3 AD 20 50 B3 59 30 50 B3 58 30 50 B3 58 30	1718
0ED0:	50 B3 58 30 01 D6 5C B0 01 D7 5D B0 01 D8 65 30	1729
0EE0:	02 F7 65 30 02 FA 6D 30 12 F8 6D 30 02 F8 75 30	1645
0EF0:	02 F7 74 B0 02 F6 7D 30 01 F6 7D 30 01 F7 85 B0	1939
0F00:	11 F5 85 30 E4 D4 8D A0 9B B3 A6 3E 94 B2 B6 3D	2312
0F10:	98 B0 B5 BE 94 AF B5 3F A8 BE AD 20 50 AD A4 A3	2353
0F20:	A2 8C 9C A2 92 8C 9C 22 90 8C A4 24 50 A9 B4 26	2047
0F30:	61 A9 94 26 64 AA 93 A4 98 CF 93 24 54 CB 89 3F	2062
0F40:	25 B4 70 10 01 F0 5D 5E 25 B4 70 10 69 B4 7C A0	1687
0F50:	99 B4 7C 20 56 B5 74 20 05 B4 5B BF 05 B3 64 3D	1716
0F60:	15 B4 7C A7 59 B3 85 21 A9 B4 95 2D E9 B4 96 2C	2076
0F70:	AE B5 9E A7 BA B6 9F 23 5D B7 8F 22 55 B8 6F 20	2107
0F80:	A4 D9 56 3B A4 FA 45 BD B4 FA 45 20 64 FA 45 20	2180
0F90:	B8 FA 4D A0 B4 DA 56 2A 74 DA 6F 22 34 DA BF 20	2121
0FA0:	2B DA 9F A0 6B DA A7 A0 64 D9 A7 20 94 D8 A6 BF	2463
0FB0:	D8 D7 A6 BF AB D6 A6 3D 54 D6 A5 B1 65 D5 A5 30	2564
0FC0:	A5 B4 B5 B0 9B B4 CE A9 9B B3 D6 A0 9C B4 D6 BF	2855
0FD0:	9C B4 CE BF 9C B1 A5 BC 54 B2 62 B0 10 D4 90 30	2375
0FE0:	10 D4 90 30 40 B5 CB 30 45 D5 CB B0 40 D9 FC 30	2158
0FF0:	00 B5 74 A0 9B B2 95 A0 9C B4 96 23 FD B5 96 A0	2341
1000:	B0 B5 96 31 65 96 75 3C 50 97 4D 31 91 B6 44 14	1925
1010:	99 B6 44 BF A4 D8 7D A1 AB D8 B1 A5 AB D8 96 A2	2647
1020:	9B D9 96 A0 94 D9 96 A2 94 B9 96 A0 94 D9 9E A2	2684
1030:	94 D9 9E A2 A4 D9 96 A3 A4 D9 96 A3 64 DA 96 A3	2704
1040:	6B DA 9E A2 6B D9 8E 3E FB B9 75 BE F9 B9 74 FE	2647
1050:	8D B8 B4 30 B9 B8 6E 30 AD B7 64 30 6C B7 65 3B	2929

1060:	74 B7 50 A1 A5 E7 55 A0 90 B9 4C BE E1 B9 30 BA	2397
1070:	A2 D9 3C BE 20 D9 04 A0 21 B8 13 30 00 B8 F8 30	1804
1080:	00 B8 F8 30 00 B8 F8 30 40 D8 C3 E0 48 D8 9C A0	2218
1090:	A3 D8 9D A9 EB D8 9E 24 D6 D7 A6 A0 98 B6 BE E0	2819
10A0:	94 D5 BE A0 98 B3 AE 3D A4 B5 9E 3F D8 B6 96 21	2424
10B0:	D8 B8 BE 20 A8 D9 75 BE A8 D9 65 BE 94 DA 6E 20	2451
10C0:	94 D9 7E A0 98 BE BE A0 E8 B6 97 20 E8 B6 9F 3F	2522
10D0:	98 B4 AF 3F 58 B1 B6 BF A9 AF A6 3D 5A B1 95 37	2250
10E0:	05 D0 9D 30 B4 D1 7C 30 80 94 7C 30 05 BE 04 23	1565
10F0:	14 BE 04 A1 11 8C 04 BF 02 8C 04 BE 02 68 04 3E	1190
1100:	44 B0 94 B0 90 8D 8C BF 91 AF A4 BE 98 AF AC B0	2559
1110:	98 AF AC BF 94 AE A4 BF 91 BE 9C 3F 91 8F 8B A0	2460
1120:	00 AF B2 BF 21 49 C2 30 41 97 C1 B0 D0 D6 B9 30	2084
1130:	84 D7 C0 30 00 5C 4C 2F 45 B4 38 10 01 D6 EF 30	1625
1140:	11 B1 FE 30 84 EF F4 B0 22 AF 04 30 12 AB B3 B0	2092
1150:	43 AA A3 30 9A AA 7C A0 9A A9 7C 3E E9 AC 85 20	2135
1160:	A6 AC 9E 22 67 AE B6 21 A7 AD BE 22 B6 AF BE A2	2292
1170:	B7 AF BF 24 B6 AF BE A0 B6 B0 B6 3F A4 CF B0 A3	2618
1180:	71 AF 44 23 92 AF 4B 2F 51 B0 51 B0 51 B1 58 10	1710
1190:	00 78 30 47 04 B1 F8 10 02 F6 AB 30 12 F7 A4 30	1628
11A0:	02 F9 9C B0 02 F7 9D 30 02 F6 95 30 02 F7 BE 30	1921
11B0:	02 F7 B6 30 12 F6 7E 30 64 B5 7E A0 64 B5 7F 20	1876
11C0:	61 B5 7F 20 61 B5 87 40 51 B5 7F 3C 61 B5 7E BB	1954
11D0:	51 B5 76 34 10 B6 7C BE 10 D6 85 30 14 D6 B3 B0	1896
11E0:	20 B7 89 B0 00 E5 A8 70 04 B6 D0 70 04 B6 F0 70	2033
11F0:	44 B9 FE 30 00 DA F5 E0 01 B8 FD 30 48 D9 94 B0	2294
1200:	58 D8 7B 30 58 D8 78 10 00 74 21 3D 00 99 FB 10	1542
1210:	01 B9 FF B0 96 B8 77 A0 A5 B7 9F A0 A9 B5 C7 A0	2606
1220:	A9 B4 CF A0 A9 B3 CF 23 A8 B3 CF 22 A8 B4 CF 29	2490
1230:	58 B5 C6 AD 54 B5 9D A5 64 D5 AA 30 44 B5 B9 B0	2368
1240:	90 B5 AB 30 90 B5 AB 30 90 B5 AB 30 90 B5 AB 30	2164
1250:	44 BA FB B0 40 D6 E5 B0 14 B4 FD 30 B2 D7 FD B0	2639
1260:	40 F5 FC B0 30 73 F8 30 90 B2 5B B1 60 D4 64 B1	2374
1270:	20 B5 78 B1 A4 B5 5A B1 A4 D5 48 10 04 EC 28 42	1936
1280:	45 B7 70 10 E6 D9 5E A0 D6 D9 5F 20 EA D9 67 3E	2255
1290:	EA D9 67 3C EA D9 5E BE EA D9 5E A0 EA D9 5E A0	2759
12A0:	EA D9 5E BF EA D9 5E BF EA D9 5E A0 EA D9 5E 20	2761
12B0:	D5 D9 55 20 D1 BA 5C A2 D5 B9 5C 3F D0 BA 5A A0	2393
12C0:	94 D5 60 70 D4 B2 30 70 C0 B4 28 50 C0 B4 2A 30	2073
12D0:	55 B5 26 3D 55 D9 3D A7 96 D8 3D 23 91 D9 46 24	1825
12E0:	96 D8 56 A5 96 D8 5F 25 6A D8 6F 21 6A B7 7F A0	2157
12F0:	2A B6 97 A2 2A B6 97 A0 2A B5 8F A1 1B B5 67 3E	1972
1300:	6B B4 34 BA A5 B3 3B 3D D1 B3 2A 3B 10 D6 05 3D	1774
1310:	06 B4 4F 2A 5B B3 77 A0 5A B0 77 20 5A AE 7E A1	1824
1320:	1A AB 76 22 4A AD BE 25 1A AD 66 23 2A AF 4E A0	1566
1330:	1A B0 2D A0 14 AF 42 30 B1 B2 71 30 45 D5 7B 30	1637
1340:	15 D3 07 B0 42 B5 FE B0 22 B1 04 B0 21 B1 B5 30	1874
1350:	5A B0 B0 31 BA B0 BE B8 B5 B1 B6 3E 6A B2 B6 B0	2305
1360:	6A B2 75 B9 16 B5 04 B0 15 B3 06 30 16 B5 F6 B0	1848
1370:	16 B7 FE 30 16 B7 FE B0 25 B5 FE B0 45 B5 FE 30	2342
1380:	45 B5 FD 30 C5 B6 FD 30 C9 B9 FD 30 26 AF A5 3F	2359
1390:	75 B0 B6 20 B9 B0 BE 20 B8 B1 C6 20 B8 B1 C6 3F	2303
13A0:	B8 B1 CE 40 B8 B1 CE C1 78 B1 CE C1 B8 B2 C6 A2	2809
13B0:	64 B2 C6 A2 64 B2 B6 A2 68 B3 A6 A2 64 B4 B6 20	2317
13C0:	A8 B3 5C B0 A5 B4 5C 20 55 D4 44 20 66 D4 4C 22	1918
13D0:	66 D4 64 A3 BA D4 6D A5 B4 B3 75 24 64 B3 74 A5	2321
13E0:	50 B4 68 30 50 D4 48 30 00 95 40 70 00 95 40 70	1477
13F0:	00 95 40 70 00 95 40 70 00 95 40 70 00 95 40 70	1300
1400:	80 BA FD 30 10 E7 FE E0 21 B6 4D AA 57 95 57 3B	2088
1410:	56 95 67 3C 56 94 6F 3F 56 92 66 BE 16 90 3D BF	1748
1420:	01 AD B5 20 02 B8 04 A0 02 AC 45 22 03 AF 4D A2	1210
1430:	07 A0 0E A1 47 B6 4F 20 52 B7 46 BC 01 B6 4D B0	1677
1440:	01 F3 55 30 11 F5 66 30 A9 B5 7F A1 A5 B5 B7 A2	2070
1450:	A6 B5 B7 3F A9 B5 76 BE 5L B6 4D BE AA B7 6E A1	2367
1460:	EA B8 6F A0 EA D8 6F A0 AA BE 6F A0 66 B6 56 3B	2464
1470:	9C 94 44 62 9D 94 44 A1 9C 94 44 A0 69 B2 76 A0	2161

1480:	69 B0 AF 20 A6 E3 C7 3F AA B3 C7 20 6A B3 BF 20	2190
1490:	6A B2 B0 A0 54 B2 4A B0 A4 B2 92 30 90 B3 B7 30	2141
14A0:	51 B3 70 30 09 B6 B7 30 14 B4 B4 B0 56 AF BC B0	2023
14B0:	25 AA AC B0 11 AA AC B0 11 AD 9F B0 3B AC 95 37	2046
14C0:	5B AC B5 A0 27 AB 9E 21 26 AB 8C 20 22 AB 96 20	1775
14D0:	22 AC B6 20 23 AB 56 21 27 AA 2E 21 37 AA 06 21	1248
14E0:	37 AA 06 22 37 AB 06 22 26 AC 6E 22 26 AD B6 A2	1392
14F0:	2A AE 6E 20 9E AF 44 3E B5 B7 3B BF 11 B2 3E 30	1667
1500:	5B B1 76 44 67 B1 B6 3C 56 B1 B5 A2 5B B0 BE A4	2160
1510:	AB B1 97 23 B8 B1 97 23 76 B2 9F 20 B9 B2 96 3A	2062
1520:	A5 B2 95 37 64 B3 9C 20 51 B1 B1 B0 00 B2 40 70	1931
1530:	00 B2 40 70 00 B2 40 70 00 B2 40 70 10 B0 70 50	1446
1540:	52 AE 9D AC 66 AE 9E A6 77 AE 9E A1 2B AD A7 20	2212
1550:	27 AD 96 A0 1B AD 7E 22 1B AC 05 A0 1A AB 05 20	1480
1560:	01 AA 04 BF 06 AF 3C A1 B1 96 4C A4 95 B1 55 21	1731
1570:	AB B0 55 A0 9A B4 5E BE 9A B7 56 BC 51 DB 36 B9	2357
1580:	11 D7 06 3C 16 D7 05 A0 69 B7 5E 22 A9 B7 67 A4	1735
1590:	AD B8 6F A1 A9 B8 77 BE ED B7 77 3E ED B6 66 BC	2601
15A0:	D9 B5 2B BA B4 B4 32 B0 B1 93 3A B0 53 B4 44 B0	2186
15B0:	AE B4 77 22 6A B2 7F A2 26 B0 77 A0 16 AD 06 A0	1934
15C0:	16 AA 06 23 16 AB 06 A2 27 AC 06 A0 27 CF 2F 20	1296
15D0:	67 D1 4E 3E B6 B4 34 3C B2 B5 34 3F 5B B6 4D BF	1845
15E0:	AB B7 56 20 AE B7 4E 20 AA B7 4E 20 AA B7 4E 3E	1895
15F0:	96 B7 4E 3D A7 D7 3E 3C AB B7 3D A1 EB B7 3D A0	2191
1600:	A6 B8 45 A1 E7 D8 4E A0 E6 D8 46 BF E6 D9 3E 3F	2544
1610:	E8 F9 3E 20 E2 D9 3E 3F E2 D9 3E 20 E6 D9 3E 20	2221
1620:	E6 D9 3E 21 E7 D9 46 A0 E7 D9 46 A1 E7 B8 4E A0	2552
1630:	A7 B7 4E 3F 57 B6 56 3E 17 B5 3D BF 17 B3 55 A0	1811
1640:	1A AF 65 A2 56 AE 75 BF 16 AB 5C 3C 56 AE B5 25	1807
1650:	5B AE 95 A0 6B AF 96 A4 7B B0 96 A1 BA B1 BE 20	2317
1660:	F6 92 B5 20 E2 92 7C A0 E4 B1 72 B0 D0 B2 B1 B0	2599
1670:	50 B3 70 70 50 B3 70 70 50 B3 70 70 50 B3 70 70	1932
1680:	50 B3 70 50 44 B7 96 B0 69 B2 A7 2F AE B2 C7 27	2115
1690:	B9 B3 CF 20 BA B3 CF 22 AA B3 CF 20 6B B2 BE BE	2462
16A0:	5B B1 A5 3A 27 AF B4 3C 27 AE B3 A0 27 AC 74 A0	1888
16B0:	27 AC 7D 20 27 AC B0 A2 2B AD 9E A2 6B AE A6 A4	2029
16C0:	6B B0 A7 21 7F B1 9F 21 AA B4 97 20 AA B4 5E A0	2116
16D0:	9E B5 5E A0 9B B7 56 22 E6 B8 5F 3D E5 D9 5F 3F	2225
16E0:	E5 DA 56 A0 E6 B9 56 A0 CA B6 3D 3B CD B4 35 20	2328
16F0:	CD B5 35 3F CE B4 35 3E B8 B7 3D BF D8 FA 3E 3F	2168
1700:	E8 FA 3D BE E2 DA 3D 3E AB FB 34 BF B8 FB 23 30	2432
1710:	B0 BA 30 30 45 B6 20 30 45 B6 20 30 14 92 B4 B0	1546
1720:	11 94 F5 30 20 B2 FC B0 A0 B1 FD B0 31 AF CE 30	2340
1730:	51 CE CD B0 21 AD B4 B9 51 AE 9C BE 41 A6 9C BF	2418
1740:	AA A7 74 BF AA A6 74 BE AA A7 74 20 AA A9 7C 3D	2295
1750:	AA AA B4 3D 5A AB B4 3E 5A B0 B4 3D 5A B6 A3 3E	1944
1760:	5A B8 A9 B0 B0 99 B0 10 00 9B 2A 47 44 B8 E0 10	1849
1770:	A7 AC 74 20 A7 AC 75 A0 A7 AD B6 A3 9B AB BE BF	2399
1780:	1A A5 A6 BF 46 AB 95 BC 55 AA 7C 3A 6A A9 5B BA	2112
1790:	27 AB 43 30 06 AD 43 B0 00 D4 4C 35 15 B4 4D 20	1395
17A0:	00 B2 4D A1 11 B3 6D A1 15 B2 9D A1 85 B0 96 21	1763
17B0:	66 AC 7D A1 6B BE B6 22 6B BF 96 A1 6B 90 9F 3E	2010
17C0:	7A 91 A6 BF BA 91 AE BE BA 91 BE 3F B4 B1 B6 3F	2505
17D0:	B2 91 C6 20 B5 B2 C6 20 A0 B2 C6 3F 54 B2 BD A0	2352
17E0:	A5 B1 AD BF 4B AE BD A0 B5 B2 C6 3F AA B1 BE 21	2491
17F0:	B6 B1 AD A0 6A B0 B4 BF 76 B0 74 A1 71 B0 5C BF	2440

— AU CLAIR DE LA LUNE et DU BON TABAC. Il s'agit ici des codes de deux expressions chantées créées au moyen d'un programme de chant (CHANT-64 pour C64 ou MZ CHANTEUR pour MZ-700). Il n'y a pas de table d'adresses pour ces deux expressions isolées.

CLAIR DE LUNL

0000:	02 BC 00 41 C7 AC B4 A0 C7 AC B4 A0 C6 AE B5 20	2067
0010:	C6 AB B5 20 C6 AB B5 20 C6 AB B5 20 D6 AB B4 A0	2279
0020:	D6 AB B4 A0 09 97 88 30 0D 97 8F B0 09 97 8E B0	1982
0030:	67 96 B6 20 74 B5 55 20 74 B5 55 20 74 B6 2E 20	1623
0040:	74 B6 2E 20 28 B5 55 20 A6 B6 B5 20 A6 B6 B5 20	1836
0050:	B6 B6 BF 20 B6 B6 BF 20 B6 B6 B7 20 B6 B6 B7 20	2332
0060:	FA B7 A5 20 FA B7 A5 20 3A B3 B4 20 36 B3 B0 A0	2195
0070:	79 B2 96 20 97 B1 CD 20 B0 B5 18 20 B0 B5 1E A0	1907
0080:	10 D5 04 A0 A0 D5 0C 20 40 D2 2C A0 40 D9 5E 20	1695
0090:	AF B3 B5 20 AF B3 B5 20 AE B3 7E 20 AE B3 7E 20	2054
00A0:	AF B3 B6 20 AB B3 B6 20 FF B3 B5 20 FF B3 B5 20	2230
00B0:	74 B5 55 28 74 B5 55 20 74 B6 2E 20 74 B6 2E 20	1588
00C0:	28 B5 55 20 B6 B3 CD A0 B6 B3 CD A0 B6 B2 D6 A0	2380
00D0:	B6 B2 D6 A0 96 B2 CE A0 96 B2 CE A0 97 B1 CD A0	2767
00E0:	97 B1 CD A0 74 B5 55 29 74 B5 55 20 74 B6 2E 20	1906
00F0:	74 B6 2E 20 28 B5 55 20 B7 B7 5D 20 B7 B7 5D 20	1696
0100:	B7 B7 66 C0 B7 B7 66 C0 B7 B7 66 C0 B7 B7 66 C0	2640
0110:	B7 B7 5D 40 B7 B7 5D 40 B8 B4 55 37 48 B4 5E 20	1880
0120:	4C B4 5F 20 4A B4 5E 20 AF B3 B5 20 AF B3 B5 20	1801
0130:	AB B3 7E 40 AB B3 7E 40 AB B3 B6 40 AB B3 B6 40	2176
0140:	FF B3 B5 40 FF B3 B5 40 4C B4 5C B8 4C B4 5C A0	2302
0150:	4C B4 5D 20 4C B4 5D A0 4A B4 5E A0 4A B4 5E 20	1778
0160:	4A B4 5D A0 4A B4 5D 20 B2 D1 A5 20 B2 D1 A5 A0	2086
0170:	B3 CF AE A0 C3 CE AE A0 C3 D1 9E A0 D2 D3 96 20	2732
0180:	92 D2 95 20 92 D2 95 20 B8 B4 55 2F 48 B4 5E 22	1902
0190:	4C B4 5F 20 4A B4 5E 20 B6 B3 CD A0 B6 B3 CD A0	2119
01A0:	B6 B2 D6 A0 B6 B2 D6 A0 96 B2 CE A0 96 B2 CE A0	2760
01B0:	97 B1 CD A0 97 B1 CD A0 4C B4 5C B7 4C B4 5C A0	2425
01C0:	4C B4 5D 20 4C B4 5D A0 4A B4 5E A0 4A B4 5E 20	1778
01D0:	4A B4 5D A0 4A B4 5D 20 AF DA 5D 20 AF DA 5D 20	1922
01E0:	AF DA 5E 20 AF DA 5E 20 AF DA 66 20 AF DA 66 20	2092
01F0:	AF DA 65 20 AF DA 65 20 C2 B9 38 30 C2 B9 38 30	2018
0200:	11 B6 97 B0 16 B3 B8 30 5B B2 B0 A0 AF DA 5D 20	2005
0210:	AF DA 5D 20 AF DA 5E 00 AF DA 5E 00 AF DA 66 00	1987
0220:	AF DA 66 00 AF DA 65 00 AF DA 65 00 A6 B6 B5 20	2044
0230:	A6 B6 B5 20 B6 B6 BF 00 B6 B6 BF 00 B6 B6 B7 00	2218
0240:	B6 B6 B7 00 FA B7 A5 00 FA B7 A5 00 3A B3 B4 38	2168
0250:	36 B3 B0 A0 79 B2 96 40 97 B1 CD 40 C7 AC B4 A0	2307
0260:	C7 AC B4 A0 C6 AB B5 40 C6 AB B5 40 C6 AB B5 40	2361
0270:	C6 AB B5 40 D6 AB B4 C0 D6 AB B4 C0 AB B3 B0 20	2494
0280:	AB B3 B0 20 AB B3 B0 20 AB B3 B0 20 00 00 00 00	1530

DU BON TABAC

0000:	02 80 00 41 1E BA 7C 30 1E BA 7D A0 1E BA 76 B0	1594
0010:	1E BA 76 A0 1E BA 75 B0 BB BB 85 A0 BB B8 85 A0	2331
0020:	BB B8 87 20 BB B8 87 20 BB B8 87 20 BB B8 87 20	2152
0030:	FB B8 85 A0 FB B8 85 A0 B0 B5 18 28 B0 B5 1B A0	2325
0040:	10 D5 04 A0 A0 D5 0C 20 40 D2 2C A0 40 D9 5E 20	1695
0050:	B7 B7 5D 20 B7 B7 5D 20 B7 B7 66 A0 B7 B7 66 A0	2238
0060:	B7 B7 66 A0 B7 B7 66 A0 B7 B7 5D 20 B7 B7 5D 20	2238
0070:	80 D6 18 29 B0 D6 1A A0 90 B2 13 A0 A4 E2 13 A0	1957
0080:	A5 B1 24 A0 EA 91 76 20 B2 D1 A5 20 B2 D1 A5 A0	2267
0090:	B3 CF AE A0 C3 CE AE A0 C3 D1 9E A0 D2 D3 96 20	2732
00A0:	92 D2 95 20 92 D2 95 20 B6 F6 90 30 B6 F6 94 30	2318
00B0:	B6 F6 96 30 9B B4 B0 A0 B6 B3 CD B1 B6 B3 CD B1	2665
00C0:	B6 B2 D6 A0 B6 B2 D6 A0 96 B2 CE A0 96 B2 CE A0	2760
00D0:	97 B1 CD A0 97 B1 CD A0 B0 D6 18 28 B0 D6 1A A0	2320
00E0:	90 B2 13 A0 A4 B2 13 A0 A5 B1 24 A0 EA 91 76 20	2089
00F0:	B6 B3 CD A0 B6 B3 CD A0 B6 B2 D6 C0 B6 B2 D6 C0	2792
0100:	96 B2 CE C0 96 B2 CE C0 97 B1 CD C0 97 B1 CD C0	2902
0110:	B0 B5 18 20 B0 B5 1B A0 10 D5 04 A0 A0 D5 0C 20	1671
0120:	40 D2 2C A0 40 D9 5E 20 52 CA B0 20 52 CA B0 20	1895
0130:	97 CA B0 A0 97 CA B0 A0 97 CA B0 A0 97 CA B0 A0	2808
0140:	B3 CA B0 20 B3 CA B0 20 4C B4 5C A9 4C B4 5C A0	2133
0150:	4C B4 5D 20 4C B4 5D A0 4A B4 5E A0 4A B4 5E 20	1778
0160:	4A B4 5D A0 4A B4 5D 20 B6 B3 CD A0 B6 B3 CD A0	2242
0170:	B6 B2 D6 A0 B6 B2 D6 A0 96 B2 CE A0 96 B2 CE A0	2760
0180:	97 B1 CD A0 97 B1 CD A0 B6 F6 90 30 B6 F6 94 30	2630
0190:	B6 F6 96 30 9B B4 B0 A0 B6 B3 CD A5 B6 B3 CD A0	2623
01A0:	B6 B2 D6 C0 B6 B2 D6 C0 96 B2 CE C0 96 B2 CE C0	2888
01B0:	97 B1 CD C0 97 B1 CD C0 B0 D6 18 20 B0 D6 1A A0	2376
01C0:	90 B2 13 A0 A4 B2 13 A0 A5 B1 24 A0 EA 91 76 20	2089
01D0:	B6 B3 CD A0 B6 B3 CD A0 B6 B2 D6 C0 B6 B2 D6 C0	2792
01E0:	96 B2 CE C0 96 B2 CE C0 97 B1 CD C0 97 B1 CD C0	2902
01F0:	B6 F6 90 30 B6 F6 94 30 B6 F6 96 30 9B B4 B0 A0	2506
0200:	AF DA 5D 3B AF DA 5D 20 AF DA 5E 20 AF DA 5E 20	2101
0210:	AF DA 66 20 AF DA 66 20 AF DA 65 20 AF DA 65 20	2106
0220:	A6 B6 B5 20 A6 B6 B5 20 B6 B6 BF 20 B6 B6 BF 20	2296
0230:	B6 B6 B7 20 B6 B6 B7 20 FA B7 A5 20 FA B7 A5 20	2418
0240:	3A B3 B4 20 36 B3 B0 A0 79 B2 96 40 97 B1 CD 40	2045
0250:	AF B3 B5 20 AF B3 B5 20 AB B3 7E 20 AB B3 7E 20	2054
0260:	AB B3 B6 20 AB B3 B6 20 FF B3 B5 20 FF B3 B5 20	2230
0270:	AB B3 B0 20 AB B3 B0 20 AB B3 B0 20 AB B3 B0 20	2040

# Caractéristiques principales du MEA 8000

---

Cette rubrique indique les principales caractéristiques du MEA 8000 (brochage, tensions, chronogrammes des signaux). Associées aux explications du chapitre 3, et aux exemples d'application décrits dans ce livre, ces données suffiront en général à la mise en œuvre du circuit pour d'autres applications. Pour plus de détail, on pourra se reporter à la documentation du constructeur. Ces caractéristiques sont extraites de la documentation RTC.

## — Valeurs à ne pas dépasser (risque de destruction au-delà)

Tension d'alimentation (toutes bornes sauf 2 et 20)	- 0,5 à + 7 V
Tension d'alimentation des bornes 2 et 20 (via R.)	max. + 15 V
Température de stockage	- 20 à + 125° C
Température ambiante de fonctionnement	0 à + 70° C

## — Caractéristiques statiques

Tension d'alimentation	5 V ± 10%
Courant d'alimentation	typ. 30 mA
Niveaux logiques, toutes entrées et sorties (1)	TTL
Courant de référence (borne 19) (2)	max. 0,3 mA
Courant de sortie (borne 20) à Iref = 0,1 mA (3)	typ. 1,7 mA
Fréquence d'horloge (Quartz ou horloge externe)	typ. 4,0 MHz

## — Caractéristiques temporelles (voir chronogrammes p. 308)

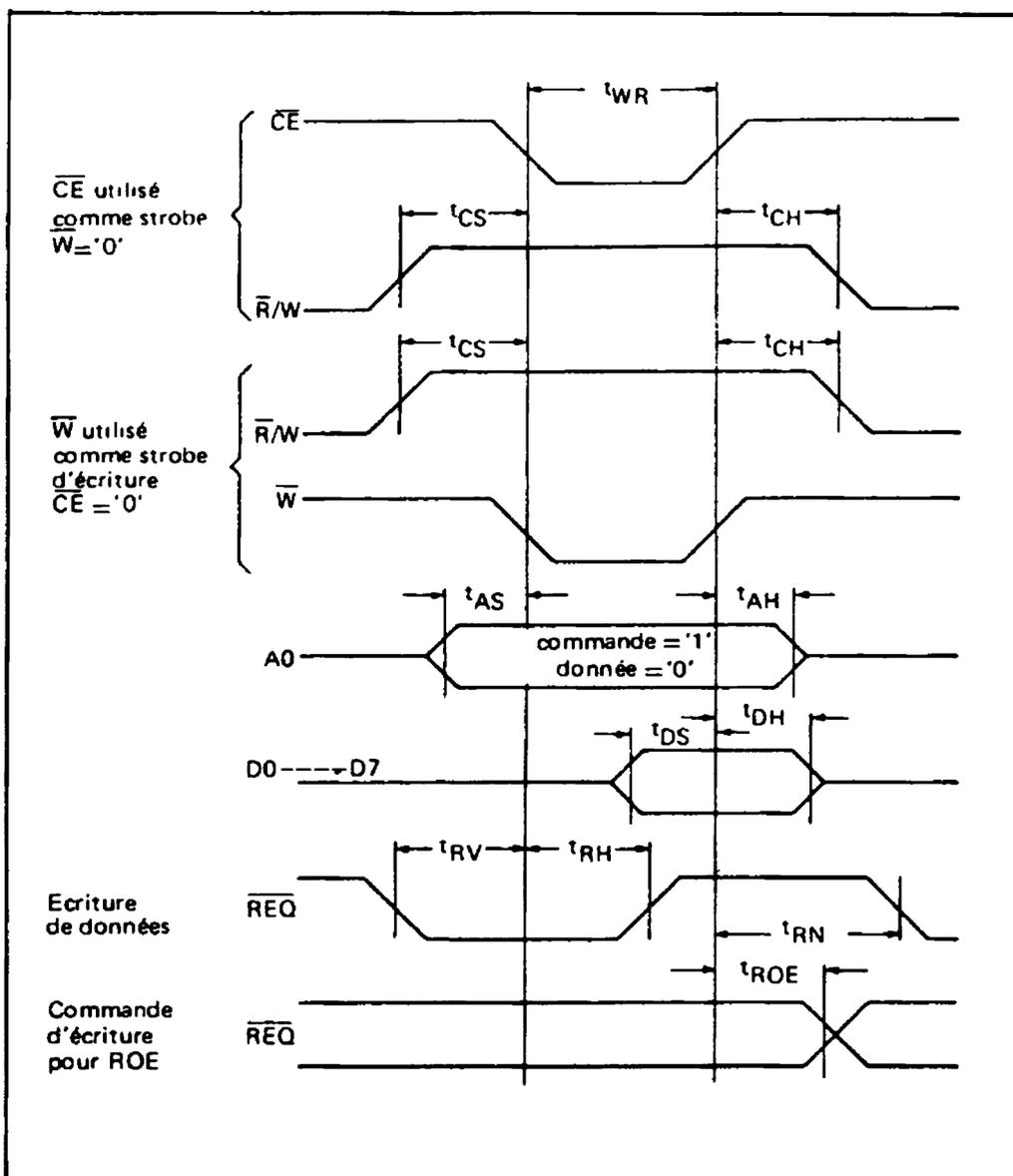
Temps de.... : (4)	
Validation d'écriture	t <sub>WR</sub> min. 200 ns
Etablissement d'adresse	t <sub>AS</sub> min. 30 ns
Maintien d'adresse	t <sub>AH</sub> min. 30 ns
Etablissement des données (écriture)	t <sub>DS</sub> min. 150 ns
Maintien des données (écriture)	t <sub>DH</sub> min. 30 ns
Maintien de REQ	t <sub>RH</sub> max. 350 ns
Apparition du signal REQ suivant (5)	t <sub>RN</sub> max. 3 μs
Validation de lecture	t <sub>RD</sub> min. 200 ns
Apparition des données en lecture	t <sub>DD</sub> max. 150 ns
Disparition des données en lecture	t <sub>DF</sub> max. 150 ns
Validité de REQ avant écriture	t <sub>RV</sub> min. 0 ns
Réponse à la demande de validation de sortie	t <sub>ROE</sub> max. 350 ns
Etablissement de contrôle	t <sub>CS</sub> max. 20 ns
Maintien de contrôle	t <sub>CH</sub> max. 20 ns

Notes:

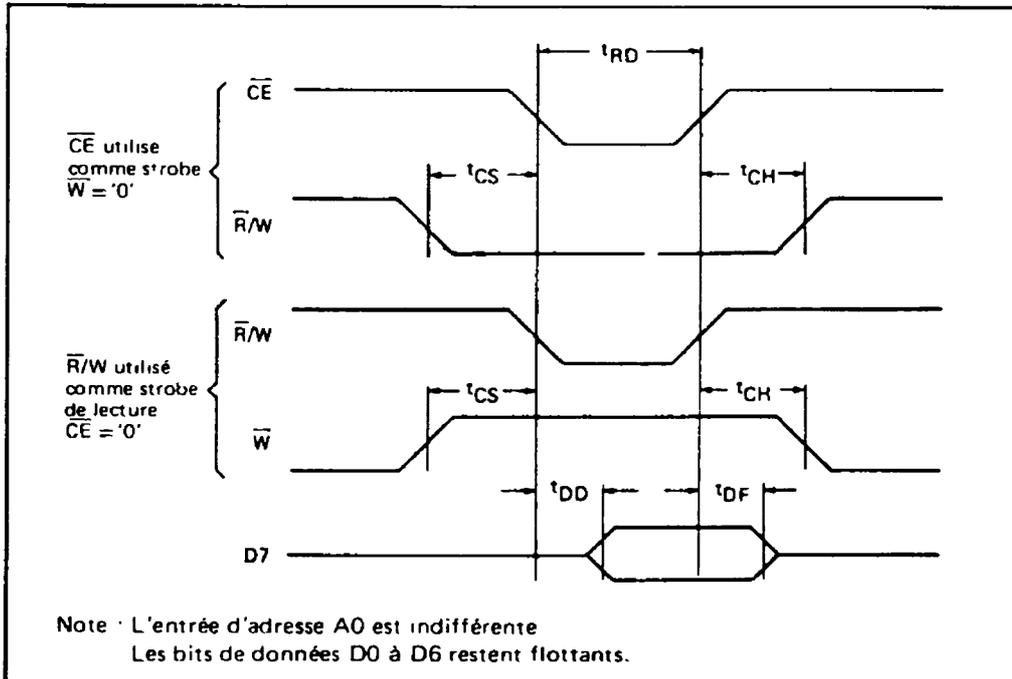
- (1) REQ est une sortie "drain ouvert", nécessitant une résistance de rappel externe ("pull-up" de 3,3 KΩ à 10 KΩ).
- (2) Déterminé par résistance externe au + 5V.
- (3) Sortie "drain ouvert" (résistance de rappel de 470 Ω au + 5V).
- (4) Niveau de référence: 1,5 V.
- (5) Entre deux opérations d'écriture à l'intérieur d'une même trame.

— Chronogrammes

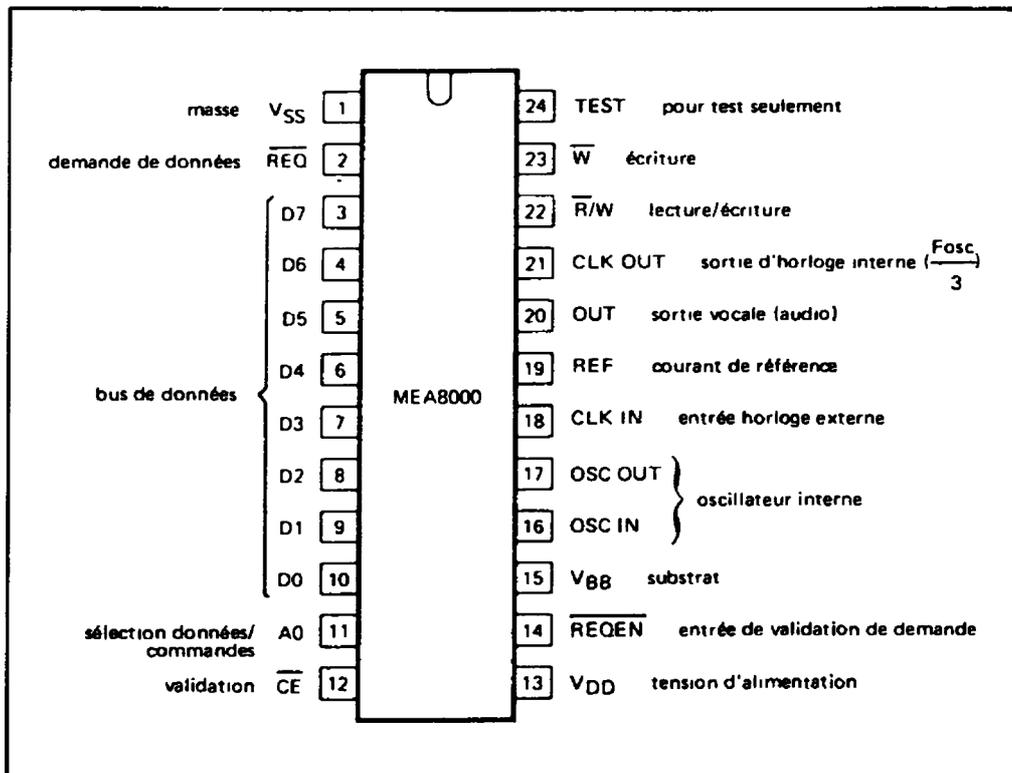
Ecriture



## Lecture



## — Brochage



# Glossaire des termes employés

On trouvera ci-après l'explication des termes spécifiques à la synthèse de parole employés dans ce livre. La connaissance de ces termes facilitera la lecture d'autres ouvrages au lecteur désireux d'approfondir le sujet.

## **Analyse**

En synthèse de parole, désigne le processus permettant de décomposer le signal vocal en éléments plus simples, fondamentaux et formants par exemple. Cette phase utilise en général la technique de la "prédiction linéaire" (ou L.P.C.= Linear Predictive Coding).

## **Coarticulation**

Modification d'un son par son prédécesseur et son successeur lors de la parole. Cause principale des limitations des phonèmes.

## **Concaténation**

Mise bout à bout d'éléments de vocabulaire (phonèmes, diphtongues, mots) pour constituer une expression.

## **Conduit vocal**

Partie de l'appareil vocal situé au-dessus des cordes vocales.

## **Conversion analogique/numérique (C.A.N) ou analogique/digitale**

Transformation d'une grandeur électrique (analogique) en un nombre binaire (numérique) en vue de son traitement par un ordinateur. C'est la phase préalable à l'analyse. Nécessite l'échantillonnage du signal.

## **Conversion numérique/analogique (C.N.A) ou digitale/analogique**

Opération inverse de la précédente permettant la restitution d'un signal audio (analogique) à partir d'échantillons numériques.

## **Diphthongue ou diphtongue**

Segment de parole s'étendant du milieu de la partie stable d'un son (phonème) au milieu de la partie stable de celui qui le suit, donc centré autour de leur transition. On en utilise environ 1200 différents en français pour une synthèse à partir du texte de haute qualité.

**Echantillonnage**

Mesure à intervalles fixes d'un signal analogique, en général en vue de sa conversion sous forme numérique. La fréquence d'échantillonnage doit être au moins égale au double de la fréquence maximale du signal (Théorème de Shannon).

**Fondamental**

Fréquence la plus basse dans le spectre d'un signal périodique. Pour le signal vocal, ceci correspond à la fréquence de vibration des cordes vocales, appelée pitch dans la littérature anglo-saxonne.

**Formant**

Résonance du conduit vocal, favorisant certaines harmoniques du signal. On en compte en général 3 (voix féminine) ou 4 (voix masculine) dans une bande de 4 kHz; leur position relative caractérise le son émis.

**Harmonique**

Composante d'un signal dont la fréquence est un multiple de celle du fondamental. Un signal périodique est décomposable en une somme de signaux harmoniques (série de Fourier).

**Hauteur**

En synthèse de parole, utilisé en synonyme de fondamental (voir ce nom).

**Interpolation linéaire**

Calcul d'un paramètre à un instant donné à partir de deux valeurs connues avant et après cet instant, en supposant qu'il varie linéairement dans cet intervalle de temps.

**Locuteur**

Nom désignant la personne qui parle (en anglais, speaker).

**Non voisé**

Désigne un son produit sans vibration des cordes vocales (consonnes "sourdes": f, s, ch, p, t, k). Egaleme nt caractéristique du chuchotement, au cours duquel tous les sons, voyelles comprises, sont non voisés.

**Phonème**

Composant sonore élémentaire du vocabulaire d'une langue. C'est un élément abstrait dont le changement d'un seul dans un mot transforme celui-ci en un autre (exemple: bout et tout ne diffèrent que par leur premier phonème). Leur nombre varie de 30 à 60 selon les langues (37 en français), et tout le vocabulaire d'une langue peut se décomposer à leur aide.

**Pitch**

Terme anglais désignant le fondamental d'un signal vocal. Voir ce nom.

**Prosodie**

Information transportée par le message vocal, supplémentaire par rapport à sa forme écrite, et non contenue dans les phonèmes. C'est l'ensemble formé par l'intonation, l'accent tonique et le rythme de la phrase.

**Reconnaissance vocale**

Technique consistant à reconnaître un segment de parole, en général des mots, parmi un ensemble de segments préalablement stockés dans la mémoire de la machine, afin de lui faire exécuter une tâche précise. La reconnaissance vocale peut être mono-locuteur ou multi-locuteur, par mots isolés ou en parole continue. Elle est souvent associée à la synthèse de parole pour la communication bilatérale homme-machine.

**Règles (synthèse par)**

Synthèse à partir du texte dans laquelle les transitions entre phonèmes sont calculées à partir de modèles d'évolution (règles), et non stockées en mémoire comme dans la synthèse par diphtonges.

**Spectre fréquentiel**

Ce terme désigne la représentation graphique de l'amplitude d'un signal en fonction de la fréquence.

**Synthèse**

Opération inverse de l'analyse. Permet de reconstituer le signal vocal à partir des éléments fournis par l'analyse (caractéristiques de la source et formants par exemple).

**Trame sonore**

Période élémentaire de l'ordre de 10 à 20 millisecondes utilisée pour la reconstruction d'un signal vocal synthétique, et au cours de laquelle celui-ci peut être considéré comme constant.

**Voisé**

Se dit d'un son dont la source est la vibration des cordes vocales. Toutes les voyelles et les consonnes dites "sonores" (b, d, g, j, l, m, n, r, v, z) sont de ce type lors de la parole normale (non chuchotée).

**Voisement**

Etat de vibration des cordes vocales. (Voir voisé et non voisé).

# Adresses utiles

---

Cet ouvrage se voulant essentiellement pratique, voici une liste d'adresses où le lecteur pourra trouver les composants, kits ou réalisations complètes basés sur le MEA 8000.

## Composants (MEA 8000)

Les distributeurs RTC commercialisent la plupart des composants nécessaires aux réalisations décrites, mais ne vendent généralement pas à l'unité et exigent un minimum de facturation par produit. C'est pourquoi le particulier s'adressera plutôt aux revendeurs détaillants.

### Distributeurs RTC (Bureau principal)

#### Région Est

HOHL & DANNER      STRASBOURG      88 20 90 11

#### Région Nord

SANELEC      LILLE      20 98 92 13

#### Région Ouest

SERTRONIQUE      LE MANS      43 84 24 60

#### Région Parisienne

DICOMEL      ANTONY (92)      (1) 46 66 21 82

OMNITECH      SURESNES (92)      (1) 47 72 81 81

R.T.F Diffusion      PARIS 15e      (1) 45 31 16 50

#### Région Rhône-Alpes

DISOLEC      LYON 3e      78 53 00 25

#### Région Sud-Est

AIRTELEC      AIX EN PROVENCE      42 60 01 77

#### Région Sud-Ouest

I.C.C.      TOULOUSE      61 26 14 10

**A l'étranger (pays francophones)**

**Suisse**

PHILIPS ELCOMA, Allmendstrasse 140-142, CH 8027 ZURICH

Tel: (01) 488 22 11

**Belgique**

PHILIPS M.B.L.E., 9 rue du Pavillon, B - 1030 BRUXELLES

Tel: (02) 242 74 00

**Canada**

PHILIPS ELECTRONICS LTD, 601 Milner Avenue, SCARBO-  
ROUGH,

ONTARIO M1B1M8

Tel: 292 51 61

**Points de vente au détail (liste non exhaustive)**

**ACER**

42 bis, Rue de Chabrol, 75010 PARIS

**ATLANTIQUE COMPOSANTS**

27, Chaussée de la Madeleine, 44000 NANTES

**COMPTOIR DU LANGUEDOC**

26, Rue du Languedoc, 31000 TOULOUSE

**DECOCK**

4, Rue Colbert, 59000 LILLE

**ELECTRONIC DIFFUSION**

62, Rue de l'Alouette, 59600 ROUBAIX

**E.N.C.**

2, Place de la Révolution, 50100 CHERBOURG

**FACHOT ELECTRONIQUE**

5, Bd Robert Serot, 57000 METZ

**J. REBOUL**

72, Rue Trépillot, 25000 BESANÇON

**MAGNETIC-FRANCE**

**11, Place de la Nation, 75011 PARIS**

**MICRO PROSS**

**79, Rue du Gal de Gaulle, 68000 COLMAR**

**MIRAGE DES ONDES**

**44, Cours Julien, 13000 MARSEILLE**

**ORMELEC**

**30, Cours Emile Zola, 69300 VILLEURBANNE**

**PENTASONIC**

**5, Rue M. Bourdet, 75016 PARIS**

**RADIO SELL COMPOSANTS**

**17, Rue Gaston Planté, 29239 GOUESNOU**

**SAINT QUENTIN RADIO**

**6, Rue de St Quentin, 75010 PARIS**

**S.N.D.E.**

**9, Rue du Grand St Jean, 34000 MONTPELLIER**

**SODICELEC**

**29, Rue d'Alsace-Lorraine, 33000 BORDEAUX**

**TOUT POUR LA RADIO**

**66, Cours Lafayette, 69003 LYON**

## Modules fonctionnels et kits

On trouvera ci-dessous une liste de réalisations pour micro-ordinateurs, utilisant le MEA 8000, et commercialisées à notre connaissance à la date où nous écrivons ces lignes.

Ordinateur	Produit	Fabricant	Points de vente
THOMSON (MO5, TO7 et TO9)	Module + logiciels	CEDIC-NATHAN	Librairies et bouti- ques informatiques.
AMSTRAD MSX, etc.	Module + logiciels	TECHNI-MUSIQUE <sup>1</sup>	Rue Fontaine du Bac Tél. 73 26 21 04 63000 CLERMONT-Fd
APPLE II	Carte + logiciel (Porte-Parole)	EDICIEL	Librairies et bouti- ques informatiques
ORIC 1	Kit ou circuit imprimé	R. PAULMIER S.A.	40. rue Castagnary Tél. 42 50 19 (X) 75015 PARIS
SHARP MZ-700	Kit	Club des Sharpentiers	151-153 Av. J.Jaurès 93307 AUBERVILLERS Tél. (1) 48 34 93 44

Ces listes n'ont pas la prétention d'être complètes. Vous pourrez sans doute trouver d'autres points de vente dans des publicités de revues d'électronique et de micro-informatique.

<sup>1</sup>Cette société assure également le codage de vocabulaire à la demande pour le MEA 8000.

Pour vous procurer la synthèse vocale "PAROLE" (accompagnée d'une cassette de démonstration) et les logiciels PHONETRAM et NOMAPHON pour micro-ordinateurs THOMSON, veuillez remplir le bon de commande ci-dessous et le renvoyer aux Editions Cedic/Nathan, accompagné de votre règlement.

Cedic/Nathan  
6-10 boulevard Jourdan, 75014 Paris  
Téléphone : (1) 45 65 06 06

NOM .....

PRENOM .....

ADRESSE .....

CODE POSTAL .....

Je souhaite recevoir la synthèse vocale "PAROLE" et les logiciels PHONETRAM et NORMAPHON.

Ci-joint mon règlement par  chèque bancaire  chèque postal  
d'un montant de ..... 765 F  
(+ frais de port) ..... 10 F  
TOTAL ..... 775 F

DATE ..... SIGNATURE .....

Page blanche

L'impression de ce livre  
a été réalisée sur les presses  
des Imprimeries Aubin  
à Poitiers/Ligugé



pour les Éditions Cédic

Achevé d'imprimer en décembre 1985  
N° d'impression, L 20829  
Dépôt légal, décembre 1985

*Imprimé en France*

Page blanche