

SYSTEME INFORMATIQUE POUR LE
TRAITEMENT DE DONNEES VOCALES

La présente invention concerne l'édition de données vocales.

La présente invention a pour objet un système de traitement d'information comportant des moyens d'acquisition
5 de signal continu destinés à l'acquisition d'un signal électrique à variation continue correspondant à un message vocal, des moyens de numérisation destinés à numériser le signal électrique à variation continue pour produire des données vocales discrètes correspondant à la qualité audible
10 du message vocal, des moyens d'acquisition de données discrètes destinés à l'acquisition de données discrètes correspondant à des caractères alphanumériques, des moyens d'acquisition de signaux discrets destinés à l'acquisition de signaux discrets comprenant des ordres d'édition et de
15 commande, une mémoire destinée à l'enregistrement de données sous forme discrète, des moyens de visualisation destinés à produire une visualisation, et un processeur, tous ces éléments étant interconnectés fonctionnellement par des conducteurs de commande et des canaux de transfert de données,
20 avec un programme pour le fonctionnement du processeur enregistré dans la mémoire de façon que le processeur commande le fonctionnement du système pour : enregistrer les données vocales discrètes dans la mémoire, simultanément à l'acquisition du message vocal, enregistrer les données de caractères dans la mémoire simultanément à l'entrée de caractères,
25 établir dans la mémoire un enregistrement de séquence indiquant un ordre unifié des données de message vocal et de caractères, visualiser une séquence de marques de symbole pour les données vocales et de marques de caractères, chaque marque

de symbole représentant un incrément prédéterminé de message vocal acquis et chaque marque de caractère correspondant à l'un des caractères introduits, la séquence visualisée correspondant à la séquence présente dans l'enregistrement, 5 et réviser l'enregistrement de séquence, sous la dépendance d'ordres d'édition introduits, pour tenir compte de changements d'édition dans l'ordre des données vocales et de caractères.

L'invention peut en outre faire intervenir un programme de fonctionnement tel que le processeur commande de 10 plus le fonctionnement du système pour : réagir à des signaux discrets prédéterminés acquis simultanément à l'acquisition du message vocal, pour indiquer dans l'enregistrement de séquence le point auquel chaque signal discret 15 prédéterminé a été acquis ; présenter sur la visualisation une indication distinctive du moment auquel chacun de ces signaux acquis simultanément a été acquis, par rapport à d'autres éléments des données vocales; établir en mémoire un pointeur définissant une position de pointeur dans la séquence 20 de données, présenter dans la visualisation une marque visible correspondant à la position de pointeur ; déplacer, sous la dépendance de signaux d'entrée acquis, la position de pointeur définie dans la séquence et déplacer cette position de façon correspondante dans la visualisation ; produire, 25 sous la dépendance de signaux d'entrée acquis, un signal audio variant de façon continue qui correspond aux données vocales discrètes enregistrées en mémoire, la production de ce signal commençant à un point dans la séquence de données vocales qui correspond à la position de pointeur définie, 30 telle qu'elle est définie au moment considéré, et suivant l'ordre qui est défini au moment considéré dans l'enregistrement de séquence ; et avancer le pointeur dans les données de message vocal, en correspondance avec la progression de la production du signal audio. L'invention peut également faire 35 intervenir un circuit destiné à détecter l'activité d'acqui-

sition d'information audio, et interrompre l'enregistrement des données de message vocal dans la mémoire, en l'absence d'activité.

L'invention offre à un auteur une représentation 5 graphique visible de la structure du texte qu'il dicte, avec des indications qu'il peut insérer concernant des paragraphes ou d'autres divisions fonctionnelles. Elle permet à un auteur d'éditer avec une grande souplesse le texte qu'il dicte, par déplacement, suppression, insertion et reproduction, tandis que la présentation visuelle l'aide à suivre 10 les opérations d'édition et à localiser le point auquel il doit effectuer des révisions d'édition. L'invention permet également à l'auteur d'introduire par un clavier des notes et des instructions intercalaires dans son enregistrement 15 dicté.

L'invention sera mieux comprise à la lecture de la description qui va suivre d'un mode de réalisation, donné à titre d'exemple non limitatif. La suite de la description se réfère au dessin annexé qui représente sous forme de 20 schéma synoptique un système conforme à l'invention.

Un système d'édition de données vocales 10 conforme à l'invention comprend des connexions 12 destinées à acquérir et à fournir un signal électrique à variation continue correspondant à un message vocal. Un signal acquis peut 25 être obtenu à partir d'un microphone 50 ou d'une ligne téléphonique 52, par l'intermédiaire d'un circuit d'interface 54 comme le montre la représentation de la figure, ou d'autres manières. On peut utiliser le signal fourni pour attaquer un haut-parleur 56 comme il est représenté, ou d'autres manières. Les connexions 12 sont connectées à un convertisseur 30 analogique-numérique 14, qui effectue une conversion bidirectionnelle. Le convertisseur 14 est lui-même connecté à un convertisseur série-parallèle 30 fonctionnant dans les deux directions. Un capteur audio 28 est connecté aux connexions 35 12 et il émet un signal de commande qui indique s'il y a une

activité sur le canal d'acquisition de message vocal. Le système 10 comprend également une unité de visualisation 31 qui peut avantageusement comporter un écran cathodique, et un clavier 16, qui comporte une section 18 pour l'entrée de 5 caractères alphanumériques et une section 20 pour l'entrée de signaux d'édition et de commande.

Le système 10 comprend également un processeur 26, qui peut être le modèle Z 80 fabriqué par Zilog, et une mémoire 22 qui est destinée à enregistrer des données sous 10 forme de bits et qui comporte une section 24 dans laquelle est enregistré un programme de fonctionnement. Tous les éléments du système décrit ci-dessus sont interconnectés par un bus de données 58, un bus d'adresse 60 et des conducteurs de commande 62, comme l'indique la figure. Tous les éléments du 15 système 10 décrit ci-dessus sont des articles classiques disponibles dans le commerce, et la manière de les interconnecter est bien connue de l'homme de l'art.

Le programme de fonctionnement de l'éditeur de message vocal qui est enregistré en mémoire commande le 20 fonctionnement du système, en association avec le processeur 26, dans l'accomplissement de toutes les fonctions de l'éditeur de message vocal. Lorsqu'un auteur utilisant le système parle dans un micro, le message vocal acquis par le système sous la forme d'un signal analogique est numérisé et intro- 25 duit en mémoire sous forme discrète. Simultanément, une représentation du message vocal utilisant une série de symboles vocaux, chacun d'eux représentant une seconde de message vocal, est produite et visualisée sur l'écran cathodique. Pendant les pauses de la voix, l'entrée de données est 30 supprimée pour éviter un gaspillage de la capacité de mémoire. Simultanément à la dictée, l'auteur peut introduire par le clavier des signaux d'interruption qui produisent des pointeurs de mémoire indiquant le moment auquel l'introduction a été faite, dans l'enregistrement de données, et qui 35 provoquent la visualisation de symboles vocaux successifs à partir

de la ligne de visualisation suivante, ce qui simule une coupure correspondant à un paragraphe. Simultanément, le système produit un nombre porté en marge pour permettre d'identifier aisément la coupure. L'auteur peut également interrompre la dictée avec un signal introduit au clavier et introduire un texte alphanumérique par le clavier. Ce texte est introduit en mémoire et est visualisé sur l'écran cathodique de visualisation.

Le système fonctionnant sous la commande du programme conserve un enregistrement indiquant une séquence unifiée de données vocales, de données de texte et d'indications de coupure. L'ordre de cette séquence est initialement l'ordre temporel dans lequel le système a acquis les données. Le système produit également un pointeur de mémoire qui indique une position de pointeur dans la séquence de données. Une marque de curseur est présentée sur la visualisation, à une position correspondante. L'auteur peut manipuler ce pointeur et cette marque de curseur liés pour désigner n'importe quel point particulier dans la séquence de données unifiée. En utilisant le curseur et les signaux d'édition de clavier, parmi lesquels "insertion", "suppression", "remplacement", "déplacement" et "copie", l'auteur peut accomplir toutes ces fonctions d'édition, en les appliquant sans avoir à déterminer si les données sont des données vocales, de texte ou de marques. La présentation faite sur le dispositif de visualisation fait apparaître tous les changements d'édition au fur et à mesure qu'ils sont effectués. En utilisant le curseur et des signaux introduits au clavier, l'auteur peut également reproduire le message vocal avec n'importe quel dispositif audio connecté.

Une description plus détaillée du programme de fonctionnement est faite ci-dessous.

Un programme de fonctionnement d'éditeur de message vocal est enregistré dans la mémoire 22 et, en association avec le processeur 26, il commande le fonctionnement du système.

me dans l'accomplissement de toutes les fonctions de l'éditeur de message vocal. Le programme de l'éditeur de message vocal utilise une file d'attente de programmes, et des sous-programmes appelés par l'éditeur de message vocal sont tout
 5 d'abord introduits dans la file d'attente de programmes, et ils sont exécutés par la suite lorsque le processeur arrive à leur niveau. Avec une telle file d'attente, un programme de gestion d'interruption introduit dans la file un sous-programme destiné au traitement de l'interruption, puis il re-
 10 lide immédiatement les interruptions et retourne au programme principal. Les sous-programmes sont introduits dans la file et sont pris en charge par le processeur lorsque ce dernier est disponible pour le faire. Un module de file d'attente de programmes contient des sous-programmes destinés à manipuler
 15 la file d'attente de programmes de l'éditeur de message vocal. Ces sous-programmes sont les suivants :

INIT\$FILE\$PROG : Initialise la file d'attente de programmes.
 INTR\$FILE\$PROG : Introduit une adresse de procédure et un paramètre d'adresse dans la file d'attente de programmes.
 EXEC\$FILE\$PROG : Vérifie si une paire procédure/paramètre est dans la file. Dans l'affirmative, il appelle la procédure en lui transmettant
 20 le paramètre d'adresse unique.

Le programme principal de l'éditeur de message vocal est très simple, à cause de la file d'attente de programmes de l'éditeur de message vocal. Le programme principal de l'éditeur de message vocal remplit deux fonctions : 1) Il appelle
 25 un programme d'initialisation, INIT\$EDITEUR\$VOCAL, pour initialiser toutes les structures de données et les dispositifs d'entrée/sortie matériels qu'utilise l'éditeur de message vocal ; 2) Il se reboucle ensuite indéfiniment, en appelant EXEC\$FILE\$PROG pour exécuter n'importe quels autres sous-
 30 programmes présents dans la file d'attente de programmes. Si l'utilisateur indique qu'il désire sortir de l'éditeur de message vocal, par exemple, la procédure SORTIE\$EDITEUR est

introduite dans la file d'attente de programmes. Le proces-
seur appelle ce programme dès qu'il le peut, ce qui fait
retourner l'éditeur de message vocal vers l'application qui
l'a appelé.

5 La description précédente permet de voir qu'une
fois que l'éditeur de message vocal a été appelé et a initia-
lisé les variables et le matériel, il se reboucle simplement
en attendant que quelque chose apparaisse dans la file
d'attente de programmes. On utilise des procédures d'inter-
10 ruption pour placer quelque chose dans cette file d'attente.
Des procédures d'interruption sont exécutées lorsque apparaît
une interruption de matériel. Lorsque ceci se produit, le
processeur invalide les interruptions, il introduit dans la
pile le compteur de programme courant et passe à une procédu-
15 re destinée à la gestion de l'interruption.

L'éditeur de message vocal fonctionne dans le mode
d'interruption 2 du Z80, et il reçoit des interruptions pro-
venant des dispositifs suivants, donnés dans l'ordre de prio-
rité des interruptions :

- | | | |
|----|----------------|---|
| 20 | 1. Canal CTC 0 | Compte de Blocs : ce canal produit une interruption lorsque le matériel audio vient juste de terminer l'enregistrement ou la reproduction d'une mémoire tampon d'information audio numérisée. |
| | 2. Canal CTC 1 | Sonnerie du Téléphone : ce canal produit une interruption chaque fois que le téléphone sonne. |
| | 3. Canal CTC 2 | Appui sur une touche : l'éditeur de message vocal programme ce canal de façon à produire |
| 25 | | une interruption chaque fois qu'un appui sur une touche est reçu. |
| | 4. Canal CTC 3 | Rythmeur : l'éditeur de message vocal programme ce canal de façon qu'il produise une interruption toutes les 10 ms (0,010 seconde). |

Les adresses des programmes de gestion d'interrup-
tion pour les dispositifs ci-dessus se trouvent dans une table
30 de vecteurs d'interruption, en mémoire. Lorsque l'un quelcon-
que des dispositifs ci-dessus produit une interruption,
l'adresse correspondante dans la table de vecteurs d'interrup-

tion est appelée.

Les programmes de gestion d'interruption de l'éditeur de message vocal se trouvent dans deux modules : le module d'interruptions et le module de programmes de gestion
5 d'entrée/sortie.

Le module d'interruptions consiste seulement en un groupe de programmes au niveau assembleur, à raison d'un pour chaque dispositif produisant des interruptions. Tous ces programmes sauvegardent les registres dans la pile, ce qu'on
10 appelle une procédure PLM, puis ils rétablissent les registres, valident les interruptions et retournent. Les programmes de gestion sont les suivants :

audio :	Programme de gestion du canal CTC 0, appelle la procédure PLM INTERRUPT\$AUDIO.
sonnerie :	Programme de gestion du canal CTC 1, appelle la procédure INTERRUPT\$SONNERIE.
15 GEST\$TOUCHE :	Programme de gestion du canal CTC 2, accomplit une opération IN (00) pour déterminer la touche qui a été enfoncée, sauvegarde cette information dans une variable TOUCHEBRUTE, appelle la procédure PLM TOUCHE\$OBTENUE.
rythmeur :	Programme de gestion du canal CTC 3, appelle la procédure PLM RYTHMEUR\$DIX\$MS.

Le module de programmes de gestion d'entrée/sortie contient des procédures PLM qui effectuent la majeure partie
20 de la gestion des interruptions. Il contient également quelques autres programmes divers. Les programmes d'interruption sont décrits brièvement ci-dessous :

INTERRUPT\$SONNERIE :	Introduit dans la file d'attente de programmes une procédure qui présentera le message : "Votre téléphone sonne, appuyez sur TAB".
25 TOUCHE\$OBTENUE :	Sa fonction caractéristique consiste simplement à introduire la procédure TRAITEMENT\$TOUCHE dans la file d'attente de programmes. C'est TRAITEMENT\$TOUCHE qui accomplit réellement la gestion de l'appui sur une touche.
RYTHMEUR\$DIX\$MS :	Appelle d'autres procédures PLM qui accomplissent des contrôles périodiques portant sur certaines conditions.

30 Presque toutes les fonctions de l'éditeur de message

vocal sont déclenchées lorsque l'utilisateur appuie sur une touche. L'éditeur de message vocal utilise un système basé sur une table pour déterminer la procédure à appeler en réponse à l'appui sur une touche donnée.

- 5 Les touches du poste de travail sont divisées en 16 catégories différentes. Un nombre de 0 à 15 est affecté à chaque catégorie. Aucune touche ne peut apparaître dans plus d'une catégorie. Les numéros de catégorie et les touches de chaque catégorie sont indiqués ci-après.

10	<u>Numéro de catégorie</u>	<u>Description</u>	<u>Touches</u>
	1	catégorie "touche d'enregistrement"	INSERTION
	2	catégorie "touche d'arrêt"	STOP
	3	catégorie "touche de reproduction/arrêt"	Barre d'Espacement, (POSITION INITIALE)
	4	catégorie "curseur"	Curseur Nord, Curseur Est, Curseur Sud, Curseur Ouest
15	5	catégorie "passage"	PASSAGE A LA PAGE SUIVAN- TE
	6	catégorie "nombre"	0 à 9
	7	catégorie "texte"	A - Z, a - z, virgule, point ! # \$ % & * () - = +] [; : ' " / ? %
	8	catégorie "espacement arrière"	Touche d'Espacement Arrière
20	9	catégorie "marque"	RETOUR, NOTE
	10	catégorie "renuméro- tage"	
	11	catégorie "édition"	SUPPRESSION, REPRODUCTION, DEPLACEMENT, COPIE
	12	catégorie "exécution"	EXECUTION
	13	catégorie "annulation"	ANNULATION
	14	catégorie "touche d'aide"	ORDRE, (AIDE)
	15	catégorie "touche de téléphone"	TAB
25	0	catégorie "touche invalide"	Toutes les autres tou- ches

Il existe une table de traduction qui convertit les codes de touche de matériel bruts en numéros de catégorie correspondants (0 - 15). Cette table se trouve dans le secteur zéro du

fichier "TABLCAT-VOC". Le secteur un de ce fichier contient la table de traduction pré-WISCII standard pour l'appui sur les touches. Il est important de noter que la table de catégorie est indépendante du positionnement de clavier. Par exemple, les touches ANNULATION et POSITIONNEMENT, ANNULATION, sont toutes deux dans la catégorie "ANNULATION" (13). Ceci n'affecte cependant pas les caractères de texte en majuscules et en minuscules, du fait qu'ils sont tous dans la catégorie "texte" (7).

10 L'éditeur est divisé en différents états de fonctionnement. Les touches peuvent avoir des significations différentes en fonction de la valeur de l'état courant, ce qui fait qu'une table de procédure est définie pour chaque état. Ces tables de procédure sont appelées des tables
15 d'état. Les tables d'état sont définies dans le module de tables d'état.

Les tables d'état de l'éditeur de message vocal contiennent des index qui désignent des rubriques d'une grande table de procédures. On peut trouver cette table dans le
20 module de tables de programme, contenant 36 rubriques.

Au passage initial à l'éditeur, l'état principal est l'état de fonctionnement courant. Lorsque de nouveaux états de fonctionnement sont mis en fonction, les anciens états, accompagnés d'un index du message guide-opérateur courant sur l'écran, sont introduits dans une pile d'états. On
25 supposera que l'utilisateur appuie sur la touche SUPPRESSION, alors qu'il est dans l'état principal. L'état principal est introduit dans la pile d'états et l'état de définition de segment devient maintenant l'état courant. Le message guide-
30 opérateur "Que faut-il supprimer ?" apparaît sur l'écran.

On supposera maintenant que l'utilisateur appuie sur la touche PASSAGE A LA PAGE SUIVANTE. L'état de définition de segment est introduit dans la pile, et le message guide-opérateur est également introduit dans la pile d'état.
35 Le nouvel état est l'état de passage. Le message guide-opéra-

teur "Où faut-il aller" apparaît sur l'écran. L'utilisateur frappe un nombre et appuie sur EXECUTION. Ceci exécute une procédure de passage au nombre introduit.

A ce point, l'état de définition de segment et le
5 message guide-opérateur sont extraits de la pile. Le message
"Que faut-il supprimer ?" est à nouveau présenté sur l'écran.
L'utilisateur frappe EXECUTION, et une procédure est appelée
pour supprimer la partie concernée du fichier de message
vocal. L'état principal est ensuite extrait de la pile, et on
10 est ramené à l'état de fonctionnement d'origine.

En plus des tables d'état elles-mêmes, le module de
tables d'état contient également des procédures pour manipu-
ler la pile d'état. Ces procédures sont les suivantes :

- INIT\$ETAT : Initialise la pile d'état.
- NOUV\$ETAT : Introduit l'état ancien dans la pile, prend
l'état spécifié comme état courant.
- 15 EXTR\$ETAT : Extrait un état de la pile, prend cet état comme
état courant.

Le module de table d'état contient également un programme qui,
à partir d'un numéro de catégorie, retourne l'adresse de la
procédure qui correspond à cette catégorie pour l'état cou-
rant :

- 20 ADR\$PROGRAMME : Une catégorie étant donnée, cette procédure
recherche dans la table de l'état courant
l'adresse de la procédure qui correspond à
cette catégorie.

La décision d'appeler une procédure particulière
peut ainsi être résumée de la façon suivante :

- 1) Interruption d'appui sur une touche.
- 25 2) GEST\$TOUCHE sauvegarde les registres, place le code de
touche de matériel dans la variable TOUCHEBRUTE, appelle
TOUCHE\$OBTENUE.
- 3) TOUCHEBRUTE accomplit les actions suivantes :
 - a) si une erreur irrécupérable s'est produite,
passage en sortie.
 - b) si on a frappé POSITIONNEMENT\$PAGE, accomplisse-
ment d'un vidage.
 - c) si la touche précédente n'a pas encore été trai-
tée, la touche courante est ignorée.
 - 30 d) introduction de l'adresse de la procédure

TRAITEMENT\$TOUCHE ainsi que du paramètre
TOUCHRBRUTE dans la file d'attente de programmes.

- 4) TRAITEMENT\$TOUCHE est extraite de la file d'attente de programmes et exécutée, ce qui accomplit les actions suivantes :
- a) traduction de l'appui sur une touche, en utilisant la table de traduction.
 - b) détermination du numéro de catégorie pour cette touche, en utilisant la table de catégories.
 - 5 c) si le bit de fort poids du numéro de catégorie est zéro, prise en charge de cet appui sur une touche.
 - d) effacement de tout message d'erreur.
 - e) arrêt des fonctions audio, à l'exception de RETOUR et de la catégorie "reproduction/arrêt".
 - f) appel de ADR\$PROGRAMME, et transmission de cette procédure à la catégorie, pour obtenir l'adresse de la procédure à laquelle on doit passer.
 - 10 g) introduction dans la file d'attente de programmes de cette adresse de procédure et de l'appui sur une touche traduit.
- 5) Le programme approprié ainsi que l'appui sur une touche traduit sont extraits de la file d'attente de programmes et exécutés.

Des procédures supplémentaires peuvent être divisées grossièrement en deux parties. Il y a pour chaque structure de données des modules de bas niveau qui effectuent des opérations sur cette structure. L'index de fichier (index audio, table de marques, table de notes), des fonctions audio et l'écran constituent des exemples représentatifs de modules de bas niveau.

20 La seconde partie est constituée par les programmes de haut niveau. Ces procédures sont appelées de façon caractéristique par le système de traitement des appuis sur les touches (leurs adresses sont dans la table de programmes), et elles peuvent elles-mêmes appeler les programmes de niveau

25 inférieur qui accomplissent la majeure partie du travail. On peut donc les considérer comme une interface entre les procédures de gestion des appuis sur les touches et les procédures de bas niveau effectuant le gros du travail.

Le module d'interface d'utilisateur (V:voix.rrr.plm. 30ve.intutilisateur) contient des procédures de haut niveau pour

des fonctions audio, de marquage de section et de renumérotage :

- 5 REPROD\$ARRET : Procédure appelée chaque fois qu'on appuie sur une touche dans la catégorie "reproduction/arrêt". Si la fonction audio est arrêtée au moment considéré, cette procédure amène le curseur au début du secteur audio suivant et commence la reproduction. Si la fonction audio est en cours de reproduction ou d'enregistrement, elle l'arrête.
- 10 INSERTION\$MARQUE : Procédure appelée lorsqu'on appuie sur une touche dans la catégorie "marque". Si on a introduit une marque de paragraphe, elle indique sa position exacte sur l'écran et appelle le programme de module de fenêtre approprié pour passer à ce programme. Si on a appuyé sur la touche NOTE, la procédure détermine si le curseur est sur une note au moment considéré. Dans la négative, elle en crée une. Dans un cas comme dans l'autre, on passe au mode de texte.
- 15 RENUMEROTAGE : Procédure appelée lorsqu'on appuie sur une touche dans la catégorie "renumérotage". L'éditeur est placé dans l'état de renumérotage et le message guide-opérateur "Renumérotage des marques ?" est présenté.
- REN\$EXECUTION : Procédure appelée lorsqu'on appuie sur EXECUTION alors qu'on est dans l'état de renumérotage. Elle appelle une procédure de table de marques pour renuméroter les marques, pour renouveler l'information présentée sur l'écran, et pour extraire de la pile l'état précédent.
- 20 REN\$ANNULATION : Procédure appelée lorsqu'on appuie sur CANCEL alors qu'on est dans l'état de renumérotage. Elle extrait de la pile l'état précédent.

Le module d'espacement arrière met en oeuvre la fonction d'espacement arrière. Le fait d'appuyer sur la touche d'espacement arrière fait reculer le curseur de cinq 25 secondes et provoque la reproduction pendant cinq secondes. Si on appuie N fois, le curseur recule de $N * 5$ secondes et la reproduction a lieu pendant la même durée. Pendant la reproduction, l'appui sur n'importe quelle touche autre que

la touche d'espacement arrière arrête la reproduction, ce qui annule complètement la fonction d'espacement arrière.

Lorsqu'on appuie sur la touche d'espacement arrière, il s'écoule une durée de 350 millisecondes avant le début de la reproduction. Ceci a pour but de laisser à l'utilisateur le temps d'appuyer de façon répétée sur la touche d'espacement arrière avant que la reproduction ne commence. Le module d'espacement arrière utilise trois variables pour accomplir ces fonctions :

- 10 mode\$ea : Etat "1" si on effectue un espacement arrière, "0" dans le cas contraire.
- temps\$ea : Le temps correspondant au curseur lorsque l'utilisateur a appuyé pour la première fois sur ESPACEMENT ARRIERE. La reproduction aura lieu jusqu'à cette position, mais pas au-delà, indépendamment du nombre d'appuis sur cette touche.
- compt\$reprod\$ea : Compteur décrémenté par la procédure RYTHMEUR\$DIX\$MS. Utilisé pour compter la durée d'attente de 350 ms.

15 La fonction d'espacement arrière fournit les procédures suivantes :

- EA : Procédure appelée lorsqu'on appuie sur la touche d'espacement arrière. Si l'appui est le premier, elle positionne mode\$ea à "1" et rappelle temps\$ea. Elle initialise temps\$attente\$ea à 350 ms.
- 20 COMPTEUR\$ATTENTE\$EA : Procédure appelée toutes les 10 ms par RYTHMEUR\$DIX\$MS. Cette procédure décrémente temps\$attente\$ea, et lorsque 350 ms se sont écoulées, elle introduit dans la file d'attente de programmes une procédure qui provoquera la reproduction à partir de la position de curseur courante jusqu'à temps\$ea.
- 25 CONTR\$TOUCHE\$EA : Procédure appelée par TRAITEMENT\$TOUCHE. Cette procédure annule le mode d'espacement arrière si on appuie sur une touche autre que la touche d'espacement arrière.

Le module de curseur comporte toutes les fonctions de curseur de haut niveau. Ici encore, ces procédures sont simplement des interfaces entre les programmes de traitement

de touche et de l'écran qui déplacent en réalité le curseur sur l'écran.

- 5 **PROG\$CURSEUR :** Procédure appelée dans la plupart des états lorsqu'on appuie sur une touche dans la catégorie "curseur". Elle appelle simplement un programme d'écran parmi quatre, en fonction de la touche de curseur sur laquelle on a appuyé.
- 10 **PROG\$PASSAGE :** Procédure appelée lorsqu'on appuie sur la touche PASSAGE A LA PAGE SUIVANTE. Elle introduit l'ancien état dans la pile et fait en sorte que l'état courant soit l'état "Passage". Elle présente le message guide-opérateur "Où faut-il aller ?" et déplace le curseur jusqu'à la position suivant immédiatement le message guide-opérateur. On notera qu'au moment de la traduction du fichier de message, ce message guide-opérateur doit être cadré à droite.
- 15 **PASSAGE\$SORTIE :** Cette procédure est appelée lorsqu'on appuie sur ANNULATION alors qu'on est dans l'état "Passage". Elle ramène le curseur dans la partie audio/marque de l'écran et extrait de la pile l'état précédent.
- 20 **PASSAGE\$CURSEUR :** Procédure appelée lorsqu'on appuie sur l'une des touches de curseur alors qu'on est dans l'état "Passage". Elle appelle un programme d'écran parmi quatre en fonction de la touche de curseur sur laquelle on a appuyé. Elle appelle ensuite PASSAGE\$SORTIE pour retourner à l'état précédent.
- 20 **PASSAGE\$ACCEPT\$NOMBRE :** Procédure appelée lorsqu'on appuie sur une touche dans la catégorie "nombre", alors qu'on est dans l'état "Passage". Cette procédure affiche le nombre sur l'écran juste après le message guide-opérateur et met à jour la position du curseur.
- 25 **PASSAGE\$EXECUTION :** Procédure appelée lorsqu'on appuie sur EXECUTION alors qu'on est dans l'état "Passage". Si un nombre est présent sur l'écran, il est converti de ASCII en binaire et un programme d'écran est appelé pour positionner le curseur au-dessous de la marque appropriée. Elle appelle ensuite PASSAGE\$SORTIE

pour retourner à l'état précédent.

Le module d'entrée de texte contient des programmes destinés à l'introduction de notes de texte, dans le mode de texte. On utilise les variables suivantes :

5	tampon\$texte (60)	tampon destiné à conserver la note de texte pendant son introduction.
	indext	position courante (0-59) dans le tampon de texte.
	curseurt	position courante du curseur sur l'écran.
	index\$note	index désignant la table de note de la note de texte sur laquelle on travaille au moment considéré.
10	premier	indicateur, à l'état "1" si la note qui a été introduite vient juste d'être créée. Dans ce cas, l'appui sur ANNULATION supprimera cette note. S'il s'agit d'une note ancienne qui est modifiée, l'appui sur ANNULATION rétablira simplement la note à sa forme d'origine.

Les programmes suivants sont fournis :

15	TEXT\$FIXE\$PREMIER :	Procédure appelée par INSERT\$MARQUE pour indiquer au module d'entrée suivant que cette note vient juste d'être introduite.
	ENTREE\$MODE\$TEXTE :	Procédure appelée par INSERT\$MARQUE lorsqu'on appuie sur la touche NOTE. Introduit l'ancien état dans la pile et fixe le nouvel état "Texte". Fait apparaître le message guide-opérateur "Entrez le texte". Prélève la note dans la table de note et la place dans le tampon de texte.
20	ANNUL\$TEXTE :	Procédure appelée lorsqu'on appuie sur ANNULATION alors qu'on est dans l'état "Texte". Si on a introduit une nouvelle note, cette note est supprimée. Dans le cas contraire, le tampon de texte est rejeté et l'écran présente à nouveau la note ancienne inchangée. Rétablit l'état précédent.
25	EXECUT\$TEXTE :	Procédure appelée lorsqu'on appuie sur EXECUTION alors qu'on est dans l'état "Texte". Remplace l'ancienne note par le contenu du tampon de texte. Rétablit l'état précédent.

- CURSEUR\$TEXTE :** Procédure appelée lorsqu'on appuie sur une touche de curseur alors qu'on est dans l'état "Texte". Déplace le curseur en avant ou en arrière. Affiche un message d'erreur si on appuie sur Curseur Nord ou Curseur Sud.
- 5 ESPACE\$ARRIERE\$TEXTE :** Procédure appelée lorsqu'on appuie sur la touche d'espacement arrière alors qu'on est dans l'état "Texte". Recule le curseur d'une position, puis efface le caractère sous lequel se trouve le curseur.
- ENTREE\$TEXTE :** Procédure appelée lorsqu'on appuie sur une touche dans les catégories "texte", "nombre" ou "reproduction/arrêt". Introduit le caractère dans le tampon de texte et sur l'écran et fait avancer le curseur d'une position.
- 10 TEXTE :** Procédure appelée lorsqu'on appuie sur une touche de texte alors qu'on est dans l'état "Principal". Si le curseur est sur une note, cette procédure fait passer au mode de texte et introduit l'information de la touche enfoncée dans le tampon de texte et sur l'écran. Si le curseur n'est pas sur une note, cette procédure présente le message "Déplacez le curseur".
- 15** Le module d'édition établit une interface entre le système de traitement de touche et les programmes d'index de fichier de bas niveau, concernant l'écran, qui accomplissent réellement les manipulations sur le fichier.
- Le module d'édition conserve la trace des parties
- 20** du fichier qui sont éditées. On utilise une structure de points pour désigner des positions dans le fichier. Cette structure est de la forme :
- structure de points (
- | | | |
|--|-------|------------|
| | temps | adresse, |
| | index | multiplet) |
- 25** dans laquelle : "temps" est le temps écoulé dans le fichier, et "index" est l'index de marque de la marque courante, ou de la marque suivante dans le fichier s'il n'y a pas de marque à cette position.

On utilise les structures de points suivantes pour conserver la trace de positions pendant l'édition :

pointdébut	le début d'un segment à supprimer/déplacer/ copier
pointfin	la fin d'un segment à supprimer/déplacer/ copier
5 pointdest	le point de destination pour une opération de déplacement/copie.

Pour supprimer une partie du fichier, on retire du fichier le segment compris entre pointdébut et pointfin (inclus).

Pour déplacer ou copier une partie du fichier, on
10 déplace ou on copie le segment compris entre pointdébut et pointfin (inclus), pour l'amener à pointdest.

Lorsqu'on effectue une insertion dans le fichier, pointdest devient le point d'insertion. La fin de fichier courante est pointdébut, et l'enregistrement commence à la
15 fin du fichier.

Lorsque l'utilisateur appuie sur STOP, le programme effectue un déplacement, comme décrit ci-dessus, en déplaçant vers pointdest le segment qui est délimité par (pointdébut, pointfin).

20 Pour remplacer un segment du fichier, on utilise trois structures de points supplémentaires :

rpointdébut	contient le début du segment à supprimer.
rpointfin	contient la fin du segment à supprimer.
rpointdest	contient le début du segment à insérer.

La procédure de remplacement se déroule de la façon suivante : on définit initialement le segment à rem-
25 placer entre pointdébut et pointfin. Une fois que le segment est défini, on copie pointdébut vers rpointdest, et pointfin vers rpointfin, et on fixe rpointdébut à la fin du fichier. On accomplit ensuite la procédure d'insertion classique, en enregistrant à la fin du fichier. Comme avec l'insertion,
30 lorsqu'on appuie sur STOP, la nouvelle matière, c'est-à-dire le segment (pointdébut, pointfin) est déplacée vers le point d'insertion, c'est-à-dire pointdest, pour achever l'insertion.

Pendant le remplacement, l'utilisateur peut insérer, reproduire, actionner les touches de curseur et introduire des marques de paragraphe et des notes de texte. Toutes les insertions sont accomplies de la manière normale, en utilisant pointdébut, pointfin et pointdest. Bien entendu, toutes les insertions sont limitées à la région située au-delà de rpointdébut.

Si l'utilisateur appuie sur ANNULATION, le remplacement est annulé en replaçant à rpointdébut l'instant final du fichier vocal, ce qui rétablit le fichier à sa forme d'origine.

Si l'utilisateur appuie sur EXECUTION, le remplacement est exécuté en supprimant tout d'abord le segment (rpointdest, rpointfin), puis en assignant rpointdest à pointdest, et la fin de fichier à pointfin, puis en effectuant l'insertion en utilisant un déplacement normal du segment (pointdébut, pointfin) vers pointdest.

Le module de fonctions audio contient des programmes destinés à la reproduction et à l'enregistrement dans des fichiers vocaux. Il utilise un module associé, le module d'entrée/sortie, qui contient des structures de données et des procédures pour manipuler les tampons et les demandes de file d'attente dirigées vers le module maître.

Pendant la reproduction ou l'enregistrement, des données audio doivent être enregistrées en tampon de façon que la reproduction ou l'enregistrement ne soit pas interrompu par l'attente de l'achèvement d'une opération d'écriture ou de lecture dans un tampon. Le logiciel de poste de travail audio est conçu de façon à utiliser au moins deux tampons, mais on peut en utiliser davantage si l'espace le permet. A l'heure actuelle, le poste de travail audio utilise 6 tampons audio.

L'éditeur de message vocal utilise des tampons qui ont de 1 à 16 secteurs de longueur. Ces tampons sont alignés par page en mémoire. Chaque tampon correspond à un bloc audio

dans le fichier vocal. Le module d'entrée/sortie contient des structures, appelées structures d'information, qui gèrent les tampons audio. Le module d'entrée/sortie contient une file d'attente de demandes d'entrée/sortie, qu'on utilise pour organiser les blocs (RCB) en file d'attente. Le rythmeur de 10 ms contrôle cette file d'attente toutes les 10 ms. Si elle contient quelque chose, la procédure du rythmeur extrait elle-même la demande de la file d'attente et la présente au module maître.

10 La file d'attente de demandes d'entrée/sortie utilise les structures de données suivantes :

file d'attente	un tableau d'adresses, constituant la file d'attente de demandes d'entrée/sortie.
haut	index du haut de la file d'attente.
bas	index du bas de la file d'attente.
compte	nombre d'éléments dans la file d'attente.

15 Les programmes suivants manipulent la file d'attente.

INTROD\$ES : Introduit l'adresse d'un RCB dans la file d'attente de demandes d'entrée/sortie.

20 EXTRACT\$ET\$EMISSION : S'il y a quelque chose dans la file d'attente et si le SCA est libre, cette procédure extrait l'adresse de RCB de la file d'attente et la place dans le SCA. Cette procédure est appelée dès qu'on place quelque chose dans la file d'attente (elle tente de l'extraire immédiatement). Elle est également appelée toutes les 10 ms par la procédure RYTHMEUR\$DIX\$MS.

25 Du fait que l'éditeur de message vocal n'insère que des données enregistrées, il n'effectue pas d'enregistrement en superposition, et l'enregistrement commence toujours à la fin du fichier. Les données insérées sont enregistrées à la fin du fichier et sont ensuite déplacées vers le point d'insertion.

30 Pour enregistrer, on accomplit les opérations suivantes :

1) partir avec la sixième structure d'information.

- a) remplir la première adresse de tampon
 - b) remplir la taille de tampon
 - c) si on enregistre dans le dernier bloc du fichier, positionner l'indicateur d'arrêt.
- 2) indiquer au matériel l'adresse du premier tampon.
- 3) indiquer au matériel de commencer l'enregistrement.
- 5 4) accomplir la procédure suivante :
- a) indiquer au matériel la taille du tampon dans lequel il enregistre au moment considéré.
 - b) introduire en file d'attente une demande d'écriture pour le tampon précédent, s'il ne s'agit pas du premier tampon.
 - c) si l'indicateur d'arrêt est positionné pour ce tampon, arrêter.
 - d) déterminer si une demande d'écriture antérieure pour ce tampon a été satisfaite ; dans la négative, arrêter la fonction audio jusqu'à ce que la demande ait été satisfaite.
 - e) remplir le RCB pour ce tampon.
 - f) incrémenter les variables de façon à être prêt à traiter le tampon suivant.
- 10

Une fois que le matériel a achevé l'enregistrement dans le premier tampon, une interruption de compte de bloc est produite (canal CTC 0). Lorsque ceci se produit, la procédure INTERRUPT\$AUDIO est appelée. Cette procédure effectue un contrôle pour voir si le mode de reproduction ou d'enregistrement est en fonction, et elle appelle une procédure d'interruption de reproduction ou d'enregistrement. L'opération 4), ci-dessus, est la procédure d'interruption d'enregistrement INTERRUPT\$ENREG. Pendant que l'enregistrement progresse, elle est appelée chaque fois qu'un tampon est terminé.

20

La reproduction est similaire à l'enregistrement. On effectue quelques opérations d'initialisation, puis on indique au matériel de commencer la reproduction. On appelle immédiatement le programme INTERRUPT\$REPROD. Pendant la reproduction de chaque tampon, on appelle à nouveau INTERRUPT\$REPROD pour préparer le tampon suivant pour la reproduction, et pour mettre en file d'attente une demande de lecture d'un autre tampon, à partir du disque.

25

30

Pendant l'enregistrement, la cadence d'échantillon-

nage est toujours fixée à la valeur correspondant au littéral CAD\$ECHANT, qui définit la cadence d'échantillonnage.

Cependant, pendant la reproduction, on peut changer la cadence d'échantillonnage. Toutes les 10 ms, la procédure

5. RYTHMEUR\$DIX\$MS appelle la procédure FIXATION\$CADENCE. Cette procédure appelle un programme destiné à convertir le réglage courant de la commande de vitesse pour donner la cadence d'échantillonnage appropriée. La valeur de cette cadence d'échantillonnage est ensuite transmise au matériel.

- 10 L'écran de l'éditeur de message vocal est divisé en deux sections : la section d'état et la section audio/marques. La section d'état est formée par les deux premières lignes et par la dernière ligne de l'écran. On utilise cette zone pour afficher des messages guide-opérateur, le temps
- 15 correspondant au curseur, la longueur, etc. On utilise la section audio/marques, qui est formée par les lignes 3 à 21, pour présenter le contenu du fichier vocal, c'est-à-dire les blocs audio, les notes de texte et les marques de paragraphe.

- 20 Le module de visualisation commande la section d'état de l'écran. Ce module contient en outre toutes les procédures GROUPEMENU. Il contient des procédures pour initialiser GROUPEMENU, et pour visualiser le temps correspondant au curseur, le mode audio, un aide-mémoire, le mode de
- 25 téléphone, le titre, les messages guide-opérateur, la longueur et les messages d'erreur.

- Le module de fenêtre contient les programmes destinés à la visualisation et à la mise à jour de la partie audio/marques de l'écran. Ce module est assisté par les modules
- 30 les suivants :

conversion	(V:voix.rrr.plm.ve.conversion)	Programmes de conversion de la structure de position
temps	(V:voix.rrr.plm.ve.temps)	Programmes de conversion temps-position

	23		
ligne	(V:voix.rrr.plm.ve.ligne)	Mise en oeuvre de la structure de lignes	
région	(V:voix.rrr.plm.ve.région)	Chercheur d'index d'édition	
défilement	(V:voix.rrr.plm.ve.défilement)	Manipulations de fenêtre de bas niveau	
5	Le fichier vocal comprend un en-tête, une table de marques, une table de notes, un plan d'implantation des secteurs et un plan d'implantation des blocs. Les modules suivants contiennent un programme permettant d'accéder au fichier vocal :		
10	indexfichier	(V:voix.rrr.plm.ve.indexfichier)	Réalisation de l'index de fichier
	éditindex	(V:voix.rrr.plm.ve.éditindex)	Opérations d'édition sur l'index de fichier
	marque	(V:voix.rrr.plm.ve.marque)	Réalisation de la table de marques
	note	(V:voix.rrr.plm.ve.note)	Réalisation de la table de notes
15	voixgrm	(V:voix.rrr.plm.ve.voixgrm)	Programmes de création, d'initialisation et de mise en oeuvre du fichier vocal
	extension	(V:voix.rrr.plm.ve.extension)	Programmes d'extension et de troncature du fichier vocal
20	irrécup	Erreur irrécupérable, programme de gestion ABEND	

Le Module d'Erreur contient des procédures pour les programmes ABEND, concernant les erreurs irrécupérables et récupérables. Un indicateur, INDVIDAGE, fixé dans le groupe, 25 est utilisé pour déterminer si une erreur conduit ou non à un vidage. Si INDVIDAGE est OFFh, les vidages sont validés. Si sa valeur est 0, les vidages sont invalidés.

Les procédures fournies sont les suivantes :

ERREUR\$RECUP : Vidage si l'indicateur est positionné, affi-

chage de l'erreur VE: XXX, en désignant par XXX un numéro d'erreur transmis. Ces numéros d'erreur sont définis dans (V:voix.rrr.lit.ve.ERR). Il y a également affichage d'une section de données à 16 multiplats (de façon caractéristique un RCB) en cas de transmission d'un paramètre.

- 5 INFORM\$ERREUR : Affichage d'un message d'erreur non VE, retour à l'application appelante, après appui sur une touche quelconque. Les messages d'erreur non VE sont simplement les erreurs normales telles que "Déplacer le curseur" qui sont affichées dans la partie inférieure de l'écran. Ces messages d'erreur sont définis dans (V:voix.rrr.lit.ve.MERREUR).
- 10 ERREUR\$IRRECUP : Identique à ERREUR\$RECUP, à l'exception du fait que cette erreur n'est pas récupérable. L'éditeur retourne au programme appelant lorsque l'utilisateur appuie sur une touche quelconque.

Le système de récupération de l'éditeur de message vocal permet la récupération à partir de coupures d'alimentation du poste de travail ou d'opérations accidentelles de chargement de programme initial (IPL) pendant le processus d'enregistrement. L'éditeur de message vocal utilise certaines structures de données courantes, et trois modules contiennent ces structures et des programmes pour les manipuler.

La file d'attente de programmes utilise les procédures suivantes :

- 20 INIT\$QUEUE : Cette procédure définit une file d'attente. L'utilisateur spécifie l'adresse de la file, la taille de la file, la taille de chaque élément dans la file et un pointeur désignant une structure qui contient toutes les caractéristiques importantes de la file. Cette structure identifie la file d'attente. Elle doit être transmise sous la forme d'un paramètre aux programmes d'introduction et d'extraction qui sont décrits ci-dessous.
- 25 INTR\$QUEUE : Cette procédure introduit un élément dans une file d'attente spécifiée.
- EXTR\$QUEUE : Cette procédure extrait un élément de la tête d'une file d'attente spécifiée.

Le module de pile (V:voix.rrr.plm.ve.pile) est une réalisation d'une pile avec des programmes d'introduction et

d'extraction. La pile de module de table d'état utilise des procédures venant du module de pile pour réaliser la pile d'état. Contrairement au module de file d'attente, les programmes du module de pile ne peuvent fonctionner que sur une
 5 seule pile, définie dans le module de la façon suivante :

pile (12)	espace (en multiplats) réservé pour la pile.
pp	pointeur de pile.

Deux programmes manipulent la pile :

INTRODUCTION : Introduit un élément dans la pile.

EXTRACTION : Extrait un élément de la pile.

Le module de table de bits (V:voix.rrr.plm.ve.bit)
 10 peut positionner, restaurer et tester des bits dans une table de bits spécifiée par l'utilisateur. La table ne peut pas contenir plus de 256 multiplats. La table de marques utilise une table de bits pour déterminer le numéro de la marque de paragraphe suivante à créer. Le module d'édition d'index
 15 de fichier utilise une table de bits pour ordonner tous les blocs libres dans l'index, de façon que les extensions de fichier soient accomplies d'une manière optimale. Le module de table de bits contient les procédures suivantes :

·POS\$BIT :	Positionne un bit dans une table de bits.
·REST\$BIT :	Restaure un bit dans une table de bits.
20 TEST\$BIT :	Teste un bit pour voir s'il est positionné ou restauré.

Toutes les instructions PLM d'entrée et de sortie pour l'éditeur de message vocal sont contenues dans le module de commande de matériel audio (V:voix.rrr.plm.ve.comaudio). Ce module contient de petites procédures qui font
 25 fonction d'interface entre le matériel et la partie principale du code PLM de l'éditeur de message vocal.

Le module de mode d'interruption fixé (V:voix.rrr.z80.ve.modintfixé) contient deux procédures, l'une pour placer le poste de travail dans le mode d'interruption 2 et l'autre pour le ramener au mode d'interruption
 30 0. Les programmes PLM, INIT\$POSTETRAVAIL et RESTAUR\$POSTETRAVAIL, qui se trouvent dans le module de com-

mande de matériel audio, appellent les deux programmes situés dans le module de mode d'interruption fixé. Les tout premiers multiplets de ce module contiennent les tables de vecteurs d'interruption pour le CTC et le PIO (programme de gestion
5 des entrées/sorties pour les périphériques). Ces tables doivent résider en mémoire dans des limites correspondant à un facteur de huit, et il faut donc prendre soin de vérifier que ceci est réalisé dans la table de liens.

Il va de soi que de nombreuses modifications
10 peuvent être apportées au dispositif décrit et représenté, sans sortir du cadre de l'invention.

REVENDICATIONS

1. Système de traitement d'information, caractérisé en ce qu'il comprend : des moyens d'acquisition de signal continu (54) destinés à acquérir un signal électrique variant
5 de façon continue qui correspond à un message vocal ; des moyens de numérisation (14) destinés à numériser le signal électrique à variation continue, pour produire des données vocales discrètes correspondant à la qualité audible du message vocal ; des moyens d'acquisition de données discrètes
10 (18) destinés à acquérir des données discrètes correspondant à des caractères alphanumériques ; des moyens d'acquisition de signaux discrets (20) destinés à acquérir des signaux discrets comprenant des ordres d'édition et de commande ; une mémoire (22) destinée à l'enregistrement de données sous forme
15 discrète ; des moyens de visualisation (31) destinés à créer une visualisation ; et un processeur (26) ; les moyens d'acquisition de signal continu (54), les moyens de numérisation (14) , les moyens d'acquisition de données discrètes (18), les moyens d'acquisition de signaux discrets (20), la
20 mémoire (22), les moyens de visualisation (31) et le processeur (26) étant interconnectés fonctionnellement par des conducteurs de commande et des canaux de transfert de données (58, 60, 62) ; et en ce qu'un programme de fonctionnement pour le processeur est enregistré dans la mémoire (22) de
25 façon que le processeur (26) commande le fonctionnement du système pour : enregistrer les données vocales discrètes dans la mémoire (22), simultanément à l'acquisition du message vocal ; enregistrer les données de caractères dans la mémoire (22) simultanément à l'entrée de caractères, établir dans la
30 mémoire (22) un enregistrement de séquence indiquant un ordre unifié de message vocal et de données de caractères, visualiser une séquence de marques de symboles vocaux et de marques de caractères, chaque marque de symbole représentant un incrément prédéterminé du message vocal acquis et chaque marque de

caractère correspondant à l'un des caractères entrés, la séquence visualisée correspondant à la séquence dans l'enregistrement, et réviser l'enregistrement de séquence, sous la dépendance d'ordres d'édition entrés, pour tenir compte de
5 changements d'édition dans l'ordre des données vocales et de caractères.

2. Système selon la revendication 1, caractérisé en ce que le programme de fonctionnement est conçu de façon que le processeur (26) commande en outre le fonctionnement du
10 système pour : réagir à des signaux discrets prédéterminés acquis simultanément à l'acquisition du message vocal, pour indiquer dans l'enregistrement de séquence l'instant auquel chacun des signaux discrets prédéterminés a été acquis, et présenter dans la visualisation une indication distinctive de
15 l'instant d'acquisition de chaque signal acquis simultanément, par rapport à d'autres éléments des données vocales.

3. Système selon l'une quelconque des revendications 1 et 2, caractérisé en ce que le programme de fonctionnement est conçu de façon que le processeur (26) commande en
20 outre le fonctionnement du système pour : établir dans la mémoire (22) un pointeur définissant une position de pointeur dans la séquence de données, présenter sur la visualisation une marque visible correspondant à la position du pointeur, et déplacer la position du pointeur qui est définie dans la
25 séquence, et déplacer cette position de façon correspondante dans la visualisation, sous la dépendance de signaux d'entrée acquis.

4. Système selon l'une quelconque des revendications 1 à 3, caractérisé en ce que le programme de fonctionnement est conçu de façon que le processeur (26) commande en
30 outre le fonctionnement du système pour produire, sous la dépendance de signaux d'entrée acquis, un signal audio variant de façon continue qui correspond aux données vocales discrètes enregistrées dans la mémoire (22), la production de
35 ce signal commençant à un point dans la séquence de données

vocales qui correspond à la position de pointeur définie, telle qu'elle est définie au moment considéré, et suivant l'ordre qui est défini au moment considéré dans l'enregistrement de séquence.

5 5. Système selon l'une quelconque des revendications 1 à 4, caractérisé en ce que le programme de fonctionnement est conçu de façon que le processeur (26) commande en outre le fonctionnement du système pour faire avancer le pointeur dans les données de message vocal, en correspondance avec la progression de la production du signal audio.

10 6. Système selon l'une quelconque des revendications 1 à 5, caractérisé en ce qu'il comprend un circuit (28) destiné à détecter une activité d'acquisition de signal audio et à interrompre l'enregistrement des données de message vocal dans la mémoire (22) en l'absence d'activité.

