

Denis BONOMO

Eddy DUTERTRE

COMMUNIQUEZ AVEC VOTRE AMSTRAD CPC 464 - 664 - 6128



SORACOM
éditions
INFORMATIQUE

Denis BONOMO

Eddy DUTERTRE

**COMMUNIQUEZ
AVEC VOTRE AMSTRAD
CPC 464 - 664 - 6128**

SORACOM
editions
INFORMATIQUE

CONCEPTION DU MATERIEL

Présentation du CPC.....	12
Architecture interne du CPC.....	14
Le contrôleur d'écran 6845.....	16
Le Gate Array.....	18
Le PPI 8255.....	19
Le générateur sonore AY-3-8912.....	21
Le microprocesseur Z80.....	22
Un 8ème bit pour l'imprimante.....	24

CONCEPTION DU LOGICIEL

Notions de langage et programmation.....	26
Routines et variables système utiles.....	31

LES PROGRAMMES

Un désassembleur.....	37
Le Locator.....	43
Conversion de coordonnées en Locator.....	48
Suivi de Contest.....	51
Création d'une mire TV.....	65
Emission-Réception en Morse.....	71
Un moniteur de Morse.....	85
Emission-Réception en SSTV.....	93
Emission-Réception en RTTY.....	113
Réglage du démodulateur RTTY.....	137
Utilisation d'une carte 'entrée' en RTTY.....	139
Réception Fac-similé (FAX).....	143

LES INTERFACES

Carte FAX et SSTV.....	176
Démodulateurs CW et RTTY.....	178
Adaptation des cartes 'MAGECO'.....	183

CONCLUSION.....	191
-----------------	-----

ANNEXES

Les différents modes d'émission.....	192
Le choix du matériel radio.....	196
Des fréquences pour commencer l'écoute.....	198
Références bibliographiques.....	201

AVERTISSEMENT

Nous avons attaché une grande importance au contrôle des logiciels proposés dans cet ouvrage, et tous ont été testés sur CPC 464 et 664. Les listings étant reproduits directement par un procédé photo, il ne peut y avoir, à ce niveau, d'erreur d'impression.

Les auteurs restent néanmoins ouverts à toute suggestion visant à améliorer les logiciels ici proposés.

Dans le reste de cet ouvrage, sauf indication contraire, nous désignerons par CPC, aussi bien le 464 que le 664 ou encore, le 6128

DROITS DE REPRODUCTION

Pour se garantir du piratage sous toutes ses formes, incluant la reproduction de tout ou partie des logiciels et interfaces décrits dans ce livre, les auteurs se réservent tous les droits de commercialisation. En conséquence, aucun logiciel, aucune interface (montée ou en kit), ne pourront être commercialisés sans l'autorisation écrite des auteurs et ce, par quelque société que ce soit.

Le prix raisonnable des lecteurs de disquettes du système AMSTRAD laisse à penser que, en raison du confort d'utilisation et des capacités qu'ils apportent, bon nombre des utilisateurs ayant dépassé le simple stade du jeu en sont, ou en seront, équipés.

Néanmoins, ce livre a été écrit dans l'optique de présenter la plus totale compatibilité entre les différents AMSTRAD. C'est pourquoi nous glissons ici quelques conseils à nos lecteurs ne possédant que le CPC 464 de base.

Les programmes faisant appel à des 'fichiers' extérieurs tels que le RTTY (pour ses mémoires) ou le 'CONTEST', ou encore la 'SSTV', s'ils sont plus faciles à utiliser avec une disquette, n'en sont pas moins exploitables sur cassettes. Il faut simplement s'astreindre à une certaine discipline: les programmes sur une cassette, les 'fichiers' sur une autre. Tenir à jour une petite fiche avec les index compteur est peut-être fastidieux, mais s'avèrera rapidement très utile.

COMMENTAIRES AU SUJET DES LISTINGS DE CE LIVRE

Les listings de ce livre sont édités volontairement en 40 colonnes. Ce choix a été fait en pensant aux lecteurs qui possèdent un moniteur couleur, sur lequel il est extrêmement pénible de travailler longtemps sur des textes en 80 colonnes. En regardant votre listing, vous avez l'image exacte de ce que vous devez voir sur l'écran.

D'autre part, un signe particulier est parfois utilisé: il s'agit du symbole d'élévation à la puissance (flèche montante). Ce symbole est représenté sur l'imprimante comme suit ^. Ne vous méprenez pas, vous le trouverez sur la même touche que le symbole de la livre anglaise...

Enfin, et toujours pour les possesseurs d'un écran couleur, nous suggérons de mettre en début de programme une ligne:

```
INK 0,13:INK 1,0:BORDER 0
```

Ceci aura pour effet d'écrire en noir sur fond gris ce qui nous semble plus reposant que le jaune sur fond bleu.

ATTENTION ! Utilisateurs du CPC 464, lors de l'ouverture de fichiers, un bug gênant perturbe le fonctionnement de la machine. Vous devrez placer en début de programme (si des fichiers y sont manipulés, comme dans RTTY et CONTEST par exemple) une ligne :

```
numéro OPENOUT "FICTIF" : MEMORY HIMEM-1 : CLOSEOUT
```

Ne l'oubliez pas... Ceci ne concerne pas les CPC 664 et 612B.

Les listings désassemblés (ou en "assembleur" commenté) sont fournis aux seules fins de documenter le lecteur. Ils ne devront évidemment pas être saisis, le langage machine correspondant étant le plus souvent intégré au programme BASIC utilisateur sous la forme de DATA's.

Des cassettes et disquettes, contenant les logiciels décrits dans cet ouvrage seront commercialisées par les Editions SORACOM. Le lecteur pourra se les procurer afin d'éviter la saisie des différents listings.

Ce livre a été entièrement composé, par les auteurs, à l'aide du logiciel de traitement de texte 'TASWORD'.

INTRODUCTION

Communiquer est un besoin que nous avons tous... Notre Hobby, ou loisir pour rester français, qu'il soit simplement l'écoute ou l'émission-réception, nous conduit à utiliser les techniques les plus récentes, alors pourquoi ne pas utiliser l'ordinateur ?

L'ordinateur est capable de réaliser les tâches les plus répétitives, les plus astreignantes, sachons en tirer parti...

Déjà les robots supplantent les hommes dans les usines pour des raisons d'efficacité, au prix il est vrai de pertes d'emploi, mais là n'est pas notre propos. Partout où la sécurité est de rigueur, l'ordinateur est présent: guidage automatique des avions ou des bateaux, contrôles d'accès, etc...

L'ordinateur n'a rien de secret pour qui veut bien s'y intéresser et on table bien souvent sur notre ignorance en ce domaine pour nous faire avaler les pires excuses lors de retards (pas de commentaires...)

Alors l'ordinateur s'est mis au quotidien: vous pouvez désormais accéder aux plus gros d'entre eux grâce au réseau téléphonique par l'intermédiaire d'un simple MINITEL. Les banques de données s'ouvrent aux particuliers et même des amateurs y déposent des informations au service d'autres amateurs: éphémérides pour la prévision de passage des satellites, infos DX etc...

Ainsi le micro-ordinateur est arrivé sur votre table, à côté du récepteur de trafic: à vous de savoir l'utiliser et c'est là le but de cet ouvrage. Nous espérons que, au fil des pages, l'envie vous viendra d'aller plus loin et de compléter, de personnaliser, d'améliorer les logiciels que nous vous proposons. Notre but sera alors atteint.

Préparez votre CPC et courage !

PRESENTATION DU CPC

Souhaitant que cet ouvrage puisse satisfaire autant les programmeurs expérimentés, n'ayant pas le temps de concevoir leurs logiciels de communication, que les novices désireux de découvrir un nouveau monde, nous décrirons, à l'intention de ces derniers, la machine avec un maximum de détails. Que les premiers nous pardonnent et passent à la partie "programmes" de ce livre.

Tout travail de programmation nécessite, auparavant, une phase d'analyse qui requiert, autant que de bonnes connaissances du langage utilisé, une maîtrise des possibilités matérielles de la machine. C'est le lien qui existe entre le matériel (hardware ou "hard") et le logiciel (software ou "soft").

LE SYSTEME MINIMUM

Le système minimum de notre machine se compose de l'unité centrale, que nous détaillerons, du magnétophone ou/et de la disquette comme mémoire de masse, et de l'écran (moniteur) pour la visualisation.

Dans le cas des applications que nous proposons ici, exception faite peut-être de la télévision d'amateur, il n'est pas nécessaire et...même peu souhaitable pour les yeux de posséder un moniteur couleur.

Pour le stockage des programmes, que vous utilisiez des disquettes ou des cassettes, il sera bon de prendre quelques précautions et de suivre ces quelques conseils :

- Ne jamais poser vos cassettes et disquettes sur le Haut-Parleur de votre récepteur ou près du transformateur de votre alimentation de puissance.
- Doublez toujours vos sauvegardes de programmes sur des supports différents entreposés en des lieux, eux aussi, différents.
- Manipulez ces disquettes et cassettes avec soin. Ne pas retirer les disquettes tant que le moteur est encore en rotation. Rembobinez vos cassettes avant de les retirer du magnétophone.

- Le magnétophone requiert un minimum d'entretien: nettoyage périodique des têtes, avec un coton tige imbibé d'alcool dénaturé, et séances de démagnétisation, au moyen d'un outil adapté, sont les gages de garantie de lectures fiables.

- Par contre, ne touchez jamais à votre lecteur de disquettes: son entretien n'est pas du domaine de l'amateur non outillé.

LES ATOUTS DU CPC

Le CPC possède bien des atouts, c'est d'ailleurs la raison probable pour laquelle nous l'avons choisi... Outre son rapport qualité/prix inégalé, cause de son énorme succès, le CPC offre du jamais vu dans le domaine des ordinateurs familiaux, ne serait-ce que par sa configuration complète.

- un clavier de bonne qualité, bien que sur le 664 et sur les derniers 464 cela ne soit plus vrai, hélas, économies obligent (et ont permis une baisse de prix).

- la possibilité d'afficher des écrans sur 80 colonnes est certainement un avantage incontestable. Evidemment ceci se traduit par une image bien plus difficile à lire sur les écrans couleur, à cause de leur médiocre résolution. Par contre, il n'y a rien à dire sur la configuration monochrome...

- les interfaces essentielles sont incorporées : CENTRONICS pour l'imprimante, ou l'on peut tout à la fois contester et regretter le choix des concepteurs sur la liaison 7 bits au lieu de 8. Ceci conduit à acquérir, ou à construire, une interface pour bénéficier des possibilités graphiques de l'imprimante, à moins de jongler avec un logiciel spécialisé. Ouvrons, à ce propos, une parenthèse pour déconseiller l'imprimante DMP1 qui, à notre avis, offre peu d'avantages en regard de son prix. L'acheteur potentiel aura tout intérêt à réfléchir avant de choisir et de signer son chèque car, pour 2 à 300 Francs de mieux, il pourra trouver des modèles plus séduisants, plus rapides, et moins bruyants... D'ailleurs, cette imprimante n'est plus fabriquée : elle est remplacée, dans la gamme, par la D2000 offrant davantage de possibilités.

L'autre interface sera citée ici pour mémoire puisqu'il s'agit de l'entrée 'Joystick', plus destinée aux applications ludiques qu'à celles que nous vous proposons dans cet ouvrage. Néanmoins, il est possible de s'en servir comme entrée logique, pour lire des informations, voire des contacts, mais on reste tributaire du 'scanning' du clavier puisque la prise 'joystick' y est reliée.

- le BASIC est extrêmement complet, ce qui simplifiera la vie du programmeur au prix, il est vrai, de pas mal d'heures de travail pour apprendre à maîtriser toutes ses possibilités: gestion d'interruptions, fenêtrage, instructions graphiques, couleurs, et sonores très performantes.

- enfin, et ce n'est pas un moindre détail, il semblerait que la fiabilité de cette machine soit assez bonne si on en juge par le faible taux de retour en Service Après Vente. Ceci est du, en grande partie, à une conception et une architecture interne saines.

ARCHITECTURE INTERNE DU CPC

Le cœur du système est le microprocesseur Z80 (fabriqué par ZILOG), vieux cheval qui n'en finit pas de partir en retraite et que l'on donnait pour 'dépassé' il y a déjà deux ans... Belle leçon ! Il partage le rôle de vedette avec un circuit complexe, qui remplace à lui seul une centaine de circuits intégrés, que nous nommerons ULA (Uncommitted Logic Array) et qui commande une grande partie du système.

Côté mémoire, nous devons faire la différence entre la RAM et la ROM. La RAM (Random Acces Memory), est la mémoire dans laquelle on peut écrire et lire des données. C'est elle qui contiendra vos programmes, l'image vidéo (celle que vous voyez sur votre écran), et toutes sortes d'informations, propres au système, que nous nommerons abusivement 'variables'. La ROM (Read Only Memory) est une mémoire qui ne peut être que lue. Elle est programmée en usine et contient le BASIC et ce que nous appellerons 'le système d'exploitation' (à ne pas confondre avec celui du disque). Notre cher CPC contient 64 Kilo-octets (Ko) de mémoire 'vive' (RAM) et 32 Ko de mémoire 'morte' (ROM), répartie en deux zones de 16 Ko. Rappelons à nos lecteurs débutants, qu'un Kilo-octet égale 1024 octets... Les plus chevronnés, en additionnant ces deux valeurs, 64 et 32, s'étonneront peut-être de trouver 96 Kilo-octets, ce qui peut paraître paradoxal quand on sait que le Z80, comme ses frères à 8 bits ne peut adresser plus de 64 Ko... Ceci est rendu possible grâce à un ingénieux mécanisme logique qui permet de valider, à une même adresse, selon certaines conditions, soit de la RAM, soit de la ROM. Les composants utilisés sont, pour la RAM des très classiques 4164, mémoires dynamiques de 64 K sur 1 bit. Elles sont donc au nombre de 8 pour former les 64 Kilo-octets annoncés. La ROM est constituée par un 'chip' unique 23256 organisée en 32 Kilo sur 8 bits soit 32 Kilo-octets. Ajoutons enfin que la mémoire peut être étendue, en ROM ou en RAM, par des extensions externes en blocs de 16 ko jusqu'à 252 Ko. Voilà un système capable d'évoluer !

Le microprocesseur a besoin, pour fonctionner, d'une horloge: sa fréquence est de 4 MHz sur le CPC mais la vitesse de travail n'est que de 3.5 Mhz environ pour des raisons que nous ne détaillerons pas ici. En fait, le 4 MHz est obtenu à partir d'un quartz de 16 MHz qui rythme tout le travail du CPC. C'est l'ULA qui reçoit ce signal d'horloge issu de l'oscillateur à quartz 16 MHz et qui le divise avant d'envoyer le 4 Mhz vers le Z80 et du 1 MHz vers le générateur sonore.

L'autre circuit important, le Gate Array, que nous nommons aussi ULA, est construit spécialement pour AMSTRAD. Ne le cherchez pas sur des catalogues de circuits intégrés, il a spécialement été conçu par les ingénieurs pour cette machine et fabriqué uniquement pour elle. Peu d'informations sont connues à son sujet, sinon qu'il sert à la gestion des accès mémoire et remplit des fonctions dans le traitement des couleurs.

Les esclaves de ces deux maîtres sont le circuit de contrôle d'affichage, appelé VDU (Video Display Unit, 6845 de MOTOROLA), le contrôleur d'entrées-sorties appelé PPI (Programmable Peripheral Interface, 8255 de INTEL) et le générateur sonore PSG (Programmable Sound Generator, AY-3-8912 de General Instruments). Ils accomplissent bien des tâches sous les ordres du grand chef d'orchestre et de son acolyte. Nous ne détaillerons pas ici le fonctionnement de tous ces circuits mais nous jugeons nécessaire de vous donner quelques informations que les moins curieux de nos lecteurs pourront sauter... Avant que ceux-ci ne passent au chapitre suivant, nous voudrions attirer leur attention sur la curieuse interface imprimante.

L'INTERFACE IMPRIMANTE

S'il y a quelque chose de contestable sur le CPC, c'est bien l'interface imprimante qui a été réalisée au maximum d'économie. Elle est réduite à sa plus simple expression : un latch 74LS273... et comme un latch n'a que 8 sorties et que l'une d'entre elles est utilisée pour servir de STROBE (signal validant le transfert des données vers l'imprimante) vous comprendrez aisément qu'il va manquer un bit à notre liaison CENTRONICS.

Pour ceux qui ne le sauraient pas, la liaison de type CENTRONICS est un ensemble de 8 'fils' acheminant les 8 bits de donnée et d'au moins 2 signaux de contrôle : BUSY, indiquant que l'imprimante est en train de travailler, et ne peut par conséquent accepter des données et STROBE défini ci-dessus. Le bit manquant va nous faire défaut pour passer l'imprimante en mode graphique... Pourquoi ? Simplement parce que la majorité des imprimantes s'en servent pour déterminer ce mode. De plus, le 8ème bit de la sortie imprimante est fixé à la masse.

Dans leur souci de mettre des masses partout sur cette liaison imprimante, les concepteurs ont fixé à 0 le fil qui est relié au point 14 du connecteur standard CENTRONICS, correspondant à un saut de ligne automatique sur l'imprimante. Vous comprenez maintenant pourquoi, en utilisant une imprimante autre que la DMP1, on obtient, sur le papier, une ligne sur deux en blanc... Un simple cutter, ou un coup de fer à souder, pourront résoudre ce problème.

Signalons enfin que le 'mini-port' de sortie, constitué par les sept lignes de données, et l'entrée sur la ligne 'BUSY' pourront être utilisés pour d'autres tâches, mais n'anticipons pas...

QUELQUES DETAILS SUR LE HARD INTERNE

Sans entrer en profondeur dans le fonctionnement des circuits situés au cœur du CPC, nous pensons que l'esprit technicien de nos lecteurs mérite quelques précisions sur les circuits principaux.

LE CONTROLEUR D'ECRAN (CRTC) 6845

CRTC signifie Cathode Ray Tube Controller. Le 6845 a été développé par MOTOROLA, il y a déjà quelques années, et repris par EFCIS. En tant que contrôleur d'écran, il est capable de générer les signaux de synchronisation (Horizontale et Verticale), de gérer la mémoire génératrice de caractères, la mémoire (RAM) vidéo (ou sont rangées les pages écran). Il assure aussi la gestion du curseur et d'un éventuel 'crayon optique'.

La RAM vidéo est de 16k. C'est aussi le 6845 qui en assure le rafraîchissement.

La puce est reliée au monde extérieur par 40 connexions.

Le fonctionnement du 6845 est régi par 19 registres, l'un s'occupant de l'adressage des 18 autres.

Les registres qui nous intéressent sont appelés R0 à R17. Nous avons la possibilité d'y accéder par l'intermédiaire de l'instruction OUT du BASIC ou encore, par une routine en langage machine. Ceci permet d'obtenir quelques effets assez intéressants.

Le CRTC occupe deux emplacements en mémoire:

- &BCxx pour l'adressage des registres internes.
- &BDxx pour l'envoi des données dans le registre adressé.

Essayez le petit programme suivant, pour vous convaincre des résultats que l'on peut obtenir en 'jouant' avec ces registres.

```
10 OUT &BC0D,13
20 FOR N= 1 TO 40
30 OUT &BD0D,N
40 FOR K= 1 TO 15:NEXT
50 NEXT
```

Faites d'abord CLS:LIST puis RUN.
Pas mal le Scrolling latéral, non ?

Les registres R0 à R3 sont responsables du Timing Horizontal.
Les registres R4 à R9 sont responsables du Timing Vertical.
Les registres R10/11 et R14/15 gèrent le curseur.
Les registres R12/13 rafraichissent la RAM vidéo.
Les registres R16/17 gèrent le crayon optique.

LE ROLE DES REGISTRES

R0	Nombre TOTAL de caractères affichables sur une ligne.
R1	Nombre de caractères affichés sur une ligne.
R2	Position synchro. horizontale.
R3	Largeur du signal de synchro.
R4	Nombre total de lignes affichables.
R5	Ajustement de la synchro. verticale.
R6	Nombre de lignes affichées sur l'écran.
R7	Position du signal de synchro. verticale.
R8	Balayage entrelacé ou non.
R9	Nombre de lignes de balayage pour un caractère (hauteur).
R10	Ligne de départ du curseur + clignotement éventuel.
R11	Ligne de fin du curseur.
R12	Adresse de départ de la RAM vidéo (octet poids fort).
R13	Adresse de départ de la RAM vidéo (octet poids faible).
R14	Position actuelle du curseur (poids fort).
R15	Position actuelle du curseur (poids faible).
R16	Position RAM vidéo au moment du STROBE Light pen (Fort).
R17	Position RAM vidéo au moment du STROBE Light pen (Faible).

VALEURS CONTENUES PAR LES REGISTRES EN MODE 1

R0	63
R1	40
R2	46
R3	142
R4	38
R5	0
R6	25
R7	30
R8	0
R9	7

R10 à R15 sont en permanence modifiés par le Soft.

LE GATE ARRAY

Ce circuit complexe à 40 broches, remplace à lui tout seul une floppée de circuits intégrés. Sans lui, le prix du CPC ne serait certainement pas ce qu'il est. Conçu exclusivement pour le CPC, (vous ne le trouverez dans aucun catalogue de composants), il gère la RAM, en contrôlant les accès, fabrique les signaux d'horloge, produit la vidéo et les couleurs.

Il occupe un seul canal d'entrées-sorties, les 2 bits supérieurs de la donnée envoyée lors de son adressage permettant l'accès aux 4 registres principaux qui sélectionnent ses différentes fonctions.

On peut y accéder par un OUT &7Fxx.

	Bit 7	Bit 6	Registre sélectionné	
R0	0	0	Registre numéro de couleur (palette)	<=> PEN
R1	0	1	Registre valeur de couleur	<=> INK
R2	1	0	Aiguillage ROM et sélection mode écran.	
R3	1	1	Role non précisé.	

R0 : Les bits 5,6,7 sont inutilisés.
Bit 4=0 on charge une couleur d'encre.
Bit 4=1 on charge une couleur BORDER.
Bit 0à3 donne la couleur de l'encre.

R1 : Les bits 5,6,7 sont respectivement à 0,1,0.
Les autres bits donnent le numéro de la couleur (0 à 31).

R2 : Les bits 5,6,7 sont respectivement à 0,0,1.
Bit 4=1 Force compteur synchro V à zéro.
Bit 3=0 Valide la ROM supérieure. (invalide si 1)
Bit 2=0 Valide la ROM inférieure. (invalide si 1)
Bit 1à0 Sélection du mode écran. (voir tableau)

Bit 1	Bit 0	Mode écran
0	0	0
0	1	1
1	0	2
1	1	Inutilisable

Vous pouvez expérimenter, sans autre risque que la perte de contrôle du clavier, l'envoi de différentes valeurs vers ces registres. Les registres du GATE ARRAY ne peuvent pas être lus, on peut seulement y envoyer des données.

Nous n'avons pas trouvé d'application pratique à l'utilisation directe de ces registres, toutes les sélections qu'ils provoquent étant accessibles, en général par des routines système plus adaptées.

LE PPI 8255

Sous cette abréviation se cache son nom exact: Parallel Peripheral Interface. C'est un circuit universel d'entrée sortie pouvant être utilisé avec d'autres microprocesseurs (tel le 8080). Physiquement, il se présente sous la forme d'un boîtier à 40 broches. Il intègre 3 ports d'entrées sorties. Selon son mode de fonctionnement les lignes des ports sont groupées différemment.

S'il est difficile de 'récupérer' le 8255 du CPC pour des applications personnelles, il est néanmoins intéressant de savoir à quoi il sert.

Dans l'espace des entrées sorties du CPC, il occupe 4 emplacements:

```
CONTROLE  &F7xx
PORT A    &F4xx
PORT B    &F5xx
PORT C    &F6xx
```

Rappelons simplement qu'un PORT est un groupe de 8 lignes...

Le registre de contrôle sélectionne le mode de fonctionnement du PPI et valide le sens des différents PORTS.

Le bit 7 de ce registre, mis à 1, indique que la donnée est un mot de contrôle, sinon qu'elle s'adresse au PORT C.

Dans ce qui suit, 'mode' désigne l'un des 3 modes de fonctionnement du PPI (rien à voir avec les modes écran...).

```
Bit 6      0 -> mode 0
           1 -> mode 2
Bit 5      0 -> mode 0
           1 -> mode 1
Bit 4      0 -> PORT A en SORTIE
           1 -> PORT A en ENTREE
Bit 3      0 -> PORT C en SORTIE  (Bits de poids fort du PORT C)
           1 -> PORT C en ENTREE  (   =       =       =   )
Bit 2      0 -> mode 0
           1 -> mode 1
Bit 1      0 -> PORT B en SORTIE
```

1 -> PORT B en ENTREE
 Bit 0 0 -> PORT C en SORTIE (Bits de poids faible du PORT C)
 1 -> PORT C en ENTREE (= = =)

MODE 0: Chaque groupe de 12 lignes peut être programmé en ENTREE ou en SORTIE, par blocs de 4.

MODE 1: Chaque groupe peut être programmé avec 8 lignes en ENTREE ou en SORTIE. 3 des 4 restantes servent au Handshaking et Interruptions.

MODE 2: Le 8255 devient un coupleur de bus bidirectionnel (8 bits) avec 5 lignes de Handshaking.

Notez que le PORT C peut être divisé en 2 fois 4 bits...

Sur le CPC, le 8255 est utilisé comme suit:

Le PORT A est programmé en entrée sortie et assure la liaison avec le générateur sonore (8912) pour le transfert des données.

Le PORT B est utilisé ainsi:

- Bit 7: lecture cassette.
- Bit 6: signal BUSY de l'imprimante.
- Bit 5: détecte la présence d'une extension.
- Bit 0: état de la synchro verticale (retour spot).

Les bits 1 à 4 sont reliés à des straps internes, cablés en usine, fonction des pays auxquels sont destinés les CPC.

Le PORT C est utilisé ainsi:

- Bit 7: en sortie, commande BDIR du 8912.
- Bit 6: en sortie, commande BCI du 8912.
- Bit 5: écriture cassette.
- Bit 4: télécommande du moteur magnétophone.

Les bits 0 à 3 sont reliés aux lignes matrices du clavier.

Pour utiliser le PPI, on envoie un mot de contrôle au registre de CONTROLE puis on envoie (ou on lit) la donnée du PORT correspondant.

Voici un exemple, sachant que le PORT C du PPI est programmé d'origine en SORTIE. Nous allons fermer le relais de commande moteur du magnétophone.

L'adresse du PORT C étant &F6xx on écrira:

OUT &F6DD,&10 ce qui aura pour effet de mettre à 1 le bit 4.

Pour ouvrir le relais on enverra la valeur 0...

On l'appellera en abrégé PSG. Conçu par GENERAL INSTRUMENTS c'est un circuit qui est utilisé également sur d'autres micro-ordinateurs et jeux vidéo. Il est capable de produire les sons les plus fous ou les plus harmonieux... Il intègre également un port d'entrée sortie qui, sur le CPC, est utilisé pour la lecture du clavier et de la manette de jeux.

Couplé au travers du PPI, le GSP n'occupe pas d'emplacement dans les entrées sorties du système.

Ses 28 pattes laissent deviner une structure interne bigrement complexe mais l'utilisateur n'aura pas à s'en soucier puisqu'il ne le connaîtra qu'au travers de ses 15 registres (16 en fait, mais un n'est pas accessible).

Les sons sont produits à partir d'un signal d'horloge (externe) et d'un diviseur programmable. En agissant sur le facteur de division on obtient des fréquences différentes.

Les différents registres internes sont appelés au travers d'entrées de commande BDIR, BC1, BC2. Sur le CPC, BC2 n'est pas utilisée: elle est fixée au 5V.

EFFETS DES ETATS DE BDIR ET BC1:

BDIR	BC1	FONCTION
0	0	le bus de données du PSG est mis à haute impédance
0	1	on peut lire les données dans les registres PSG
1	0	on peut écrire des données dans les registres PSG
1	1	adressage d'un nouveau registre du PSG

ROLE DES DIFFERENTS REGISTRES DU PSG:

Numéro	Role	Bits utiles
R0	Réglage fin tonalité canal A	8
R1	Réglage gros =	4
R2	Réglage fin tonalité canal B	8
R3	Réglage gros =	4
R4	Réglage fin tonalité canal C	8
R5	Réglage gros =	4
R6	Période du bruit	5
R7	Contrôle mélange sons et bruits	8
R8	Volume canal A	5
R9	Volume canal B	5
R10	Volume canal C	5
R11	Réglage fin période enveloppe	8
R12	Réglage gros =	8
R13	Sélection enveloppe	4
R14	Données du PORT	8

LES BITS DU REGISTRE DE CONTROLE

Bits 7 à 6 : controle port E75

Bits 5 à 3 : controle générateur de bruit

Bits 2 à 0 : controle générateurs de tonalité

Pour les bits 0 à 5, un 0 équivaut à une autorisation, un 1 à une interdiction.

Bit 0 : géné. ton canal A

Bit 1 : géné. ton canal B

Bit 2 : géné. ton canal C

Bit 3 : géné. bruit canal A

Bit 4 : géné. bruit canal B

Bit 5 : géné. bruit canal C

L'horloge envoyée au GSP est à 1 MHz. Le premier travail du GSP sera de la diviser par 16. La valeur résultante sera alors 62.5 KHz. C'est à partir de cette fréquence d'horloge, et selon le contenu des registres (facteur de division programmable) que sera obtenue la fréquence finale du son...

La formule magique est donc $62500/\text{Fréquence désirée}=\text{Facteur}$

Ce facteur de division est mis dans les registres de réglage 'fin' et 'gros' du canal correspondant. Seuls 12 bits étant utilisés, 8 sur le réglage 'fin' et 4 sur le réglage 'gros', le facteur maximum de division sera 4095 ce qui correspond à une fréquence minimum de $62500/4095=15$ Hz.

Pour le registre de bruit, le principe est le même mais le facteur de division, limité à 5 bits ne pourra dépasser 31 ce qui donnera une fréquence de bruit minimum de l'ordre de $62500/31=2000$ Hz.

Toutes ces précisions sont données au lecteur dans le seul but de satisfaire sa curiosité car, en définitive, l'utilisateur n'aura que rarement besoin d'aller 'jouer' directement avec les registres du PSG: les routines système sont là pour cela...

LE MICROPROCESSEUR Z80A

Il était difficile de passer en revue les composants du CPC sans dire un mot du chef d'orchestre: le Z80A. Plusieurs livres lui sont consacrés et notre propos n'est pas d'en faire un pâle résumé de quelques lignes, aussi nous renverrons le lecteur curieux aux ouvrages cités en référence bibliographique.

Néanmoins, il nous paraît indispensable de donner les quelques informations qui suivent.

Le Z80 est un dérivé du 8080 de INTEL. Fabriqué par ZILOG il reste, dans ses grandes lignes, très proche de son prédécesseur avec lequel il assure une compatibilité dite 'montante' (les programmes écrits pour 8080 tournent sur Z80 mais pas l'inverse).

De la famille des 8 bits, il possède un jeu d'instructions relativement vaste. Sur le CPC, il reçoit un signal d'horloge à 4MHz et, comme il est bien secondé par d'autres composants du système, il en fait une des machines 8 bits les plus rapides du moment.

Le Z80 est bâti autour des registres suivants:

- L'accumulateur A, registre (8 bits) privilégié s'il en est puisque directement relié à l'unité arithmétique et logique.
- Le registre d'états F (Flag register, 8 bits) composé de différents indicateurs reflétant le résultat des opérations arithmétiques ou logiques.
- 6 autres registres 8 bits nommés B,C,D,E,H,L.

Tous ces registres peuvent être appariés selon le schéma suivant:

- A avec F
- B avec C
- D avec E
- H avec L

On obtient alors des registres '16 bits'.

Tous ces registres sont doublés par des registres 'alternés' ou registres 'bis'... A'F' pour AF, B'C' pour BC, D'E' pour DE, H'L' pour HL.

Les registres secondaires ainsi définis ne sont hélas pas accessibles aisément sur le CPC mais, rassurez-vous, il y a déjà de quoi faire avec les registres principaux...

Les registres 16 bits obtenus sont bien utiles pour l'adressage ou encore, pour la manipulation de données en tables. HL fait même souvent office d'accumulateur 16 bits.

Deux autres registres sont très importants: IX et IY appelés registres d'index. Ce sont eux qui servent à pointer, à désigner des données en tableaux.

Enfin SP, le pointeur de pile (Stack Pointer) et PC, le compteur programme (Program Counter) complètent la panoplie.

Le microprocesseur est relié au reste du système par un bus de DONNEES à 8 bits et un bus d'ADRESSES à 16 bits. Il possède également des signaux de CONTROLE qui lui permettent de collaborer avec les autres composants de l'ordinateur. La puce est reliée au monde extérieur par un boîtier à 40 pattes.

Voilà, le tour d'horizon 'HARD' est terminé. Redisons-le, le lecteur intéressé par la technique aura avantage à se reporter aux différents ouvrages cités en référence...

OFFREZ UN 8ème BIT A VOTRE IMPRIMANTE

Economie lourde en conséquences, le 8ème bit absent de l'interface imprimante de l'AMSTRAD, prive l'utilisateur de possibilités fort intéressantes.

Les utilisateurs d'une imprimante AMSTRAD ne sont pas concernés par cette modification.

Pour remédier à ce problème, il suffit de regarder le schéma électronique de l'AMSTRAD pour se rendre compte que, le bit de poids fort, fixé à la masse, peut être récupéré au prix, il est vrai, d'un peu d'acrobatie, de quelques centimètres de fil, d'une bonne dose de soins... et d'un peu de patience. Il faudra également utiliser un fer à souder très fin et peu puissant, surtout pas un 'chalumeau'...

ATTENTION, si votre machine est encore sous garantie, n'effectuez pas cette modification car elle annulerait tout recours.

A cette modification matérielle, il y aura lieu d'ajouter un peu de logiciel: si 2 lignes de programme et 2 soudures ne vous font pas peur, en piste pour la modification.

Ouvrir la machine avec précautions.

Repérer sur le circuit imprimé du connecteur imprimante le point 9, correspondant au bit 7. Ce point est relié à la masse.

Avec un cutter, interrompre la liaison sur environ 3 millimètres.

Couper quelques centimètres de fil de cablage, souple et fin.

Repérer, sur le circuit imprimé, la patte 12 du 8255 (circuit à 40 broches situé tout près du connecteur imprimante). Pour la déterminer, partir du point 1 situé à gauche de l'encoche pratiquée dans le boîtier du 8255.

Ce point correspond au bit 5 du port C du 8255: il sert à l'écriture des données sur la cassette. Comme on n'écrit jamais en même temps sur l'imprimante et la cassette, il n'y a pas de conflit possible.

Souder le fil souple entre les points 12 du 8255 et 9 du connecteur imprimante.

S'assurer qu'il n'y a pas de bavures de soudure.

Refermer le boîtier de l'AMSTRAD. Votre imprimante vous remercie !

Côté logiciel, c'est très simple. Pour forcer le bit 7 à 1, à chaque fois que vous en aurez besoin (par exemple pour utiliser le mode graphique de l'imprimante), il suffira d'un `OUT &F600,32`. Attention, il faut savoir que le bit 7 restera à 1 tant que vous ne l'aurez pas implicitement remis à 0. Pour ce faire, il suffira d'un `OUT &F600,0`. Ces 2 instructions seront insérées judicieusement dans votre programme BASIC.

Les utilisateurs du langage machine auront recours aux 3 lignes d'assembleur suivantes:

```
Mise à 1 du bit 7      LD B, F6H
                       LD A, 20H
                       OUT (C),A
                       RET
```

```
Mise à 0 du bit 7     LD B, F6H
                       LD A, 00H
                       OUT (C),A
                       RET
```

Cette modification nous a été communiquée par Louis GENTY, également auteur aux Editions SORACOM. Qu'il en soit ici remercié.

LA MEMOIRE D'ECRAN

Nous ne pouvions la passer sous silence car elle a une organisation assez particulière. Rappelons que son rôle consiste à maintenir toutes les informations, y compris celles qui concernent la couleur, présentées à l'écran.

Elle est organisée en 8192 paires d'octets (16Ko). En MODE 2, le plus gourmand, il faut 1 bit par point écran. On ne peut représenter qu'une seule couleur. En MODE 1, 2 bits sont nécessaires pour coder 1 point: il y a 4 couleurs possibles. En MODE 0, il faudra 4 bits par point, ce qui permet de coder 16 couleurs simultanément.

Sur bien des ordinateurs, les adresses mémoire de l'écran se suivent rigoureusement, colonne après colonne, ligne après ligne. Sur le CPC, il n'en est pas ainsi. Cette particularité ne facilite pas la tâche du programmeur pour

des applications telles que la SSTV ou le FAX. Le passage d'une ligne à une autre s'obtient en additionnant la valeur &800 à l'adresse de la ligne précédente. ATTENTION, par ligne nous entendons 'ligne balayage' et non pas 'ligne caractère écran'...

Voici une représentation de l'organisation mémoire.

	Octet 1		Octet 80
Ligne 001	C000 C001 C002 C003		C04D C04E C04F
Ligne 002	C800 C801 C802		C84F
Ligne 003	D000 D001 D002		D04F
.....			
.....			
Ligne 199	F780		F7CF
Ligne 200	FF80		FFCF

Cette structure est très complexe et certains octets demeurent inutilisés.

En résumé, nous pouvons dire que, les 8 lignes de points formant la matrice d'un caractère ne sont pas contenues dans la même adresse mémoire. On ne peut pas 'POKER' un code ASCII dans la mémoire écran pour voir s'afficher le caractère correspondant. Le processeur écrit une ligne sur huit sur tout l'écran puis remonte à la seconde ligne et réécrit une ligne sur huit etc.

Pour avoir une idée de ce qui se passe il suffit d'observer le changement d'un écran graphique (voir les présentations des cassettes de jeu). Le phénomène au ralenti, est illustré par le petit programme ci-dessous.

```

10 MODE 2
20 FOR N = &C000 TO &FFFF
30 POKE N,255
40 NEXT

```

LANGAGE ET PROGRAMMATION

Les liens qui existent entre l'opérateur (utilisateur) et sa machine sont le clavier, l'écran et, dans une certaine mesure, l'imprimante. Mais ces liens ne valent que par le logiciel contenu dans la machine, faute de quoi celle-ci ne serait qu'un ensemble inerte de circuits intégrés incapable de faire autre chose que générer des 'Syntax Error' à l'utilisateur néophyte. Son savoir, elle le tient du programmeur qui lui a inculqué, par l'intermédiaire du logiciel qu'il a créé, la manière de dialoguer avec l'utilisateur. Nous allons voir, brièvement comment le programmeur et la machine peuvent se comprendre.

Le coeur de l'ordinateur, nous le savons maintenant, est constitué d'un microprocesseur. Comme tout circuit intégré logique, ce dernier est incapable de comprendre autre chose que des états 0 ou 1. Il est capable de reconnaître et d'exécuter des instructions constituées d'un ensemble de 0 et de 1. Dans le cas du Z80, ces instructions sont groupées sur 8 bits. Ne pouvant travailler qu'en binaire, il serait horriblement long et fastidieux de dialoguer avec lui dans ce mode, c'est la raison pour laquelle on a été amené à créer des langages. Le rôle du langage est donc d'établir le lien entre le programmeur et le microprocesseur situé au coeur du micro-ordinateur.

L'instruction étant l'élément de base compris par le microprocesseur, et celle-ci ne pouvant lui parvenir n'importe comment, les instructions sont organisées en une suite logique et ordonnée capable d'être exécutée par la machine. Cette suite ordonnée d'instructions, vous l'avez compris, est le programme.

Revenons au langage. Le binaire est le seul capable d'être compris par l'ordinateur ; on parle aussi de langage machine. Pour éviter au programmeur de devoir s'abaisser au niveau de la machine, on a créé des langages 'évolués'.

Pour être compris correctement par la machine, le langage doit répondre à certains critères, de sémantique, de grammaire pourrait-on dire, tout comme dans le langage écrit ou parlé, ou on imagine mal des mots de vocabulaire assemblés n'importe comment. Ces règles de grammaire sont appelées 'SYNTAXE'. Si cette syntaxe n'est pas respectée, l'ordinateur détectera des erreurs. Le langage comprend une partie, appelée 'Analyseur Syntaxique', qui vérifie le respect de ces règles. Ceci est donc une partie importante du 'BASIC' qui équipe votre CPC.

COMPILATEUR OU INTERPRETEUR ?

La traduction du langage peut être effectuée de deux manières différentes.

LA COMPILATION : le programme est traduit en langage machine avant d'être exécuté. Cette traduction s'effectue en une ou plusieurs 'Passes'. Le résultat est la transformation du langage 'SOURCE' (celui dans lequel vous écrivez) en langage 'OBJET' (celui qui est destiné à la machine) : c'est la 'compilation'. Le programme qui accomplit cette tâche est le 'COMPILATEUR'. La compilation n'a lieu qu'une seule fois et génère des fichiers 'objets' directement (ou presque) exécutables.

L'INTERPRETATION : le programme est analysé ligne par ligne et chaque ligne est exécutée au fur et à mesure. A chaque lancement du programme, il y a une nouvelle 'interprétation' de chaque ligne. Ce travail est l'oeuvre d'un 'INTERPRETEUR'.

La première méthode est beaucoup plus efficace car elle permet une exécution beaucoup plus rapide du programme. En revanche, elle demande beaucoup plus de place mémoire au moment de la compilation. Sur certains micro-ordinateurs de haut de gamme, on trouve des BASIC compilés ou semi-compilés.

La seconde méthode, si elle a l'avantage d'économiser la mémoire (au moins un temps), a pour inconvénient sa lenteur. Le BASIC interprété permet néanmoins l'utilisation de la machine en mode conversationnel, par l'intermédiaire de ses organes d'entrée-sortie (clavier, écran, imprimante). Il est dit 'interactif'. La machine peut être utilisée immédiatement, comme une simple calculatrice.

PROGRAMMER, c'est connaître, à la fois, l'éventail des instructions acceptées par la machine (vocabulaire) et leur syntaxe (grammaire). Programmer efficacement c'est savoir regrouper ces instructions dans un ordre optimum permettant de bâtir un programme performant. Les tâches répétitives seront regroupées dans des 'Sous-programmes', programmes dans le programme.

L'ANALYSE ET L'ELABORATION DE L'ALGORITHME

L'analyse est le premier pas que devra franchir le programmeur avant d'écrire son logiciel. C'est pratiquement l'étape la plus importante, avec l'élaboration de l'organigramme, et... la plus souvent négligée par les programmeurs amateurs. Le résultat d'un programme écrit sans une analyse préalable est ce qu'on appelle souvent 'un plat de nouilles', c'est-à-dire quelque chose d'imparfait et de difficile à mettre au point et à modifier par la suite... Envisageriez-vous de construire un 'Transceiver' performant sans aucun schéma ?

La phase d'analyse permet, connaissant la fin, le but à atteindre, de définir au mieux les moyens. La dernière étape de l'analyse est l'élaboration de l'organigramme ou algorithme, véritable schéma de votre logiciel. Plusieurs algorithmes différents peuvent parfois conduire à la résolution d'un même problème. On cherchera alors la rapidité, la concision ou la clarté du programme selon le but à atteindre. L'algorithme, tout comme le plan d'analyse sur papier, aura tout intérêt à être abondamment commenté. Les commentaires essentiels pourront être reportés dans le programme, dans des lignes de REM. Toutes ces précautions permettront un 'dépannage' plus aisé du logiciel, en cas de besoin.

EXEMPLE D'ANALYSE

Notre but est de réaliser un logiciel capable de servir de 'moniteur' pour l'apprentissage de la télégraphie MORSE. Ce logiciel, compte-tenu des performances attendues et des qualités du CPC, pourra être écrit en BASIC.

Commençons l'analyse de la situation.

Que veut-on faire ?

Apprendre la télégraphie MORSE demande de suivre certaines règles pour la lecture au son. Ces règles vont définir les bases de notre programme.

Le premier pas à franchir par l'élève consiste à apprendre les différents signes qui composent l'alphabet MORSE. Ces signes ne s'apprennent pas dans n'importe quel ordre.

Les signes étant appris, il faudra s'entraîner à les reconnaître lorsqu'ils sont tous mélangés : c'est le second pas.

Capable de lire et de reconnaître 'au son' tous les signes, il faut maintenant acquérir de la vitesse pour être à l'aise au jour de l'examen : c'est la dernière étape. Bien entendu, la méthode de manipulation s'apprendra lors de la seconde étape.

Le programme devra donc permettre :

L'apprentissage des divers groupes de signes.

La génération de dictées d'entraînement.

La possibilité de lire au son à diverses vitesses.

Outre ces grandes lignes nous ajouterons :

Entraînement selon certains critères : chiffres seuls, lettres seules, ensemble des signes.

Entraînement au mode 'trafic' avec simulation par l'ordinateur de différents types de brouillages.

Les caractères émis s'afficheront à l'écran.

La dictée aura 255 caractères.

En apprentissage on émettra 23 groupes de 6 signes.

Voilà les grandes lignes de notre logiciel. La première étape est terminée. On peut en établir un premier ordinogramme.

Par la suite, on utilisera au mieux, les possibilités du CPC. Fenêtres et couleurs pour la présentation, canaux sonores multiples pour la sortie des sons. Ceci est un exemple simple. Pour des logiciels plus complexes, l'analyse devra être affinée avant d'entamer la programmation.

Vous trouverez, dans la section programmes de ce livre, le logiciel correspondant sous le nom de MONIMORSE.

BASIC OU LANGAGE MACHINE ?

S'il est facile d'apprendre le BASIC assez rapidement, il n'en va pas de même pour l'étude du langage machine. Pourtant, avec un peu de méthode, et à l'aide d'un ouvrage spécialisé, ce travail pourra être entrepris par l'amateur soucieux de découvrir un mode de programmation aussi captivant que performant.

Dans certaines situations, il sera même indispensable d'utiliser le langage machine, notamment pour des impératifs de rapidité. Il faut reconnaître que le BASIC du CPC est très rapide, ce qui permet d'envisager son utilisation même pour des tâches telles que le décodage de la télégraphie MORSE ou la réception des signaux RTTY à 45 bauds. Dans ce livre, nous ne l'avons pas fait car nous avons voulu démontrer qu'un savant dosage BASIC-Langage machine conduisait à une programmation équilibrée.

Pour ne pas pénaliser les lecteurs ne disposant pas d'un programme ASSEMBLEUR, nous nous sommes efforcés à intégrer le langage machine dans des lignes de DATA. Les modules en ASSEMBLEUR ne sont là que pour documenter les programmes...

L'assembleur utilisé par les auteurs n'est pas des plus performants, mais c'était le seul dont ils disposaient lors de l'écriture de ce livre. Reconnaissons lui quelques qualités, telles que: sa simplicité de mise en oeuvre, le fait qu'il intègre un moniteur de mise au point et un désassembleur. C'est également un des plus répandus: il s'agit de ZEN.

Le langage ASSEMBLEUR génère ce que l'on nomme le code objet (c'est-à-dire le langage machine). C'est ce code que nous avons réintroduit dans les lignes de DATA. En auscultant les zones mémoires correspondantes, au moyen d'un DESASSEMBLEUR, vous retrouveriez une suite d'instructions en MNEMONIQUES proche de celle fournie dans le listing ASSEMBLEUR commenté.

La programmation en ASSEMBLEUR est accessible à tous ceux qui sont prêts à consacrer une partie de leur temps libre à l'apprentissage du langage. Nous vous conseillons quelques ouvrages dans nos références bibliographiques.

UTILISER LES ROUTINES SYSTEME

Inutile d'écrire à nouveau ce qui existe déjà dans la ROM... Plusieurs routines système sont exploitables facilement par l'utilisateur, d'autant plus que le 'FIRMWARE' (Soft 158) décrit, avec force détails, leurs conditions d'accès et de sortie.

Nous avons résumé ici celles qui sont susceptibles d'être les plus utilisées par le lecteur de ce livre; nous avons volontairement passé sous silence les routines graphiques et sonores. Le lecteur intéressé consultera les éléments des références bibliographiques...

Dans ce qui suit, si des conditions d'entrée sont requises, elles sont précisées. Dans le cas contraire, ce point est passé sous silence. Il en est de même pour les conditions de sortie et les registres éventuellement modifiés.

ROUTINES CONCERNANT LE CLAVIER

- BB1B Teste si une touche est enfoncée.
 Carry = 1 si oui et A contient le caractère correspondant.
 Carry = 0 si non.
 A est altéré.
- BB18 Attente de la frappe d'une touche.
 Carry = 1 et A contient le caractère correspondant.
- BB24 Etat du joystick.
 A et H contiennent l'état de la manette 0.
 L contient l'état de la manette 1.
 Bit 0 : haut
 Bit 1 : bas
 Bit 2 : gauche
 Bit 3 : droite

Bit 4 : feu 2
Bit 5 : feu 1
Bit 6 : commande joystick supplémentaire
Bit 7 : non utilisé

BB21 Etat des touches CAPS LOCK et SHIFT.
 L SHIFT et H CAPS LOCK.
 00 si LOCK inactif, FF si actif.
 AF modifié.

ROUTINES CONCERNANT L'ECRAN TEXTE

BB5A Envoie au VDU un caractère ou un code de controle.
 A contient le caractère à envoyer.

BB5D Ecrit un caractère à la position courante du curseur.
 Déplace le curseur d'une position.
 A contient le caractère à envoyer.
 Tous les registres sont modifiés.

BB6D Lit le caractère à la position courante de l'écran.
 Si caractère valide, Carry = 1 et A contient caractère.
 Sinon, Carry = 0 et A contient 0.

BB75 Force la position du curseur.
 H doit contenir le numéro de colonne et L le numéro de ligne.
 AF et HL sont altérés.

BB78 Lit la position présente du curseur.
 H retourne le No de colonne, L le No de ligne.
 A contient le nombre de scrollings effectués.

BC14 Efface l'écran.
 Tous les registres sont modifiés.

BB6C Efface la fenetre courante.
 Tous les registres sont modifiés.

BB9C Passe en inversion vidéo en échangeant PEN et PAPER.
 AF et HL sont modifiés.

BC4A Inversion de deux couleurs sur un caractère.
 La position du caractère sur l'écran est fixée par:
 H numéro de colonne.
 L numéro de ligne.

- B contient une couleur encre, C l'autre couleur encre.
Tous les registres sont modifiés.
- BB4E Réinitialisation complète de l'écran Texte.
Tous les registres sont modifiés.
- BC4D Scrolling de l'écran (8 lignes de balayage, 1 ligne carac.).
B contient 0 si scrolling désiré vers le bas.
B contient <> 0 si scrolling voulu vers le haut.
Tous les registres sont altérés.
- BC05 Déroulement latéral de l'écran par modification d'offset.
HL contient le nouvel offset requis.
AF et HL sont altérés.
- BC50 Scrolling d'une fenetre.
B devra contenir 0 pour un scrolling vers le bas.
B contiendra <> 0 pour un scrolling vers le haut.
- BB66 Création d'une fenetre désignée par ses limites.
H colonne d'un coté, D colonne de l'autre.
L ligne d'un coté, E ligne de l'autre.
Tous les registres sont modifiés.
Le curseur est positionné en haut à gauche de la fenetre.
- BBB4 Change le numéro de fenetre en cours.
A contient le nouveau No à l'entrée.
A contient l'ancien No en sortie.
HL modifié.

ROUTINES CONCERNANT L'IMPRIMANTE

- BD2E Détermine si l'imprimante est prete.
Carry = 1 si BUSY
Carry = 0 si libre.
- BD2B Tente d'envoyer un caractère à l'imprimante.
Revient si elle est occupée trop longtemps.
A contient le caractère à envoyer (Bit 7 ignoré, hélas...)
Carry = 1 si tout s'est bien passé.
Carry = 0 sinon.
A modifié.

BD31 Envoi d'un caractère à l'imprimante.
ATTENTION, ne teste pas le BUSY...
A contient le caractère.
Carry = 1 au retour.
A modifié.

ROUTINES CONCERNANT LE GENERATEUR SONORE

BD34 Envoie une donnée au PSG.
A contient le No de registre auquel s'adresse la donnée.
C contient la donnée.
AF et BC sont modifiés.

BCB6 Arrête tous les sons immédiatement.
Carry = 1 si un son était en cours.
Carry = 0 s'il n'y avait pas de son en cours.
AF, BC, HL sont modifiés.

BCB9 Redémarre les sons interrompus par BCB6...
AF, BC, DE, IX modifiés.

ROUTINES DIVERSES

BD10 Initialise la valeur du compteur de temps.
Ce compteur est incrémenté tous les 1/300 èmes de sec.
La valeur à envoyer est mise dans DEHL.
D contient l'octet le plus significatif, L le moins.
AF est modifié.

BD19 Attente de synchro verticale.
Aucune condition, aucune modification...

QUELQUES VARIABLES SYSTEME

Il est parfois bon de pouvoir contrôler, au sein d'un programme BASIC (ou machine), l'état du système. Voici quelques bonnes adresses.

Les pointeurs sont sur 2 octets, les flags sur un seul...

AE7B HIMEM (le BASIC ne peut empiéter dessus).
 AE81 Début du BASIC.
 AE83 Fin du BASIC.
 AE85 Début des variables.

 E1C8 Mode écran (0,1,2).
 E1CA Offset début écran.
 E1CB Poids fort adresse début mémoire écran.

 B20C Fenetre actuelle.
 B285 Ligne curseur.
 B286 Colonne curseur.

 B4E9 Vitesse répétition touches.
 B4EA Délai répétition touches.
 B4F1 Entrée joystick.

Plusieurs zones mémoire sont réservées à des tables de valeurs ou à des tampons d'entrée.

CLAVIER B34C 240 Octets de tables de valeurs des touches.
 LE SON B551 à B7FF variables de gestion, formes d'enveloppes...
 CASSETTE B800 à B8DF Type de fichier, longueur, implantation...

 DRIVE En cas d'utilisation d'un drive (ou sur les systèmes équipés d'origine) la zone A700 à A8AF est réservée à sa gestion.

UN DESASSEMBLEUR

Suite à ce que vous venez de lire dans le chapitre précédent, vous voudriez bien voir ce qui se passe dans les zones mémoire occupées par le système et, au besoin, chercher à comprendre comment fonctionnent les routines en assembleur utilisées dans les programmes de ce livre ou dans la ROM.

Vous ne possédiez pas encore de désassembleur, en voici un. Il n'a pas de grandes prétentions: seulement celle de vous inciter à aller plus loin et certainement acquérir un logiciel de travail plus complet.

La ROM de l'AMSTRAD n'étant pas accessible directement, il faut une petite routine en langage machine pour actionner le mécanisme d'OVERLAY, ceci pour justifier l'option ROM/RAM proposée par le désassembleur.

MODE D'EMPLOI

L'adresse de début de désassemblage sera donnée en hexadécimal, le reste est explicite, les options disponibles étant rappelées à l'écran.

Le désassemblage s'effectue par 'pages', le passage de l'une à l'autre est obtenu par l'appui sur ESPACE.

Il est possible d'éditer le listing désassemblé sur imprimante, une 'basculer' imprimante ON ou OFF étant disponible.

RESTRICTION: les instructions sur IX et IY ne sont pas désassemblées, elles sont seulement listées.

Les mnémoniques sont contenus dans des lignes de DATA: il est important de veiller à ce qu'il n'y ait pas d'erreur lors de la saisie de ces lignes. Les virgules successives ont leur importance, ne les négligez pas.

```

10 REM *****
20 REM **      -DESASSEMBLEUR-          **
30 REM **
40 REM **      @-DUTERTRE EDDY          **
50 REM **
60 REM **      25/02/1985                **
70 REM **
80 REM *****
90 DATA NOP,"2LD BC,","LD (BC),A",INC BC
,INC B,DEC B,"1LD B,","RLCA,"EX AF,AF',"
ADD HL,BC"
100 DATA"LD A,(BC)",DEC BC,INC C,DEC C,"
1LD C,","RRCA,"3DJNZ ","2LD DE,","LD (DE)
,A",INC DE
110 DATA INC D,DEC D,"1LD D,","RLA,"3JR "
,"ADD HL,DE","LD A,(DE)",DEC DE,INC E,DE
C E
120 DATA"1LD E,","RRA,"3JR NZ,","2LD HL,"
,"4LD ( ),HL",INC HL,INC H,DEC H,"1LD
H,","DAA
130 DATA"3JR Z,","ADD HL,HL","7LD HL,(
)" ,DEC HL,INC L,DEC L,"1LD L,","CPL,"3J
R NC,","2LD SP,"
140 DATA"4LD ( ),A",INC SP,INC (HL),D
EC (HL),"1LD (HL),","SCF,"3JR C,","ADD HL
,SP","5LD A,( )" ,DEC SP
150 DATA INC A,DEC A,"1LD A,","CCF
160 DATA RET NZ,POP BC,"2JP NZ,","2JP "
,"2CALL NZ,","PUSH BC,"1ADD A,","RST 0,RET
Z,RET,"2JP Z,","2CALL Z,","2CALL " ,"1AD
C A,","RST 8,RET NC,POP DE,"2JP NC,","6OU
T ,A"
170 DATA"2CALL NC,","PUSH DE,"1SUB ",RST
16,RET C,EXX,"2JP C,","1IN A,","2CALL C,
","INST/IX,"1SBC A,","RST 24,RET PO,POP HL
,"2JP PO,","EX (SP),HL","2CALL PO,","PUSH
HL,"1AND ",RST 32
180 DATA RET PE,JP (HL),"2JP PE,","EX DE
,HL","2CALL PE,","1XOR ",RST 40,RET P,
POP AF,"2JP P,","DI,"2CALL P,","PUSH AF,"1
OR ",RST 48,RET M,"LD SP,HL","2JP M,","EI
190 DATA "2CALL M,","INST/IY,"1CP ",RST 5
6
200 DATA LDI,CPI,INI,OUTI,,,,,LDD,CPD,IN
D,OUTD,,,,,LDIR,CPIR,INIR,OTIR,,,,,LDDR,
CPDR,INDR,OTDR

```



```

210 DATA"IN B,(C)","OUT (C),B","SBC HL,B
C","4LD ( ),BC",NEG,RET N,IM 0,"LD I,
A","IN C,(C)","OUT (C),C","ADC HL,BC","7
LD BC,( )",,RETI,, "LD R,A","IN D,(C)"
,"OUT (C),D","SBC HL,DE","4LD ( ),DE"
220 REM ne pas effacer cette ligne
230 DATA B,C,D,E,H,L,(HL),A,B,C,D,E,H,L,
(HL),A
240 DATA RLC,RRC,RL,RR,SLA,SRA,,SRL,BIT,
RES,SET
250 DATA"ADD A,","ADC A,","SUB ","SBC A,
","AND ","XOR ","OR ","CP "
260 DATA&DF,&04,&AB,&C9,&07,&AB,&FC,&3A,
&00,&00,&32,&0E,&AB,&C9,&00,&00,&CD,&60,
&BB,&32,&17,&AB,&C9
270 MEMORY &9FFF: DIM R$(16),S$(11)
280 RESTORE 230:Q=0
290 FOR N=0 TO 15:READ R$(N):NEXT
300 FOR N=0 TO 10:READ S$(N):NEXT
310 FOR N=0 TO 7:READ T$(N):NEXT
320 FOR N=&AB00 TO &AB16:READ A:POKE N,A
:NEXT
330 MODE 1:WINDOW 1,40,1,20:WINDOW #1,1,
40,21,25:ORIGIN 0,64:DRAW 639,0,1
340 ORIGIN 0,46:DRAW 639,0,1:ORIGIN 144,
0:DRAW 0,64,1:LOCATE#1,1,2:PRINT#1,"SELE
CTION"
350 LOCATE #1,19,2:PRINT#1,"COMMANDES":L
OCATE #1,11,4:PRINT#1,"ESPACE=Suite N=
Nouv.adresse"
360 LOCATE #1,11,5:PRINT#1,"I=Imprimante
ON/";CHR$(24);"OFF";CHR$(24)
370 LOCATE #1,33,5:PRINT#1,"A=Arret"
380 CLS: PRINT"VOUS VOULEZ DESASSEMBLER
UNE ZONE DE:"
390 PRINT:PRINT"1-RAM":PRINT"2-ROM":PRIN
T:PRINT"VOTRE CHOIX ?"
400 G$=INKEY$:G$=UPPER$(G$):IF G$="" OR
(G$<>"1" AND G$<>"2" AND G$<>"I") THEN 4
00
410 IF G$="I" THEN GOSUB 1340:GOTO 400
420 PRINT:PRINT:R=VAL(G$):LOCATE #1,4,4:
IF R=1 THEN PRINT#1,"RAM" ELSE PRINT#1,"
ROM"
430 A=0:B$="":PRINT "ADRESSE DE DESASSEM
BLAGE:...."

```

```

440 FOR N=0 TO 3
450 A$=INKEY$:a$=UPPER$(a$):IF A$="" THE
N 450
460 IF A$="I" THEN GOSUB 1340:GOTO 450
470 IF A$>"9" AND A$<"A" THEN 450
480 IF A$>"F" OR A$<"0" THEN 450
490 LOCATE 26+N,9:PRINT A$:B$=B$+A$:NEXT
500 A=VAL("&"+B$):IF A<0 THEN A=65536+A
510 CLS:FOR N=1 TO 20
520 PRINT UPPER$(B$);"      ";
530 P=A:GOSUB 1290:C$=HEX$(D,2):GOSUB 67
0:GOSUB 680
540 LOCATE 27,N:FOR M=A TO A+I-1:P=M:GOS
UB 1290:PRINT HEX$(D,2);" ";:NEXT:PRINT
550 A=A+I:IF A>65535 THEN A=0
560 B$=HEX$(A,4):IF Q=0 THEN 580
570 FOR M=1 TO 40:LOCATE M,N:CALL &AB10:
PRINT#8,CHR$(PEEK(&AB17));:NEXT M:PRINT#
8
580 NEXT N
590 T$=INKEY$:T$=UPPER$(T$):IF T$="" THE
N 590
600 IF T$=" " THEN B$=HEX$(A,4):GOTO 510

610 IF T$="I" THEN GOSUB 1340:GOTO 590
620 IF T$="N" THEN 380
630 IF T$="A"THEN WINDOW 1,40,1,25:CLS:E
ND
640 GOTO 590
650 DATA IM 1,"LD A,I","IN E,(C)","OUT (
C),E","ADC HL,DE","7LD DE,( )",,,IM 2
,"LD A,R","IN H,(C)","OUT (C),H","SBC HL
,HL","4LD ( ),HL",,,,RRD,"IN L,(C)","
OUT (C),L","ADC HL,HL","7LD DE,( )",,
,,RLD,,, "SBC HL,SP"
660 DATA "4LD ( ),SP",,,,,,"IN A,(C)",
"OUT (C),A","ADC HL,SP","7LD SP,( )"
670 Y=VAL("&"+MID$(C$,1,1)):Z=VAL("&"+MI
D$(C$,2,1)):RETURN
680 IF Y>3 AND Y<8 THEN 770
690 IF D=203 THEN P=A+1:GOSUB 1290:GOTO
820
700 IF D=237 THEN P=A+1:GOSUB 1290:GOTO
1060
710 IF D=&DD THEN PRINT"INST/IX ":"GOTO
1360

```

```

720 IF D=&FD THEN PRINT"INST/IY :":GOTO
1360
730 IF D>127 AND D<192 THEN 880
740 IF D<64 THEN 890
750 IF D>191 THEN 1040
760 PRINT:I=1:RETURN
770 IF Z=6 AND Y=7 THEN PRINT"HALT":I=1:
RETURN
780 PRINT"LD ";K=(Y-4)*2
790 IF Z<8 THEN PRINT R$(K);",";
800 IF Z>7 THEN PRINT R$(K+1);",";
810 PRINT R$(Z):I=1:RETURN
820 IF D>63 THEN 840
830 K=INT(D/8):PRINT S$(K);" ";C#=HEX$(
d,2):GOSUB 670:PRINT R$(Z):I=2:RETURN
840 k=INT(d/64)+7:PRINT s$(k);
850 k=INT(d/8)-8:IF d>191 THEN k=k-16
860 IF d>127 AND d<192 THEN k=k-8
870 PRINT k;",";C#=HEX$(D,2):GOSUB 670:
PRINT r$(z):i=2:RETURN
880 k=d-128:w=INT(k/8):PRINT t$(w);:GOSU
B 670:PRINT r$(z):i=1:RETURN
890 p=INT(d/10)
900 GOSUB 1170:FOR g=0 TO (d-p*10):READ
e$:NEXT g
910 f$=MID$(e$,1,1):IF f$>"9" THEN PRINT
e$:i=1:RETURN
920 PRINT MID$(e$,2);
930 IF f$="1" THEN P=A+1:GOSUB 1290:PRIN
T HEX$(d,2):i=2:RETURN
940 IF F$="2"THEN P=A+2:GOSUB 1290:PRINT
HEX$(D,2);:P=A+1:GOSUB 1290:PRINT HEX$(
D,2):I=3:RETURN
950 IF f$<>"3" THEN 990
960 P=A+1:GOSUB 1290:PRINT HEX$(d,2);" (
";
970 IF d>127 THEN d=d-256
980 w=a+2+d:b#=HEX$(w,4):PRINT b$;")":i=
2:RETURN
990 IF f$="4" THEN LOCATE 13,n:f$="2":GO
TO 940
1000 IF f$="5" THEN LOCATE 15,n:f$="2":G
OTO 940
1010 IF f$="6" THEN LOCATE 13,n:f$="1":G
OTO 930
1020 IF f$="7" THEN LOCATE 16,n:f$="2":G
OTO 940

```

```

1030 PRINT:i=1:RETURN
1040 p=INT((d-192)/20)+7
1050 GOSUB 1170:FOR g=0 TO (d-192-(p-7)*
20):READ e$:NEXT g:GOTO 910
1060 IF d>&BB THEN RETURN
1065 IF d<124 THEN 1090
1070 RESTORE 200
1080 FOR j=0 TO d-160:READ e$:NEXT:PRINT
e$:i=2:RETURN
1090 IF d>85 THEN 1120
1100 RESTORE 210
1110 FOR j=0 TO d-64:READ e$:NEXT:T=A:A=
A+1:GOSUB 910:A=T:i=i+1:RETURN
1120 IF d>114 THEN 1150
1130 RESTORE 650
1140 d=d-22:GOTO 1110
1150 RESTORE 660
1160 d=d-51:GOTO 1110
1170 ON p+1 GOTO 1180,1190,1200,1210,122
0,1230,1240,1250,1260,1270,1280
1180 RESTORE 90:RETURN
1190 RESTORE 100:RETURN
1200 RESTORE 110:RETURN
1210 RESTORE 120:RETURN
1220 RESTORE 130:RETURN
1230 RESTORE 140:RETURN
1240 RESTORE 150:RETURN
1250 RESTORE 160:RETURN
1260 RESTORE 170:RETURN
1270 RESTORE 180:RETURN
1280 RESTORE 190:RETURN
1290 IF R=1 THEN D=PEEK(P):RETURN
1300 PH=INT(P/256):PL=P-PH*256
1310 POKE &AB08,PL:POKE &AB09,PH
1320 CALL &AB00
1330 D=PEEK(&AB0E):RETURN
1340 IF Q=0 THEN Q=1:LOCATE#1,24,5:PRINT
#1,CHR$(24);"ON";CHR$(24);"/OFF":RETURN
1350 Q=0:LOCATE#1,24,5:PRINT#1,"ON/";CHR
$(24);"OFF";CHR$(24):RETURN
1360 P=A+1:GOSUB 1290
1370 IF D=9 OR D=&19 OR D=&29 OR D=&39 O
R D=&2B OR D=&E3 OR D=&23 OR D=&E9 OR D=
&F9 OR D=&E1 OR D=&E5 THEN I=2:RETURN
1380 IF D=&CB OR D=&36 OR D=&22 OR D=&2A
OR D=&21 THEN I=4:RETURN
1390 I=3:RETURN

```

RAPPELS SUR LE LOCATOR

Le procédé de découpage de la surface du globe en "rectangles", permettant la localisation aisée d'une station, est appelé "LOCATOR". Ce découpage est très utilisé pour le trafic en VHF / UHF et par satellites. Il est bon d'en connaître le principe afin de bien comprendre le fonctionnement des différents programmes s'y rapportant.

Le codage du LOCATOR s'effectue maintenant (un autre code a existé) sur six caractères, lettres et chiffres.

D'un coup de baguette magique, transformons la terre en une bande rectangulaire, traversée par l'équateur. Cette bande est partagée en 18 gros rectangles eux-mêmes divisés en 18 autres rectangles d'une "surface" de 20 degrés sur 10. Partageons ces derniers en 100 de manière à obtenir des rectangles de 2 degrés sur 1. Enfin, pour affiner la résolution et permettre une localisation assez précise, partageons les en surfaces élémentaires de 5 minutes sur 2,5. Il faut maintenant donner un repère à ces rectangles. Nous utiliserons des lettres et des chiffres ; vous l'avez deviné, c'est le fameux LOCATOR.

De -180 (longitude ouest de Greenwich) à +180 en allant vers l'est on a les 18 lettres, de A à R par "pas" de 20 degrés. De -90 (latitude sud) à +90 en montant vers le nord on a 18 lettres de A à R par "pas" de 10 degrés. N'est-ce pas là un découpage fort simple ? Nous avons obtenu les deux premiers caractères du LOCATOR. Les 100 rectangles de 2*1 degrés sont, eux, repérés par des chiffres, de 0 à 9 d'ouest en est et du sud au nord. Ils forment les troisième et quatrième caractères du LOCATOR. Les 576 surfaces élémentaires de 5*2,5 minutes sont repérées par des lettres de A à X d'ouest en est et du sud au nord, formant les deux derniers caractères du LOCATOR.

Nous allons voir comment, grâce à ce simple et savant découpage, il est possible de convertir une position géographique quelconque du globe (latitude et longitude) en LOCATOR.

Notre premier petit programme sera "didactique" et il est possible de supprimer les PRINT des lignes 130, 150, 170, 190, 220, 330, 350, 370, 390 et 420 qui n'ont pas d'autre but que d'illustrer le procédé de conversion.

Gc désigne les "grands carrés", mc les "moyens" et pc les "petits".

En longitude, l'origine est à -180. On effectuera donc un "changement de repère"... De même avec la latitude commençant à -90...

Si vous avez bien compris le découpage précédant, le programme ne doit vous poser aucun problème. La conversion des valeurs de "coordonnées transposées" en lettres et chiffres s'effectue à partir du code ASCII sachant que la lettre A a pour code 65 et le chiffre 0 48. On obtient les variables "CAR1" à "CAR6" représentant les 6 caractères du LOCATOR. On les concatène en une chaîne qui forme le code du LOCATOR (QTH\$).

Ce premier petit programme pourra facilement être inclus, si vous le désirez, dans le programme de calculs de distances et azimuts. Il a sa raison d'être car on ne connaît pas forcément le LOCATOR de l'endroit où l'on se trouve (par exemple en vacances) alors que la position géographique repérée en coordonnées est indiquée sur toutes les cartes.

Ceci nous amène à un second petit programme permettant d'effectuer les conversions ramenant les coordonnées en degrés décimaux... pour les introduire dans le programme précédant. En effet, certaines cartes indiquent les coordonnées en grades, d'autres en degrés sexagésimaux. Un petit exercice de manipulations de chaînes et de conversions nous permet d'obtenir dans LA et GA respectivement la latitude et la longitude au "format" requis.

Ce second "module" pourrait, bien entendu, être inclus au premier... et l'ensemble au programme LOCATOR. C'est une affaire de goût ou de besoins. Notez, si vous désirez marier ces trois programmes, que les variables ont été prévues pour pouvoir être transmises: QTH\$, LA et GA.

Le programme final est, bien sûr, le calcul des distance et azimut par le LOCATOR. On entre le LOCATOR du correspondant et la machine calcule la distance qui vous sépare de lui et la direction dans laquelle vous devez tourner vos antennes pour assurer la meilleure liaison (s'il n'y a pas d'obstacles ou de réflexions...).

Le principe du programme est très simple : on découpe la chaîne formant le LOCATOR caractère par caractère et on reconvertit en coordonnées... Nous avons vu, plus haut, le procédé inverse.

Au cœur de ce programme se trouve une formule complexe mais bien connue de tous les amateurs en navigation. Elle permet de connaître la distance séparant deux points du globe de coordonnées LA GA et LB GB, qui est en fait la longueur de l'arc de grand cercle, ayant pour centre celui de la terre, et passant par ces deux points. Cette formule de trigo sphérique est suffisamment précise si la route est courte.

Rappelons, qu'à l'équateur, un angle de 1 degré correspond à environ 111 kilomètres (60 miles nautiques). 1 mile nautique égale une minute d'angle.

DISTANCE = ARCOS (sin(LA)sin(LB)+cos(LA)cos(LB)cos(GB-GA))*60

AZIMUT = ARCOS (sin(LB)-cos(D/60)sin(LA))/(sin(d/60)cos(LA))
(avec D représentant la distance calculée ci-dessus)

La distance est exprimée en kilomètres et l'azimut en degrés.

En ligne 1840, le programme reconstitue en ACSX l'arc cosinus (ARCOS de la formule) absent dans le BASIC de l'AMSTRAD.

Un test, portant sur la validité du LOCATOR introduit, permet de détecter des erreurs de frappe éventuelles.

Comme on ne peut préjuger de l'état de la touche CAPS LOCK, on transforme systématiquement et grâce à la fonction UPPER\$, la chaîne LOCATOR introduite en majuscules.

Notez le WHILE INKEY\$="":WEND servant à attendre l'appui sur une touche.

Le module de calcul de distance, situé en fin de programme, est le même que celui utilisé dans le programme CONTEST.

Le sous-programme testant la validité du LOCATOR a été placé en début du programme. En règle générale, c'est une habitude à prendre car, à l'appel d'un sous-programme, le BASIC effectue une recherche en partant du début du programme. On voit donc tout l'intérêt (gain de vitesse) à placer en tête les sous-programmes appelés fréquemment.

```
1000 '*****
1010 ' * *
1020 ' * LOCATOR *
1030 ' * *
1040 ' * F1EZH F6GKQ *
1050 ' * *
1060 ' * (C) EDEN Software 85 *
1070 ' * *
1080 '*****
1090 :
1100 GOTO 1170
1110 IF C(I)<65 OR C(I)>82 THEN ER=1
1120 RETURN
1130 IF C(I)<48 OR C(I)>57 THEN ER=1
1140 RETURN
1150 IF C(I)<65 OR C(I)>88 THEN ER=1
1160 RETURN
1170 'PRESENTATION
1180 MODE 1:INK 0,1:PAPER 0:INK 1,24:PEN
1 :BORDER 1:WINDOW #1,8,31,7,17:INK 3,8:
PAPER #1,3:CLS#1:INK 2,0:PEN #1,2
```

```

1190 PRINT #1:PRINT #1,"          LOCATOR"
1200 PRINT #1:PRINT #1,"  CALCUL  DISTANCE
/AZIMUT"
1210 PRINT #1:PRINT #1,"          F1EZH ET F6
GKQ"
1220 PRINT #1:PRINT #1,"  ";CHR$(164);"  E
DEN Software 1985"
1230 PRINT #1:PRINT #1,"          CPC 464 et
664"
1240 LOCATE #0,7,21:PRINT "< APPUYEZ SUR
UNE TOUCHE >"
1250 WHILE INKEY$="":WEND
1260 MODE 1:INK 0,13:PAPER 0:CLS:INK 1,0
:PEN 1:BORDER 13
1270 A$="####.####":B$="####.##"
1280 CLS
1290 PRINT:PRINT:PRINT"Pour initialiser
les calculs entrez : ":PRINT
1300 PRINT "(L)  Votre LOCATOR
          (C)  Vos COORDONNEES"
1310 PRINT:INPUT "          Votre choix L/C "
;cx$
1320 cx$=UPPER$(cx$):IF cx$<>"L" AND cx$
<>"C" THEN 1280
1330 IF cx$="L" THEN GOSUB 1400 ELSE GOS
UB 1360
1340 PRINT:PRINT:PRINT "< PRESSEZ UNE TO
UCHE POUR CONTINUER >":WHILE INKEY$="":W
END
1350 la=lat:ga=lon:GOTO 1650
1360 CLS:PRINT:PRINT"Introduisez
vos COORDONNEES          en degres DE
CIMAUX"
1370 PRINT:PRINT: INPUT "LATITUDE ";lat
1380 INPUT "LONGITUDE ";lon
1390 RETURN
1400 'INTRODUCTION DU LOCATOR
1410 CLS
1420 PRINT:PRINT:PRINT
1430 PRINT:PRINT:INPUT "Entrez le LOCATO
R ";qth$
1440 qth$=UPPER$(qth$)
1450 IF LEN(QTH$)=6 THEN 1470
1460 PRINT CHR$(7):PRINT:PRINT "ERREUR S
UR LA LONGUEUR DU LOCATOR ":GOTO 1570

```



```

1470 'DECODE ET TESTE VALIDITE
1480 FOR i=1 TO 6:c(i)=ASC (MID$(qth$,i
,1)): ER=0
1490 ON i GOSUB 1110,1110,1130,1130,1150
,1150
1500 IF ER=0 THEN NEXT:GOTO 1580
1510 'INDIQUE OU EST ERREUR
1520 C$=MID$(QTH$,I,1)
1530 PRINT CHR$(7)
1540 PRINT:PRINT "ERREUR SUR";
1550 IF I=1 THEN PRINT " 1er"; ELSE PRIN
T i;"eme";
1560 PRINT " CARACTERE LOCATOR (";C$;")"

1570 BORDER 0,26:PRINT:PRINT:PRINT "< PR
ESSEZ UNE TOUCHE POUR CONTINUER >":WHILE
  INKEY$="":WEND:BORDER 13:GOTO 1400
1580 'CALCUL LATITUDE ET LONGITUDE
1590 LON=-180+(C(1)-65)*20+(C(3)-48)*2+(
C(5)-64.5)/12
1600 LAT=-90+(C(2)-65)*10+C(4)-48+(C(6)-
64.5)/24
1610 PRINT:PRINT
1620 PRINT "Latitude   : ";USING A$;LAT
1630 PRINT "Longitude  : ";USING A$;LON
1640 RETURN
1650 CLS
1660 PRINT:PRINT "Calculs de DISTANCE et
d'AZIMUT"
1670 PRINT "-----"
----"
1680 PRINT:PRINT
1690 GOSUB 1420
1700 lb=lat:gb=lon:dg=ga-gb
1710 a=SIN(la/180*PI)
1720 b=SIN(lb/180*PI)
1730 c=COS(la/180*PI)
1740 d=COS(lb/180*PI)
1750 e=COS(dg/180*PI)
1760 x=(a*b)+(c*d*e)
1770 acsx=-ATN(x/SQR(-x*x+1))+1.5708
1780 dist=111.323*(acsx/PI*180)
1790 PRINT:PRINT:PRINT "DISTANCE   : ";US
ING B$;DIST
1800 DIST=DIST/1.852:R=DIST/60
1810 F=COS(R/180*PI)

```

```

1820 G=SIN(R/180*PI)
1830 X=(B-F*A)/(G*C)
1840 ACSX=-ATN(X/SQR(-X*X+1))+1.5708
1850 AZI=ACSX/PI*180
1860 IF GA-GB>0 THEN AZI=360-AZI
1870 PRINT"AZIMUT      : ";USING B#;AZI
1880 PRINT:PRINT:PRINT "< PRESSEZ UNE TO
UCHE POUR CONTINUER >":WHILE INKEY#="":W
END
1890 GOTO 1650

```

```

100 'Module de conversion coordonnees
110 'Ressort avec LAT en LA
120 '          LON en GA
130 '
140 '
200 k1=100/60:k2=90/100
210 MODE 2
220 PRINT:PRINT "Entree des coordonnees
geographiques"
230 PRINT:PRINT"      -1 En Degres Sexagesi
maux"
240 PRINT"      -2 En Grades"
250 PRINT"      -3 En Degres Decimaux"
260 PRINT:INPUT "      -> Votre choix ";cx

270 PRINT:PRINT
280 ON cx GOSUB 320,450,510
290 PRINT:PRINT:PRINT la:PRINT ga
300 PRINT:PRINT "Appuyez sur une touche
pour recommencer"
310 WHILE INKEY#="":WEND:RUN
320 INPUT "Latitude dd.mm ";lat$
330 INPUT "Longitude dd.mm ";lon$
340 cor$=lat$:GOSUB 380:la=dg*s
350 cor$=lon$:GOSUB 380:ga=dg*s
360 RETURN
370 '----- conversion sexa->decimal -----
--
380 IF LEFT$(cor$,1)="--"THEN s=-1 ELSE s
=1
390 IF s=-1 THEN cor$=MID$(cor$,2,5)

```

```

400 dg$=LEFT$(cor$,2)
410 mm$=RIGHT$(cor$,2)
420 mm=VAL(mm$)*k1*0.01
430 dg=VAL(dg$)+mm
440 RETURN
450 '----- conversion grades->degrees ----
---
460 INPUT "Latitude (grades) ";lat$
470 INPUT "Longitude (grades) ";lon$
480 la=VAL(lat$)*k2
490 ga=VAL(lon$)*k2
500 RETURN
510 '----- entree decimale directe -----
520 INPUT "Latitude (degrees decimaux) "
;la
530 INPUT "Longitude (degrees decimaux) "
;ga
540 RETURN

10 'Decomposition des coordonnees
20 'Geographiques en QTH Locator
30 '
50 INPUT "Latitude ";la
60 INPUT "Longitude ";ga
100 'Determine carres en LONGITUDE
110 '
120 ga=(ga+180)/20
130 PRINT ga
140 gc1=INT(ga)
150 PRINT gc1
160 mc1=(ga-gc1)*10
170 PRINT mc1
180 pc1=(mc1-INT(mc1))*24
190 PRINT pc1
200 car1=65+gc1;car3=48+INT(mc1);car5=65
+INT(pc1)
220 PRINT CHR$(car1);CHR$(car3);CHR$(car
5)
300 'Determine carres en LATITUDE
310 '
320 la=(la+90)/10
330 PRINT la
340 gc2=INT(la)
350 PRINT gc2
360 mc2=(la-gc2)*10

```

```
370 PRINT mc2
380 pc2=(mc2-INT(mc2))*24
390 PRINT pc2
400 car2=65+gc2:car4=48+INT(mc2):car6=65
+INT(pc2)
420 PRINT CHR$(car2);CHR$(car4);CHR$(car
6)
500 'Assemble les caracteres LOCATOR
510 '
520 qth$=CHR$(car1)+CHR$(car2)+CHR$(car3
)+CHR$(car4)+CHR$(car5)+CHR$(car6)
530 PRINT"LOCATOR ";qth$
```

CONTEST

Le suivi d'un concours VHF/UHF par ordinateur est une des applications les plus courantes de la micro-informatique : elle permet de soulager l'opérateur de tâches répétitives, telles que la recherche des doubles, le calcul des distances (et celui des points dans le cas de l'utilisation d'un multiplicateur).

Le rôle du logiciel sera de :

- Tenir à jour la feuille de concours (pour les possesseurs d'une imprimante tout particulièrement).
- Eviter une liaison "double" en prévenant l'opérateur qu'elle a déjà été effectuée.
- Calculer les distances correspondantes par le procédé du LOCATOR.
- Informer l'opérateur des performances déjà réalisées : meilleure liaison (DX), moyenne kilométrique par contacts établis, numéro du prochain contact pour la constitution du groupe de contrôle passé au correspondant.
- Afficher l'heure automatiquement, en datant les liaisons sans intervention de l'opérateur. Ce paramètre horaire n'étant plus passé par les opérateurs mais demeurant indispensable sur le compte-rendu pour le contrôle de validité des liaisons établies.
- Permettre à tout moment l'interruption du concours pour prendre quelque repos "bien mérité", sans avoir à laisser la machine sous tension. Cette fonction est réalisée par l'écriture de fichiers sur la disquette (ou sur la cassette après une petite adaptation du logiciel).

LES CARACTERISTIQUES DU PROGRAMME.

Le travail s'effectue sur écran et le mode 80 colonnes a bien sur été retenu compte-tenu des possibilités de l'AMSTRAD. Nous avons profité des "fenêtres" de l'AMSTRAD pour réaliser une présentation agréable.

L'utilisation d'une disquette est fortement souhaitable bien que la cassette puisse être utilisée, moyennant une modification des lignes contenant la sauvegarde et la relecture des fichiers.

De meme, si l'utilisation d'une imprimante est également souhaitable, elle n'est pas nécessaire... sauf pour rédiger le compte-rendu final. Cette tache pourra etre accomplie avec un bon vieux stylo si l'imprimante ne tient pas compagnie à votre AMSTRAD.

500 QSO peuvent etre mémorisés dans la machine, ce qui représente une participation plus qu'honorable.

UTILISATION DU PROGRAMME.

Après chargement, l'ordinateur demande le LOCATOR de la station effectuant le concours ainsi que l'indicatif.

Si des liaisons avaient été réalisées auparavant, le chargement des fichiers correspondants sera réalisé. On entre alors dans la boucle principale du programme...

Un menu est affiché à l'écran, après une remise à l'heure de la pendule. La machine attend alors l'entrée d'un indicatif ou de l'une des 3 options:

- S pour sauvegarder les fichiers.
- E pour lister les liaisons établies sur écran.
- P pour lister les liaisons sur imprimante.

Si vous entrez un indicatif, la machine va vérifier que la liaison n'a pas déjà été établie... Cette vérification effectuée, vous serez autorisé (ou non) à réaliser le contact. Vous devrez alors introduire à nouveau l'indicatif puis les différents paramètres du "QSO". La machine vous demandera enfin si vous validez la liaison. Dans ce cas, elle est prise en compte et les totaux sont réactualisés...sinon, il est toujours possible d'annuler la liaison.

E ou P permettent de lister tout ou partie des "QSO" établis. La machine vous demande à partir de quel numéro et jusqu'à quel numéro doit se faire le listage. A son issue, on revient dans la boucle principale du programme.

S vous permet de sauvegarder les données déjà en machine, pour interrompre le concours... ou par simple précaution prévenant une coupure de courant !

Dans le cas d'une sauvegarde, les opérations d'entrée/sortie inhibant les interruptions indispensables au fonctionnement de la pendule, il sera nécessaire de le remettre à l'heure comme vous y invitera le programme.

Le sous-programme de calcul des distances est extrait du programme LOCATOR et, grace aux fonctions DELETE et RENUM, si vous etes un peu astucieux, vous ne le taperez qu'une seule fois.

Construit en "petits modules", ce logiciel pourra etre adapté à vos besoins, les seuls commentaires particuliers portant sur la structure des fichiers. Le fichier des indicatifs contient... les indicatifs des stations déjà contactées, celui des données contient la "concaténation" des différents paramètres de chaque liaison, formant ainsi autant de chaines de caractères qu'il y a d'indicatifs. Ce procédé a été retenu pour établir un compromis entre

les temps de chargement et sauvegarde, l'occupation mémoire, et la simplification de la mise en page sur écran.

Ne pouvant préjuger de la longueur des fichiers, fonction du nombre de contacts déjà réalisés, lors du chargement, on testera l'indicateur de fin de fichier (EOF), d'ou le role de la ligne :

```
WHILE NOT EOF:INPUT #9 etc.
```

Un dernier conseil : attention au rangement et à la protection de votre disquette ou cassette fichiers. Si vous avez un quelconque doute sur le risque de coupures de secteur, deux sauvegardes valent mieux qu'une !

MODULE CALCUL ET AFFICHAGE DE L'HEURE

Nous avons inclus dans le programme CONTEST un module permettant d'obtenir l'heure en permanence, ce qui permet de dater les liaisons établies. Ce sous-programme 'PENDULE' est controlé par les interruptions système.

Il existe plusieurs types d'interruptions sur les CPC. Nous avons utilisé ici les évènements asynchrones et plus exactement le FRAME FLYBACK c'est-à-dire le retour du spot de balayage écran. Cette interruption intervient tous les 1/50èmes de seconde. Les équivalents 'soft' de ces interruptions sont appelés sur le système CPC 'EVENTS'. Leur manipulation est un peu délicate et nous vous renvoyons aux divers documents déjà publiés sur cette machine pour en avoir une description complète sortant du cadre de ce livre.

A chaque 'EVENT' est associée une zone mémoire, utilisée par le système, que le programmeur doit définir. Le registre double HL contient l'adresse de cette zone mémoire. La routine qui sera exécutée à chaque interruption a son adresse contenue dans DE. C contient une valeur fonction de l'adresse mémoire mémoire de la routine d'évènement (selon qu'elle est en RAM, ROM, ROM externe.). Le registre B contient 'la classe' définissant le type d'évènement.

Pour plus d'informations sur l'utilisation de cette particularité des CPC nous vous suggérons de consulter le 'FIRMWARE' (Soft 158 chez AMSOFT).

LISTING ASSEMBLEUR COMMENTE DU SOUS-PROGRAMME HORLOGE

```

;QTRD10 LE 16/09/85          CE LISTING NE DOIT PAS ETRE SAISI
;
TOP: EQU 0A500H              Définit l'adresse basse des compteurs de l'horloge
EVENT: EQU 0A506H           Adresse de la zone mémoire réservée au système
FRAME: EQU 0B0D7H           Routine FRAME FLYBACK
WRCHR: EQU 0BB5DH           Ecriture d'un caractère sur l'écran
SETSTR: EQU 0BBB4H          Création d'une fenetre
SETCUR: EQU 0BB75H          Force la position du curseur
GETCUR: EQU 0BB78H          Lit la position présente du curseur
;
;
DEFS 0FH                    Réserve pour compteurs et routine système
ORG 0A510H
LOAD 0A510H
NEWFRA: LD HL,EVENT         Initialisation du bloc EVENT pour prévenir le
LD B,81H                    système
LD C,0
LD DE,FLYTOC               FLYTOC est la routine d'interruption
CALL FRAME
RET
FLYTOC: LD HL,TOP          Top toutes les 20 ms
DEC (HL)                   Si TOP non nul, on sort de la routine sans modifier
JR NZ,FINAL                l'heure affichée...
;SECONDES
LD A,50                    Remet TOP à sa valeur initiale
LD (HL),A
CALL PLUS1                 Incrémente le pointeur pour lire les secondes
CP 60                      Si <>60 on sort
JR NZ,FINAL
LD A,0                    Remet secondes à zéro
LD (HL),A
CALL PLUS1                 ;UNI. MINUTES Va lire le compteur des minutes
CP 03AH                   Test de dépassement (>9)
JR NZ,FIN
CALL PLUS                 ;DIZ. MINUTES Va lire le compteur des dizaines de minutes
CP 36H                    Test de dépassement (>6) (Minutes >59)
JR NZ,FIN
LD A,30H                  Remet à zéro si c'est le cas
LD (HL),A                 ;RAZ DIZ. MIN.
;HEURES TEST +23 ?

```


INC HL	
LD A,(HL)	
CP 33H	
JR NZ,NEXT2	Teste pour savoir si on a atteint 23H59
INC HL	
LD A,(HL)	
CP 32H	
JR NZ,NEXT1	
	;24H RAZ HEURES
LD A,30H	Remise à zéro dans ce cas
LD (HL),A	
DEC HL	
LD (HL),A	
JP Z,FIN	
	;INCREMENT HEURES
NEXT1: DEC HL	
NEXT2: INC (HL)	
LD A,(HL)	Incrémentation des heures sinon.
CP 03AH	
JR NZ,FIN	
CALL PLUS	
	Ici commence la routine d'affichage
FIN: CALL GETCUR	Quelle est la position présente du curseur ?
PUSH HL	On la sauvegarde.
LD A,04	On va afficher l'heure dans la fenetre No 4
CALL SETSTR	Sélection fenetre et sauvegarde ancien
PUSH AF	No de fenetre (là ou était le curseur)
LD H,1	Positionne le curseur dans la fenetre d'affichage
LD L,1	réservée à l'heure
CALL SETCUR	
LD HL,EVENT	Pointe sur la zone mémoire contenant les caractères
CALL ECRIT	formant l'heure (à partir de &A506) et les affiche
CALL ECRIT	
LD A,3AH	séparant heures et minutes par le signe ':'
CALL ECRIT2	
CALL ECRIT	
CALL ECRIT	
	; REMET LE CURSEUR A SA PLACE
POP AF	Récupère l'ancien No de fenetre
CALL SETSTR	et le rétablit
POP HL	Récupère l'ancienne position du curseur
CALL SETCUR	et le remet à sa place
FINAL: RET	C'est fini... On revient continuer le programme.

S/P d'incrémentation des compteurs H,M, ou S

```

PLUS: LD A,30H          Remet à zéro le compteur inférieur
LD (HL),A
PLUS1: INC HL           Incrèmente le compteur suivant
INC (HL)
LD A,(HL)
RET

```

S/P d'écriture des différents caractères de l'heure

```

ECRIT: DEC HL          Il faut savoir que, par soucis de simplification
LD A,(HL)             on a utilisé directement les valeurs code ASCII
ECRIT2: PUSH HL       des caractères formant l'heure pour les afficher.
CALL WRCHR            Cela peut paraître curieux mais c'est bien plus
POP HL                simple à gérer...
RET
END

```

```

1000 ? *****
1010 ? *
1020 ? *          CONTEST          *
1030 ? *
1040 ? *          F1EZHF6GKQ      *
1050 ? *
1060 ? * (C) EDEN Software B5 *
1070 ? *
1080 ? *****
1090 GOSUB 10000: ' Implantation routine
QTR
1100 a$="#####":b$="####.####"
1103 DIM U$(500),V$(500)
1105 MODE 2:GOTO 2000
1500 ?----- ENTREE AVEC QTH$ SORTIE CI
NT(DIST) -----
1510 QTH$=UPPER$(QTH$):FOR i=1 TO 6:c(i)
=ASC (MID$ (qth$,i,1)):NEXT
1520 ?----- CALCUL LATITUDE ET LONGITU
DE -----
1530 LON=-180+(C(1)-65)*20+(C(3)-48)*2+(
C(5)-64.5)/12
1540 LAT=-90+(C(2)-65)*10+C(4)-48+(C(6)-
64.5)/24
1545 IF AA<>0 THEN 1560
1547 AA=1:LA=LAT:GA=LON:RETURN
1550 ?----- CALCUL DE LA DISTANCE ----

```

```

1560 lb=lat:gb=lon:dg=ga-gb
1570 a=SIN(1a/180*PI)
1580 b=SIN(1b/180*PI)
1590 c=COS(1a/180*PI)
1600 d=COS(1b/180*PI)
1610 e=COS(dg/180*PI)
1620 x=(a*b)+(c*d*e)
1630 acsx=-ATN(x/SQR(-x*x+1))+1.5708
1640 dist=111.323*(acsx/PI*180)
1650 DIST=CINT(DIST)
1660 RETURN
2000 '----- PRESENTATION -----
2010 MODE 1:INK 0,1:PAPER 0:INK 1,24:PEN
  1:BORDER 1:WINDOW #1,8,31,7,17:INK 3,8:
PAPER #1,3:CLS#1:INK 2,0:PEN #1,2
2020 PRINT #1:PRINT #1,"          CONTEST"
2030 PRINT #1:PRINT #1,"          SUIVI DE CONC
OURS"
2040 PRINT #1:PRINT #1,"          F1EZH ET F6
GKQ"
2050 PRINT #1:PRINT #1," ";CHR$(164);" E
DEN Software 1985"
2060 PRINT #1:PRINT #1,"          CPC 464 et
664"
2070 LOCATE #0,7,21:PRINT "< APPUYEZ SUR
UNE TOUCHE >"
2080 WHILE INKEY#="":WEND
2090 MODE 2:INK 0,13:PAPER 0:CLS:INK 1,0
:PEN 1:BORDER 13
2500 AA=0
2600 LOCATE 20,10:INPUT "Votre LOCATOR S
VP ";QTH$
2610 QRA$=QTH$:GOSUB 1500
2620 LOCATE 23,12:PRINT "Latitude  : ";U
SING b$;lat
2630 LOCATE 23,13:PRINT "Longitude : ";U
SING b$;lon
2650 LOCATE 20,15:INPUT "Votre INDICATIF
";NDI$:NDI$=UPPER$(NDI$)
2700 LOCATE 10,18:INPUT "Reprenez-vous l
e concours apres une interruption O/N ";
r$:r$=UPPER$(r$)
2710 IF r$<>"O" AND r$<>"N" THEN 2700
2720 IF r$="N" THEN 3000
2800 '----- RELECTURE DES FICHIERS ---
-----

```

```

2815 PRINT:PRINT "Preparez le lecteur et
le support magnetique pour la lecture d
es fichiers Appuyez ensuite sur un
e touche..."
2817 WHILE INKEY$="":WEND
2820 OPENIN "PARAM"
2830 INPUT #9,CUM,MOY,DER,DX
2840 CLOSEIN
2845 PRINT:PRINT"Lecture du fichier PARA
METRES terminee"
2850 OPENIN "INDIC":N=1
2860 WHILE NOT EOF:INPUT #9,U$(N):N=N+1:
WEND
2870 CLOSEIN
2875 PRINT:PRINT"Lecture du fichier INDI
CATIFS terminee"
2880 OPENIN "DATAS":N=1
2890 WHILE NOT EOF:INPUT #9,V$(N):N=N+1:
WEND
2900 CLOSEIN
2910 PRINT:PRINT"Lecture du fichier DONN
EES terminee"
2920 PRINT:PRINT"Rangez le support magne
tique et pressez une touche"
2930 WHILE INKEY$="":WEND
3000 '----- DEBUT ET MENU PRINCIPAL --
-----
3010 '----- REMISE A L'HEURE DE L'HORL
OGE -----
3020 CLS:PRINT:PRINT "Remise a l'heure d
e l'horloge "
3030 PRINT:PRINT "Entrez l'heure sous la
forme HHMM"
3035 PRINT:INPUT "Au top horaire, appuye
z sur RETURN ";QTR$
3040 IF LEN(QTR$)<>4 THEN 3020
3050 FOR n=1 TO 4:car=ASC (MID$(QTR$,N,1
))
3060 POKE &A506-N,CAR:NEXT:POKE &A500,50
:POKE &A501,0
3070 CALL &A510:' Relance l'horloge
3100 CLS:R$="...":IF DER=0 THEN DER=1
3105 WINDOW #1,1,80,1,13:PAPER #1,0:CLS
#1:PEN #1,1: 'Ecran travail
3110 WINDOW #2,5,30,19,24:PAPER #2,1:CLS
#2:PEN #2,0: 'Ecran options

```

```

3120 WINDOW #3,50,75,19,24:PAPER #3,1:CL
S #3:PEN #3,0:'Ecran suivi
3130 WINDOW #4,38,42,25,25:PAPER #4,1:CL
S #4:PEN #4,0:'Ecran QTR
3135 PRINT #4,CHR$(PEEK(&A505));CHR$(PEE
K(&A504));":";CHR$(PEEK(&A503));CHR$(PEE
K(&A502));
3140 WINDOW #5,5,75,14,14:PAPER #5,1:CLS
#5:PEN #5,0:'Ecran repere
3150 WINDOW #6,32,48,19,24:PAPER #6,0:CL
S #6:PEN #6,1:'Ecran entrees
3200 PRINT #2," Entrez INDICATIF ou : "
3210 PRINT #2
3220 PRINT #2," S Sauvegarde des donnees
"
3230 PRINT #2," E Listing sur Ecran"
3240 PRINT #2," P Listing sur Imprimante
"
4050 PRINT #5,"Indicatif";TAB(20)"QTR";T
AB(28)"LOC";TAB(34)"PASSE";TAB(41)"RECU"
;TAB(52)"DIST"
4100 FOR n=1 TO 12:PRINT #1:NEXT
4150 WHILE LEN(r$)<>1:CLS #6:PRINT #6
4160 CLS #3:PRINT #3
4165 PRINT #3," Prochain QSO No :";USING
a$;der
4170 PRINT #3," Cumul des Points:";USING
a$;cum
4175 PRINT #3," Distance moyenne:";USING
a$;moy
4180 PRINT #3," Meilleur DX :";USING
a$;dx
4200 INPUT #6,"Ind ";r$;R$=UPPER$(r$)
4250 IF r$<>"S" AND r$<>"E" AND r$<>"P"
THEN ind$=r$ ELSE GOTO 5000
4260 '----- RECHERCHE D'UN DOUBLE -----
-----
4270 FOR i=0 TO der
4275 IF ind$<> MID$(u$(i),1,LEN(ind$)) T
HEN NEXT:GOTO 4300
4280 CLS #6:PRINT #6:PRINT #6,"Deja cont
acte"
4285 PRINT #6,"QSO no ";i
4290 PRINT CHR$(7):WHILE INKEY$="" :WEND
:GOTO 4150

```

```

4300 PRINT #6,"QSO ";:PRINT CHR$(7):INPU
T #6,"O/N";rep$:rep$=UPPER$(rep$)
4310 IF rep$<>"O" THEN 4150
4320 CLS #6:PRINT #6,"Ind ";ind
$:ind$=UPPER$(ind$)
4500 INPUT #6,"Loc ";qth$:qth$=UPPER$(qt
h$)
4550 INPUT #6,"Mon ";gr$
4600 INPUT #6,"Son ";gp$
4605 QTR$=CHR$(PEEK(&A505))+CHR$(PEEK(&A
504))+": "+CHR$(PEEK(&A503))+CHR$(PEEK(&A
502))
4650 INPUT #6,"Valide (O/N) ";rep$:rep$=
UPPER$(rep$)
4700 IF rep$="N" THEN 4150
4710 GOSUB 1500:dt$=STR$(dist)
4750 v$(der)=qtr$+" "+qth$+" "+gp$+" "
+gr$+" "+dt$
4760 u$(der)=ind$
4800 PRINT #1,TAB(5)ind$;TAB(23)qtr$;TAB
(30)qth$;TAB(38)gp$;TAB(45)gr$;TAB(53)US
ING a$;dist
4850 cum=cum+dist:moy=cum/der:IF DIST>DX
THEN DX=DIST
4860 DER=DER+1:WEND
5000 '----- SAUVEGARDE DES FICHIERS ---
-----
5050 IF R$<>"S" THEN 6000
5060 CLS :PRINT:PRINT
5070 PRINT"PREPAREZ LE LECTEUR POUR LA S
AUVEGARDE DES DONNEES ET PRESSEZ UNE TOU
CHE"
5080 WHILE INKEY$="":WEND:PRINT CHR$(7)
5100 '----- SAUVEGARDE DU FICHER PARA
METRES -----
5110 OPENOUT "PARAM"
5120 PRINT #9,CUM,MOY,DER,DX
5130 CLOSEOUT
5140 PRINT:PRINT"SAUVEGARDE DU FICHER P
ARAMETRES EFFECTUEE"
5200 '----- SAUVEGARDE DU FICHER INDI
CATIFS -----
5210 OPENOUT "INDIC"
5220 FOR N=1 TO DER:PRINT #9,U$(N):NEXT
5230 CLOSEOUT

```

```

5240 PRINT:PRINT"SAUVEGARDE DU FICHER I
NDICATIFS EFFECTUEE"
5300 '----- SAUVEGARDE DU FICHER DES
DONNEES -----
5310 OPENOUT "DATAS"
5320 FOR N=1 TO DER:PRINT #9,V$(N):NEXT
5330 CLOSEOUT
5340 PRINT:PRINT"SAUVEGARDE DU FICHER D
E DONNEES EFFECTUEE"
5400 PRINT:PRINT"RANGEZ LE SUPPORT MAGNE
TIQUE EN LIEU SUR !"
5420 PRINT:PRINT:PRINT:PRINT"PRESSEZ UNE
TOUCHE POUR POURSUIVRE..."
5430 WHILE INKEY$="":WEND: PRINT CHR$(7)
5450 GOTO 3000
6000 '----- SORTIE DU LISTING SUR ECRA
N OU IMPRIMANTE -----
6050 IF R$<>"P" THEN 6150
6060 CN=8:GOTO 6160
6150 IF R$<>"E" THEN R$="...":GOTO 4150
6155 CN=1
6160 CLS:WINDOW #1,1,80,1,20:PAPER #1,0:
CLS #1:PEN #1,1
6200 WINDOW #2,1,80,21,25:PAPER #2,1:CLS
#2:PEN #2,0
6700 '----- SAISIE DE DONNEES DEBUT ET
FIN DE LISTAGE -----
6710 INPUT #2,"Numero de QSO pour le deb
ut du listage ";deb
6730 INPUT #2,"Numero de QSO pour la fin
du listage ";fin
6735 IF FIN>DER-1 THEN FIN=DER-1
6740 IF CN=1 THEN 6800 ELSE PRINT #2,"PR
EPAREZ L'IMPRIMANTE ET PRESSEZ UNE TOUCH
E"
6750 WHILE INKEY$="":WEND
6760 PRINT #8,"Station ";UPPER$(ndi$);TAB
B(32)"Locator ";UPPER$(QRA$)
6770 PRINT #8:PRINT #8
6780 PRINT #8,"Indicatif";TAB(21)"QTR";T
AB(31)"QTH";TAB(40)"Passe";TAB(48)"Recu"
;TAB(60)"Distance"
6785 PRINT #8,"=====";TAB(20)"====="
";TAB(30)"=====";TAB(40)"=====";TAB(48)
"=====";TAB(60)"====="
6790 PRINT #8:PRINT #8

```

```

6800 FOR n=deb TO fin
6810 dist=VAL(RIGHT$(v$(n),LEN(v$(n))-29
))
6820 qtr$=MID$(v$(n),1,5)
6830 qth$=MID$(v$(n),7,6)
6840 gp$=MID$(v$(n),15,5)
6850 gr$=MID$(v$(n),22,5)
6900 PRINT #(cn),u$(n);TAB(20);qtr$;TAB(
30);qth$;TAB(40);gp$;TAB(48);gr$;TAB(60)
;USING a$;dist
6950 NEXT
6960 CLS #2:PRINT:PRINT #2,"FIN DU LISTA
GE. PRESSEZ UNE TOUCHE POUR CONTINUER."
6970 PRINT CHR$(7):WHILE INKEY$="":WEND:
GOTO 3100
10000 '----- HORLOGE & AFFICHAGE -----
10010 MEMORY &A4FF:FOR N=&A510 TO &A59C:
READ D$:POKE N,VAL("&"+D$):NEXT
10020 RETURN

10050 DATA 21,06,A5,06,81,0E,00,11,1E,A5
10060 DATA CD,D7,BC,C9,21,00,A5,35,20,69
10070 DATA 3E,32,77,CD,91,A5,FE,3C,20,5F
10080 DATA 3E,00,77,CD,91,A5,FE,3A,20,28
10090 DATA CD,8E,A5,FE,36,20,21,3E,30,77
10100 DATA 23,7E,FE,33,20,0F,23,7E,FE,32
10110 DATA 20,08,3E,30,77,2B,77,CA,60,A5
10120 DATA 2B,34,7E,FE,3A,20,03,CD,8E,A5
10130 DATA CD,78,BB,E5,3E,04,CD,B4,BB,F5
10140 DATA 26,01,2E,01,CD,75,BB,21,06,A5
10150 DATA CD,95,A5,CD,95,A5,3E,3A,CD,97
10160 DATA A5,CD,95,A5,CD,95,A5,F1,CD,B4
10170 DATA BB,E1,CD,75,BB,C9,3E,30,77,23
10180 DATA 34,7E,C9,2B,7E,E5,CD,5D,BB,E1
10190 DATA C9

```


T182R/90	16:07	JM180D	55001	53020	270
F162R/70	16:08	JM93DA	57002	54034	185
G141D	16:09	JM93AF	55003	55055	136
F187/750	16:10	JM93AP	55004	55047	183
G1180	16:10	JM93AE	55005	57034	133
F1806/P	16:11	JM93BJ	55006	56043	186
F315/35	16:11	JM93DC	55007	53012	9
G180C/P	16:12	JM93JO	55008	57021	200
F1800/P50	16:13	JM93LM	55009	55067	113
T182R	16:13	JM93QO	57010	57045	188
C801A/P	16:15	T08000	53011	57054	330

APPUI SUR LA CLAVIERE, PRESSEZ UNE TOUCHE POUR CONTINUER.

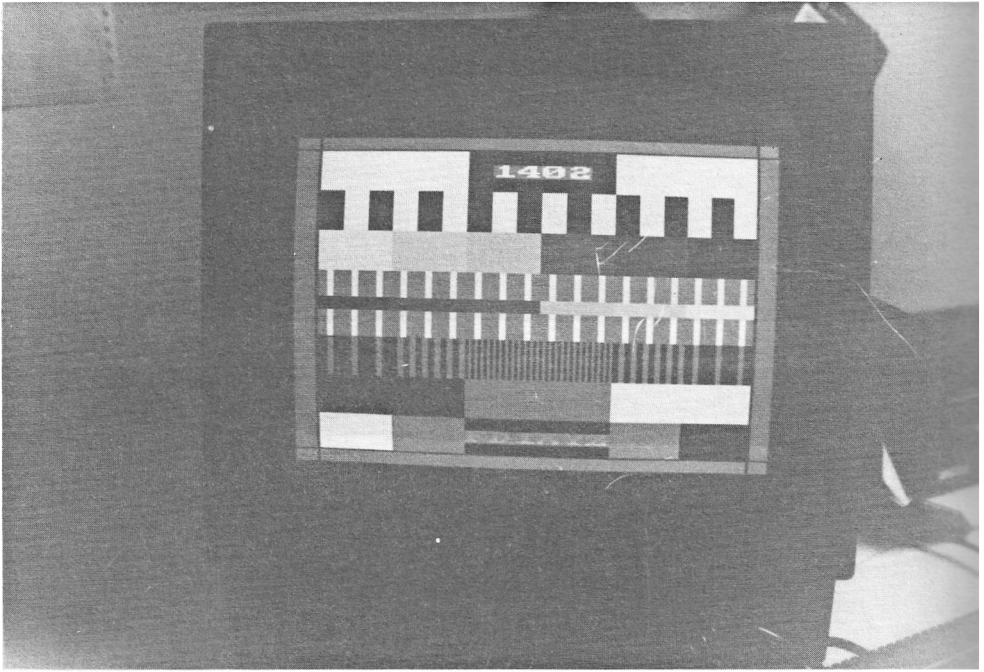
UTILISATEUR AIR LOC PASSE RECU DIS

Entrez l'INDICATEUR ou
 Sauvegarde des données
 Action sur l'écran
 Retiens sur l'imprimante

Ind → F8000

Prochaine 050 No 12
 Total des points 2014
 Distance Acquisse 183
 Milliers Km 330

16:00



MIRE TV

L'ordinateur va combler les passionnés de télévision. Est-il besoin de rappeler que les radioamateurs licenciés sont autorisés à pratiquer des émissions de télévision sur des fréquences qui leur sont réservées ? Le programme proposé pourra également servir au réglage de couleurs ou à quelques menues retouches sur un téléviseur ou un moniteur. Les amateurs de vidéo pourront, quant à eux, utiliser la mire pour leurs vidéo-cassettes... Attention, dans ce cas il sera nécessaire de faire appel aux services d'un codeur SECAM si on veut profiter des couleurs, à moins d'avoir la chance de posséder un magnétoscope muni d'une entrée RVB. Expliquons nous...

La sortie couleur de l'AMSTRAD se fait en R V B + Synchro. Il n'y a pas, à proprement parler, de codage couleur. Une sortie "luminance" permet le couplage sur un moniteur monochrome ou à une entrée vidéo quelconque (émetteur TV, magnétoscope etc.)... mais en noir et blanc. Pour obtenir des couleurs, il faut impérativement combiner les signaux R V B ou les utiliser directement sur une entrée du même nom (exemple : prise PERITEL).

Les couleurs produites par l'AMSTRAD sont de fort belle qualité. Elles ont été choisies pour rappeler celles que nous voyons sur les mires diffusées par le réseau national. L'ordre choisi permet d'obtenir un dégradé de gris sur un écran monochrome.

Outre la génération de couleurs, la mire offre un choix de barres qui sera utile à l'amateur-émetteur pour juger de la qualité de réception de son signal chez son correspondant.

Une horloge a été intégrée au programme et vient afficher l'heure en haut de la mire. L'indicatif de la station d'émission, ou tout autre texte moyennant une petite modification de logiciel, s'inscrit dans le bas de la mire.

Le format choisi est un format rectangulaire par souci de simplification.

Certains caractères de la police de l'AMSTRAD ont été redéfinis pour les besoins du programme. C'est le rôle des lignes 1240 à 1340. Ils permettent l'obtention de barres verticales plus ou moins rapprochées.

Pour obtenir les six couleurs nécessaires (plus le gris du fond), il a été nécessaire de travailler en MODE 0.

Les couleurs sont redéfinies, pour les différents enciers, à la ligne 1370.

Le listing est ensuite suffisamment commenté pour que le lecteur retrouve aisément le rôle des différents groupes de lignes.

Nous nous attarderons simplement un peu sur la fonction horloge. Elle a été obtenue en utilisant le contrôle des interruptions permis par le BASIC de l'AMSTRAD. La commande EVERY permet d'accéder à un sous-programme tous les N laps de temps. Ici, notre pendule bat la minute. La programmation du chronomètre qui permet cette gestion d'interruptions se fait par multiples de 20 ms d'où le 3000 de la ligne 2000.

L'heure est demandée lors du lancement du programme, sur 4 caractères. Le sous-programme en 1110 exécute le comptage du temps. Vous remarquerez que nous avons choisi de travailler, non sur des compteurs de 0 à 60, mais directement sur les codes ASCII des chiffres à afficher. Point n'est alors besoin de jongler avec les STR\$ et d'ajouter un 0 devant le chiffre à afficher quand heures ou minutes sont inférieures à 10.

Un compteur particulier (HEURE) permet de détecter le passage à 24 heures pour afficher 0000.

Une dernière remarque... La ligne 1350 utilisant la fonction KEY permet de remettre l'ordinateur en MODE 2 lorsque l'on quitte le programme MIRE.

```
1000 ? *****
1010 ? * *
1020 ? * MIRE TV *
1030 ? * *
1040 ? * (c) EDEN Soft *
1050 ? * *
1060 ? * CPC 464 & 664 *
1070 ? * *
1080 ? *****
1090 ?
1100 GOTO 1210
1110 ?----- S/P incrementation horloge -
-----
1120 M=M+1:IF M<58 THEN 1190
1130 M=48:MM=MM+1:IF MM<54 THEN 1190
1140 MM=48:H=H+1:HEURE=HEURE+1
```

```

1150 IF HEURE<24 THEN 1170
1160 HEURE=0:H=48:HH=48:GOTO 1190
1170 IF H<58 THEN 1190
1180 H=48:HH=HH+1
1190 LOCATE 9,3:PRINT CHR$(HH);CHR$(H);C
HR$(MM);CHR$(M)
1200 RETURN
1210 ind$="FD1XYZ"
1220 MODE 1:PRINT:PRINT
1230 INPUT "ENTREZ L'HEURE SOUS LA FORME
  HHMM ";HH$
1240 SYMBOL AFTER 139
1250 SYMBOL 139,255,255,255,255,255,
255,255
1260 SYMBOL 140,240,240,240,240,240,240,
240,240
1270 SYMBOL 141,231,231,231,231,231,231,
231,231
1280 SYMBOL 142,204,204,204,204,204,204,
204,204
1290 SYMBOL 143,219,219,219,219,219,219,
219,219
1300 SYMBOL 144,51,51,51,51,51,51,51,51
1310 SYMBOL 145,170,170,170,170,170,170,
170,170
1320 SYMBOL 146,85,85,85,85,85,85,85,85

1330 SYMBOL 147,0,0,0,255,255,0,0,0
1340 SYMBOL 148,255,255,255,0,0,255,255,
255
1350 KEY 11,"ink 0,13:ink 1,0:paper 0:pe
n 1:mode 2:list"+CHR$(13)
1360 MODE 0:INK 0,13:BORDER 0:PAPER 0:CL
S
1370 INK 1,0:INK 2,26:INK 3,0,26:INK 4,2
4:INK 5,20:INK 6,18:INK 7,7:INK 8,5:INK
9,1:'Definition palette des couleurs
1380 '----- Haut de l'ecran et emplaceme
nt horloge -----
1390 FOR y=2 TO 4
1400 LOCATE 2,y:PEN 2:FOR x=1 TO 6:PRINT
  CHR$(139);:NEXT
1410 LOCATE 8,y:PEN 1:FOR x=1 TO 8:PRINT
  CHR$(139);:NEXT

```

```

1420 LOCATE 14,y:PEN 2:FOR x=1 TO 6:PRINT
CHR$(139);:NEXT
1430 NEXT
1440 '----- Gros damier noir et blanc ---
-----
1450 FOR y=5 TO 7
1460 LOCATE 2,y:FOR n= 1 TO 9:PEN 1:PRINT
CHR$(139);:PEN 2:PRINT CHR$(139);:NEXT

1470 NEXT
1480 '----- Trace des couleurs -----
1490 FOR y=8 TO 10
1500 LOCATE 2,y
1510 FOR c=4 TO 9
1520 FOR n=1 TO 3:PEN c:PRINT CHR$(139);
:NEXT n
1530 NEXT c,y
1540 '----- Trace des barres blanches mi
lieu -----
1550 PEN 2:FOR y=11 TO 15
1560 LOCATE 2,y:FOR n=1 TO 18:PRINT CHR$(
149);:NEXT
1570 NEXT
1580 '----- Couleurs au centre -----
1590 LOCATE 2,13:FOR c=9 TO 4 STEP -1:FO
R n=1 TO 3:PEN c:PRINT CHR$(139);:NEXT n
,c
1600 '----- Trace des barres -----
1610 PEN 1
1620 FOR y=16 TO 18
1630 LOCATE 2,y:FOR n= 1 TO 3:PRINT CHR$(
141);:NEXT
1640 FOR n=1 TO 3:PRINT CHR$(142);:NEXT
1650 FOR n=1 TO 6:PRINT CHR$(145);:NEXT
1660 FOR n=1 TO 3:PRINT CHR$(142);:NEXT
1670 FOR n=1 TO 3:PRINT CHR$(141);:NEXT

1680 NEXT
1690 '----- Degrade NOIR GRIS BLANC -----
--
1700 FOR y=19 TO 21
1710 LOCATE 2,y
1720 PEN 1:FOR n=1 TO 6:PRINT CHR$(139);
:NEXT
1730 PEN 0:FOR n=1 TO 6:PRINT CHR$(139);
:NEXT

```

```

1740 PEN 2:FOR n=1 TO 6:PRINT CHR$(139);
:NEXT
1750 NEXT
1760 '----- BLANC Indicatif NOIR -----
1770 FOR y=22 TO 24
1780 LOCATE 2,y
1790 PEN 2:FOR n=1 TO 3:PRINT CHR$(139);
:NEXT
1800 PEN 0:FOR n=1 TO 3:PRINT CHR$(139);
:NEXT
1810 PEN 1:FOR n=1 TO 6:PRINT CHR$(139);
:NEXT
1820 PEN 0:FOR n=1 TO 3:PRINT CHR$(139);
:NEXT
1830 PEN 1:FOR n=1 TO 3:PRINT CHR$(139);
:NEXT
1840 NEXT
1850 '----- Trace le cadre fin -----
1860 PLOT 28,0,1:DRAW 28,399
1870 PLOT 610,0:DRAW 610,399
1880 PLOT 0,16:DRAW 639,16
1890 PLOT 0,384:DRAW 639,384
1900 '----- Ecrit indicatif -----
1910 PEN 3:LOCATE 8,23:PRINT IND$
1920 '----- Initialise la pendule -----
1930 PEN 2
1940 HH=ASC(MID$(HH$,1,1))
1950 H=ASC(MID$(HH$,2,1))
1960 MM=ASC(MID$(HH$,3,1))
1970 M=ASC(MID$(HH$,4,1))
1980 HEURE=VAL(LEFT$(HH$,2))
1990 GOSUB 1190
2000 EVERY 3000 GOSUB 1120:' Bat la minu
te
2010 WHILE INKEY$="":WEND

```


EMISSION RECEPTION EN MORSE

Nous avons tenu à décrire dans cet ouvrage un logiciel d'émission-réception en télégraphie, bien que la pratique montre que l'émission dans ce mode à partir d'un clavier n'est pas ce qu'il y a de mieux... Rien ne vaut le bon vieux manipulateur, pioche ou électronique. La réception, quant à elle, ne sera parfaite que si la manipulation de la station émettrice est correcte. Supreme avantage du cerveau humain sur la machine, le premier arrive à interpréter ce que la seconde ne comprend pas tant ses critères sont stricts.

Le décodage nécessite l'utilisation d'un circuit d'interface qui se connecte à l'AMSTRAD par la ligne 'BUSY' de la prise imprimante. Ce circuit assure la conversion des signaux BF prélevés sur la sortie haut-parleur ou magnétophone du récepteur en niveaux logiques exploitables par l'ordinateur.

Le meilleur résultat sera obtenu avec un récepteur muni d'un bon filtre à quartz prévu pour la CW.

LE LOGICIEL

Le logiciel se compose de deux parties: l'émission, écrite en BASIC et la routine de réception écrite en langage machine. Bien que nous ayons incorporé les octets du langage machine dans des lignes de DATA, pour ne pas pénaliser les lecteurs ne possédant pas d'assembleur, il nous a paru indispensable de fournir également un listing ASSEMBLEUR abondamment commenté, pour satisfaire la curiosité des amateurs de ce langage.

Les possibilités de fenêtrage de l'AMSTRAD ont été utilisées pour donner au logiciel un aspect plus agréable.

EMISSION

Pour le 464, la question du relais émission-réception ne se pose pas: on ne peut avoir aucun accès à ses contacts... Par contre, pour les CPC 664 et 6128 on peut se demander s'il n'est pas opportun d'utiliser les services du relais de télécommande cassette pour manipuler l'émetteur en CW.

Le relais de commutation cassette de l'AMSTRAD est accessible sur le cordon magnétophone cependant, pour ne pas fatiguer inutilement ce relais, nous avons utilisé la sortie BF de l'ordinateur pour produire les sons. En couplant cette sortie (jack 'stéréo') à l'entrée micro de l'émetteur, en passant au besoin par un préampli, il est possible de procéder à des émissions qui sont, en fait, de la pseudo-CW. Si les puristes veulent modifier le programme, celui-ci étant en BASIC, il sera toujours possible de commander, dans la partie émission, le relais de commutation. Attention toutefois à ne pas couper avec des courants trop intenses...

Pour commander le relais, on aura recours à l'instruction OUT du BASIC. Le relais est piloté par le PORT C du 8255 dont l'adresse est &F6xx. Le relais est câblé sur la ligne 4 du PORT C du PPI. On le programmera par:

```
OUT &F600,16 pour le coller
OUT &F600,0  pour le remettre au repos.
```

La partie émission donne en fait accès à des messages mémorisés, l'ordinateur devant être employé 'en parallèle' avec le manipulateur.

Trois vitesses émission sont prévues, restant dans la moyenne de ce qui se pratique couramment sur l'air.

Les messages mémorisés devront impérativement être écrits en MAJUSCULES dans le programme.

La tonalité et le niveau BF du signal produit par l'ordinateur sont fixés à la ligne 1390 (variables 'TON' et 'QRK') et pourront, si besoin, être modifiées.

Attention lors de l'introduction des codes MORSE, les virgules ont leur importance.

Les différents messages disponibles sont constamment affichés à l'écran et sont désignés par leur numéro.

La vitesse de transmission peut être modifiée, même en cours d'émission, par simple appui sur les touches '>' et '<'.

L'appui sur 'Q' permet de passer en réception.

RECEPTION

Le module de réception étant essentiellement écrit en langage machine, nous invitons le lecteur à se reporter aux commentaires correspondants.

On peut, tout en décodant, ajuster au mieux la vitesse, bien que ce réglage ne soit pas très pointu...

Le retour en émission s'obtient par appui sur la touche 'Q'. Les dernières lignes du texte reçu ne sont pas pour autant perdues, l'écran effectuant un scrolling vers le haut.

```
1000 ? *****
1010 ? *                                     *
1020 ? *           E/R MORSE             *
1030 ? *                                     *
1040 ? *           F1EZH F6GKQ          *
1050 ? *                                     *
1060 ? * (C) EDEN Software 85 *
1070 ? *                                     *
1080 ? *****
1090 :
1100 :
1110 MEMORY &8FFF:GOTO 1190
1120 REM Execution code MORSE
1130 IF car$=" " THEN SOUND 1,120,3*pt,0:
RETURN
1140 FOR k=1 TO LEN(car$):e$=MID$(car$,k
,1)
1150 IF e$="." THEN d=pt ELSE d=tt
1160 SOUND 1,ton,d,qrk: SOUND 1,120,pt,0
1170 NEXT k
1180 SOUND 1,120,tt,0:RETURN
1190 REM Initialisations
1200 REM Remplissage tables des codes
1210 DIM cod$(47): FOR n=0 TO 47: READ c
od$(n): NEXT
1220 REM Implantation du code machine
1230 FOR AD=&9000 TO &9191:READ d$:POKE
ad,VAL("&"+d$):NEXT
1240 REM Presentation
1250 MODE 1:INK 0,1:PAPER 0:INK 1,24:PEN
1:BORDER 1:WINDOW #1,8,30,7,17:INK 3,8:
PAPER #1,3:CLS#1:INK 2,0:PEN #1,2
```

```

1260 PRINT #1:PRINT #1,"          ERCW"
1270 PRINT #1:PRINT #1," Emission Recept
ion CW"
1280 PRINT #1:PRINT #1,"          F1EZH ET F6G
KQ"
1290 PRINT #1:PRINT #1," ";CHR$(164);" E
DEN Software 1985"
1300 PRINT #1:PRINT #1,"          CPC 464 et 6
64"
1310 LOCATE #0,6,20:PRINT "Ajustez votre
volume sonore"
1320 LOCATE #0,7,21:PRINT "et appuyez su
r une touche"
1330 LOCATE #0,2,24:PRINT"CLAVIER IMPERA
TIVEMENT EN CAPS LOCK"
1340 a$=INKEY$:WHILE a$="" :a$=INKEY$:SOU
ND 1,120,5,12:WEND
1350 REM Debut du programme principal
1360 MODE 2:INK 0,13:PAPER 0:CLS:INK 1,0
:PEN 1:BORDER 13
1370 WINDOW #3,1,80,13,25:PAPER #3,1:CLS
#3:PEN #3,0
1380 WINDOW #1,1,80,1 ,12:PAPER #1,0:CLS
#1:PEN #1,1
1390 qrk=11:ton=120:VIT=3:TT=6*(6-VIT):p
t=TT/3
1400 REM Table des differents messages
1410 REM Ecrits imperativement en MAJUSC
ULES
1420 m$(1)="CQ CQ CQ DE F6GKQ/35 F6GKQ/3
5 F6GKQ/35 + K"
1430 m$(2)="GM DR OM ES TKS FR UR CALL =
"
1440 M$(3)="MY QTH IS NR RENNES RENNES R
ENNES W FRANCE ="
1450 M$(4)="LOCATOR IS IN9BDA IN9BDA IN9
BDA ="
1460 M$(5)="MY NAME IS DENIS DENIS DENIS
="
1470 M$(6)="RIG IS YAESU FT 77 80 WATTS
ES ANT IS 4BTV VERTICAL ANT ="
1480 M$(7)="QSL VIA BUR0 = 73 ES TKS FR
NICE QSO HPE TO CUAGN ES GD TFC ="
1490 M$(8)="CQ TEST CQ TEST CQ TEST DE F
6GKQ F6GKQ F6GKQ PSE TEST K"
1500 GOTO 1790

```

```

1510 REM Partie EMISSION
1520 WINDOW #4,1,80,1,12:PAPER #4,0:PEN
#4,1
1530 WINDOW #3,1,80,13,25:PAPER #3,1:CLS
#3:PEN #3,0
1540 PRINT #3
1550 FOR N=1 TO 8:PRINT #3,N;"- ";M$(N):
NEXT
1560 PRINT #3:PRINT #3:PRINT #3," '<' D
iminuer VITESSE '>' Augmenter VITE
SSE 'Q' Quitter EMISSION "
1570 LOCATE #4,1,10:PRINT #4,
1580 T$=INKEY$:IF T$="" THEN 1580
1590 T=VAL(T$):MES$=M$(T)
1600 REM Decoupe et emet le message
1610 FOR n=1 TO LEN(mes$):c$=MID$(mes$,n
,1)
1620 PRINT #4,C$;
1630 IF C$=" " THEN CAR$=" ":GOTO 1650
1640 car$=cod$(ASC(c$)-43)
1650 GOSUB 1120:T$=INKEY$:IF T$="" THEN
1700
1660 IF T$="Q" THEN N=LEN(MES$):GOTO 170
0
1670 IF T$="<" THEN VIT=VIT-1:IF VIT=0 T
HEN VIT=1
1680 IF T$=">" THEN VIT=VIT+1:IF VIT=5 T
HEN VIT=4
1690 TT=6*(6-VIT):PT=TT/3:GOTO 1700
1700 NEXT:PRINT #4
1710 IF T$="Q" THEN 1790 ELSE 1580
1720 REM Partie RECEPTION
1730 WINDOW #4,1,80,1 ,21:PAPER #4,0:CLS
#4:PEN #4,1
1740 WINDOW #3,1,80,22,25:PAPER #3,1:CLS
#3:PEN #3,0
1750 LOCATE #3,1,2:PRINT #3," '<' Dimin
uer VITESSE '>' Augmenter VITESSE
'Q' Quitter RECEPTION"
1760 LOCATE #3,31,4: PRINT #3,"Vitesse a
ctuelle :";:PRINT #3,3;
1770 FOR N=1 TO 10:PRINT #4,:NEXT
1780 CALL &904A: ' Appel du S/P de DECOD
AGE
1790 CLS #3:WINDOW #3,1,80,22,25:PAPER #
3,1:CLS #3:PEN #3,0

```

```

1800 LOCATE #3,20,2:PRINT #3," R Pour RE
CEPTION          E Pour EMISSION"
1810 LOCATE #3,30,4:PRINT #3,"Votre choi
x -> ";:INPUT #3,cx$
1820 CLS #3:IF cx$="R" THEN 1720
1830 FOR N= 1 TO 12:PRINT #4,:NEXT:GOTO
1510
1840 REM Table des signes MORSE
1850 REM Signes de ponctuation (43 a 47)
1860 DATA .-.-.,---.---,---.---,---.---,---.
.
1870 REM Chiffres (48 a 57)
1880 DATA -----,-----,-----,-----,-----,
-----,-----,-----,-----,-----,-----,
-----,-----,-----,-----,-----,-----,
1890 REM Signes (suite) (58 a 64)
1900 DATA ,,-.---,---.---,---.---,---.---,
1910 REM Lettres (65 a 90)
1920 DATA .-,---,---,---,---,---,---,---,---,---,
---,---,---,---,---,---,---,---,---,---,---,---,
---,---,---,---,---,---,---,---,---,---,---,---,
1930 REM Code machine routine DECODAGE
1940 DATA 20,54,45,4D,4E,41,49,4F,47,4B
1950 DATA 44,57,52,55,53,20,20,51,5A,59
1960 DATA 43,58,42,4A,50,3A,4C,20,46,56
1970 DATA 48,30,39,20,38,20,20,20,37,20
1980 DATA 3E,2E,20,2C,2F,2D,36,31,20,22
1990 DATA 3F,20,2A,20,23,32,20,5B,20,33
2000 DATA 20,34,35,20,13,08,00,00,08,03
2010 DATA 00,00,00,00,3E,01,32,43,90,3E
2020 DATA 00,32,40,90,32,42,90,3E,15,32
2030 DATA 41,90,32,44,90,11,00,00,3E,03
2040 DATA 32,45,90,CD,1B,BB,DC,48,91,CD
2050 DATA 2E,BD,38,F5,CD,1B,BB,DC,48,91
2060 DATA CD,2E,BD,30,F5,3A,45,90,3F,CB
2070 DATA 07,CB,07,06,B0,10,FE,3D,20,F9
2080 DATA CD,1B,BB,DC,48,91,3A,43,90,47
2090 DATA CD,2E,BD,38,04,3E,00,18,02,3E
2100 DATA 01,B8,20,0D,3A,40,90,FE,FF,28
2110 DATA 0D,3C,32,40,90,18,CC,32,43,90
2120 DATA E6,FF,28,17,3A,44,90,CB,07,47
2130 DATA 3A,40,90,B8,30,4B,CB,02,CB,03
2140 DATA 3E,00,32,40,90,18,DC,3A,40,90
2150 DATA 4F,CB,07,81,47,3A,41,90,CB,07
2160 DATA 32,41,90,B8,30,1F,3A,40,90,47
2170 DATA 3A,41,90,B8,38,07,3A,42,90,E6

```

```

2180 DATA FF,28,0E,3E,01,32,42,90,1C,3A
2190 DATA 40,90,32,41,90,18,C7,3A,40,90
2200 DATA 32,44,90,3E,00,32,42,90,14,18
2210 DATA EA,7A,00,CB,07,8B,C6,00,FE,40
2220 DATA 38,04,3E,2A,18,0B,26,90,6F,7E
2230 DATA F5,3E,04,CD,B4,BB,F1,CD,5A,BB
2240 DATA 11,00,00,3A,44,90,CB,07,CB,07
2250 DATA 87,00,00,00,47,3A,40,90,B8,38
2260 DATA 05,3E,20,CD,5A,BB,18,80,FE,51
2270 DATA 20,02,E1,C9,FE,3E,20,0D,21,45
2280 DATA 90,7E,FE,09,28,15,34,CD,70,91
2290 DATA C9,FE,3C,20,FB,21,45,90,7E,FE
2300 DATA 01,28,04,35,CD,70,91,C9,CD,78
2310 DATA BB,E5,3E,03,CD,B4,BB,F5,26,32
2320 DATA 2E,04,CD,75,BB,3A,45,90,C6,30
2330 DATA CD,5A,BB,F1,CD,B4,BB,E1,CD,75
2340 DATA BB,C9

```

```

; --- CWD7 ---          CE LISTING NE DOIT PAS ETRE SAISI
;
ORG 9000H              Adresse du début d'implantation
LOAD 9000H
;
TABLE: EQU 9000H      Début de la table de transcodage
DR: EQU 9040H        Durée de l'élément reçu
DP: EQU 9041H        Durée du signal précédent
LSTELT: EQU 9042H    Type de lecture précédente (0=espace 1=élément)
LSTSIG: EQU 9043H    Type du signal précédent (0=point 1=trait)
DURPNT: EQU 9044H    Durée du dernier point reçu
SPEED: EQU 9045H     Base de la vitesse de décodage
;
PORTB: EQU 0F532H    Adresse physique du PORT B (Pour BUSY imprimante)
OUTVDU: EQU 0BB5AH  Sortie d'un caractère vers l'écran
CLAVIE: EQU 0BB1BH  Lecture du clavier (pour test différentes options)
BUSY: EQU 0BD2EH    Lecture de la ligne BUSY imprimante
SETCUR: EQU 0BB75H  Positionne le curseur sur l'écran
GETCUR: EQU 0BB78H  Lit la position présente du curseur
INVERT: EQU 0BB9CH  Pour affichage en vidéo inversée
SETSTR: EQU 0BB84H  Sélection du numéro de fenetre
;
;

```

;--- TABLE DE TRANSCODAGE ---

DB 20H,54H,45H,4DH,4EH,41H,49H,4FH
DB 47H,4BH,44H,57H,52H,55H,53H,20H
DB 20H,51H,5AH,59H,43H,58H,42H,4AH
DB 5DH,3AH,4CH,20H,46H,56H,48H,3DH
DB 39H,20H,38H,20H,20H,20H,37H,20H
DB 3EH,2EH,20H,2CH,2FH,2DH,36H,31H
DB 20H,22H,3FH,20H,2AH,20H,23H,32H
DB 20H,5BH,20H,33H,20H,34H,35H,20H

Ici sont logés les différents codes des caractères pour leur affichage après identification du caractère MORSE et transcodage.

Les caractères non utilisés sont forcés au code 20H (espace).

;
;

DS 0AH

Réserve les mémoires de travail

;
;
;

;--- INITIALISATIONS ---

;

LD A,1
LD (LSTSIG),A
LD A,0
LD (DR),A
LD (LSTELT),A
LD A,15H
LD (DP),A
LD (DURPNT),A
LD DE,0
LD A,3
LD (SPEED),A

Force les valeurs par défaut de toutes les variables contrôlant le fonctionnement du programme. Lecture précédente était un élément, dernier élément reçu était un point, durée du dernier reçu était 15H, ainsi que celle du dernier point reçu. Les compteurs POINT, TRAIT, ELEMENT sont mis à zéro.

La vitesse de décodage est initialisée à 3.

;

;--- ATTENTE D'UN ESPACE ---

;

WAISPC: CALL CLAVIE
CALL C,TSTKBD
CALL BUSY
JR C,WAISPC

Test du clavier pour détecter si une touche est enfoncée. Si oui, décode la touche. Boucle tant que l'entrée BUSY est à 1.

;

;--- ATTENTE D'UN ELEMENT ---

;

WAIELM: CALL CLAVIE
CALL C,TSTKBD
CALL BUSY
JR NC,WAIELM

Test du clavier pour détecter si une touche est enfoncée. Si oui, décode la touche. Boucle tant que l'entrée BUSY est à 0.

;

;--- ECHANTILLONNAGE ---

;

MESURE: LD A,(SPEED)

Valeur initiale de la boucle d'échantillonnage.


```

      CCF
      RLC A           Multiplie la valeur de tempo par 4.
      RLC A
LOOP2: LD B,OB0H     Délai entre deux lectures (échantillonnage)
LOOP1: DJNZ LOOP1
      DEC A
      JR NZ,LOOP2
      CALL CLAVIE     Test du clavier pour savoir si une touche est
      CALL C,TSTKBD   enfoncée et la décode si c'est le cas.
;
;--- LIRE ENTREE ET COMPARER AVEC
; LA LECTURE PRECEDENTE.
;
      LD A,(LSTSIG)   Préserve dans B l'ancienne lecture.
      LD B,A
      CALL BUSY       Lit l'état de l'entrée et détecte la présence
      JR C,ELEMNT     d'un état 1 (élément).
ESPACE: LD A,0        A sert d'indicateur. On y met 0 si on a lu un
      JR TEST         espace sur l'entrée et
ELEMNT: LD A,1        i si c'était un élément.
TEST:   CP B          Compare à l'élément précédent. Si différent, on
      JR NZ,NEWLST    préserve la nouvelle valeur.
;
;--- PAS DE CHANGEMENT, CONTINUONS
;
      LD A,(DR)       Il n'y a pas eu de changement, on continue la
      CP OFFH         boucle. Si le compteur d'échantillonnage est plein
      JR Z,SPACE      on affichera un ESPACE.
INCDUR: INC A        Incrémente le compteur de mesure de temps
      LD (DR),A       et le sauvegarde puis
      JR MESURE       saute à la mesure suivante.
;
;--- ELEMENT RECU VIENT DE CHANGER
; ON LE SAUVEGARDE DANS ELEMENT
; PRECEDENT.
NEWLST: LD (LSTSIG),A L'accu contient le dernier signal recu.
      AND OFFH
      JR Z,PTORTT     Le dernier signal n'était pas un ESPACE.
;
;--- DERNIER SIGNAL ETAIT 1 ESPACE
; ENTRE ELEMENTS OU CARACTERES
;
SPACE:  LD A,(DURPNT) Le dernier signal était un ESPACE. On cheche à
      RLC A           savoir si c'était un ESPACE inter-éléments ou
      LD B,A           inter-caractères. Pour cela, on calcule durée du
      LD A,(DR)       POINT*2 et on effectue la comparaison avec

```

	CP B -	le signal reçu.
	JR NC,AFFICH	Vers recherche dans la table puis affichage.
	RLC D	Rotation des registres des POINTS et des TRAITIS
	RLC E	pour préparer la prochaine réception.
RAZDUR:	LD A,0	Remet à zéro la variable contenant la durée du
	LD (DR),A	signal reçu.
	JR INC DUR	
	;	
	;--- ELEMENT RECU POINT OU TRAIT ?	
	;	
PTORTT:	LD A,(DR)	Détermine si l'élément reçu était un POINT ou
	LD C,A	un TRAIT. On teste si sa durée est inférieure
	RLC A	aux 2/3 de l'élément précédent.
	ADD A,C	
	LD B,A	Si oui, c'est un POINT, si non, c'est PEUT-ETRE
	LD A,(DP)	un TRAIT.
	RLC A	
	LD (DP),A	
	CP B	
	JR NC,POINT	
	;	
	;	
	;	
	LD A,(DR)	Compare la durée du signal reçu avec 2 fois la
	LD B,A	durée du signal précédent.
	LD A,(DP)	
	CP B	
	JR C,TRAIT	Plus grand, c'est un TRAIT, plus petit, un TRAIT.
	LD A,(LSTELT)	L'élément précédent était-il un POINT ?
	AND OFFH	Si oui, le dernier reçu est
	JR Z,POINT	aussi un POINT.
	;	
	;	
	;	
TRAIT:	LD A,1	Le type de signal précédent devient un TRAIT.
	LD (LSTELT),A	Pour cela, on force A à 1.
	INC E	Incrémente le registre des TRAITIS.
ACTUAL:	LD A,(DR)	Le signal reçu devient...
	LD (DP),A	...le signal précédent.
	JR RAZDUR	
POINT:	LD A,(DR)	L'élément reçu était un POINT. On se sert de sa
	LD (DURPNT),A	durée pour réactualiser la valeur durée POINT.
	LD A,0	Le type de signal précédent devient un POINT.
	LD (LSTELT),A	Pour cela, on force A à 0.
	INC D	Et on incrémente le registre des POINTS.
	JR ACTUAL	

```

;
;--- SOUS-PROGRAMME AFFICHAGE ---
;
;   POUR AFFICHER ON CALCULE LE
;   RANG DANS LA TABLE DE TRANSCO
;   DU CARACTERE PAR 2*D+E
;
AFFICH: LD A,D           Le rang dans la table de transcodage s'obtient
        NOP
        RLC A           par 2 fois le contenu du registre D (POINTS)
        ADC A,E         plus une fois le contenu de E (TRAITS).
        ADD A,D         Compare si la valeur ainsi obtenue ne dépasse
        CP 40H         pas la valeur limite de la table de transcodage.
        JR C,POINTE
        LD A,'*'       Affiche une * si le caractère est invalide.
        JR DSP
POINTE: LD H,90H       Poids fort de l'adresse début de table dans H.
        LD L,A         Poids faible (rang du caractère) dans L.
        LD A,(HL)     Pointe dans la table
        PUSH AF       et préserve le caractère récupéré.
        LD A,4        Numéro de fenetre pour affichage du caractère
        CALL SETSTR   (changement fenetre décodage ou vitesse)
        POP AF        Récupère le code du caractère
DSP:    CALL OUTVDU   et l'affiche...
        LD DE,0
        LD A,(DURPNT) Calcule le critère ESPACE comme étant égal à
        RLC A         5 fois la valeur de la durée du POINT.
        RLC A
        ADD A,A
        NOP
        NOP
        NOP
        LD B,A        Préserve le résultat dans B.
        LD A,(DR)
        CP B
        JR C,ICI2
INTMOT: LD A,20H      Affiche un ESPACE si le dernier élément reçu avait
        CALL OUTVDU   une durée supérieure à 5 POINTS.
ICI2:   JR RAZDUR
;
;
;--- LECTURE DU CLAVIER --- Différentes commandes pendant le décodage.
;
TSTKBD: CP 'Q'       Q pour quitter le décodage
        JR NZ,INCREM
        POP HL
        RET          et revenir au BASIC.

```

```

;
INCREM: CP '>' > pour incrémenter la vitesse de décodage.
        JR NZ,DECREM
        LD HL,SPEED Lit la valeur de la vitesse actuelle
        LD A,(HL) et compare si égale à 9. Dans ce cas, on ne
        CP 09H l'incrémentera pas...
        JR Z,NOMOD
        INC (HL)
        CALL DSPSPD Affiche la vitesse dans la fenetre réservée.
OUT:    RET
;
DECREM: CP '<' < pour diminuer la vitesse de décodage.
        JR NZ,OUT
        LD HL,SPEED Récupère la vitesse actuelle.
        LD A,(HL) Si déjà à 1 on ne la modifiera pas.
        CP 01H
        JR Z,NOMOD
        DEC (HL) Décrémente la valeur de la vitesse de décodage.
        CALL DSPSPD Affiche dans la fenetre réservée.
NOMOD:  RET
;
; --- AFFICHAGE VITESSE --- Dans une fenetre réservée, au bas de l'écran.
;
DSPSPD: CALL GETCUR Lit la position présente du curseur.
        PUSH HL La sauvegarde...
        LD A,3 Installe la fenetre numéro 3.
        CALL SETSTR
        PUSH AF Récupère numéro fenetre ou était le curseur.
        LD H,50 Force la nouvelle position du curseur sur la
        LD L,4 dernière ligne de la fenetre courante.
        CALL SETCUR
        LD A,(SPEED) Affiche la valeur de la vitesse, en calculant le
        ADD A,30H code ASCII du caractère correspondant.
        CALL OUTVDU
        POP AF Récupère le numéro de fenetre "texte".
        CALL SETSTR
        POP HL
        CALL SETCUR
        RET Replace le curseur dans la partie "texte".
;
END

```

MEGAHERTZ

REVUE EUROPEENNE D'ONDES COURTES

CHAQUE MOIS, N'OUBLIEZ PAS!



SORACOM
editions

10, Av. du Général de Gaulle
BP 46
35170 BRUZ



SELECTION VITESSE

- 1- LENTE
- 2- MOYENNE
- 3- RAPIDE

Votre choix ? 3

TYPE DE LA DICTEE

- 1- Dctee Chiffres seuls
- 2- Dctee Lettres seules
- 3- Dctee complete
- 4- Dctee avec QRM et QSB

Votre choix ? 3

Dctee de type 3 le 05/12/1986

APPUYEZ SUR UNE TOUCHE POUR COMMENCER...

ESC Interrompt la dctee, SPACE continue

3F+ K IXG? >/0787 FIJ< LF,< 9WV, XBH 51R
< H4/ H +3HLY X.O> V1 0/50 FL?I <IROD L
7G 07B LNXZ =,B FM80X C U2

MONIMORSE

Le rôle de ce logiciel est de permettre l'apprentissage de la télégraphie MORSE, l'ordinateur étant un professeur infatigable et... imperturbable.

Pour apprendre la télégraphie, il faut respecter certaines règles et ne pas tenter de bruler les étapes. Il faut d'abord s'astreindre à apprendre les différents signes (lettres, chiffres, signes de ponctuation et signes spécifiques), à savoir les identifier sans erreur, puis acquérir de la vitesse.

L'apprentissage de la manipulation ne se fera que quand les différents signes seront bien connus. Plusieurs ouvrages ont été publiés sur le sujet : nous n'y reviendrons pas.

Le logiciel proposé est capable de :

- Générer des groupes de 6 signes pendant la phase d'apprentissage.
- Composer des dictées à caractère aléatoire.
- Simuler un mode trafic proche de la réalité.

La génération de groupes de signes :

Sept groupes de six caractères ont été définis, respectant un certain ordre généralement admis pour l'étude de la télégraphie MORSE (CW).

Ces 7 groupes pourront être étudiés à la cadence propre à l'élève. Quand un groupe est bien acquis, on passe au suivant. Ce n'est que lorsque tous les groupes de signes seront parfaitement maîtrisés que l'élève pourra se lancer dans les dictées.

Il y a quatre types de dictées :

La première permettra de réviser toutes les lettres, la seconde les chiffres et la troisième, un mélange de chiffres, de lettres et de signes de ponctuation. La quatrième simule un mode proche du trafic perturbé habituellement rencontré sur les bandes amateurs.

Ce dernier mode simule des baisses de niveau sur le signal (comme un réel fading ou QSB) et des interférences avec une autre émission (brouillage ou

QRM). L'élève devra s'entraîner à décoder le signal le plus aigu. Notons au passage que, décoder l'autre signal aurait été plus difficile, tenant compte des caractéristiques de l'oreille.

Le texte produit par l'ordinateur est, en même temps, affiché sur l'écran. L'élève qui écoute ne doit, bien sûr, pas regarder. Le texte intégral de la dictée pourra être édité sur imprimante permettant ainsi un suivi des progrès de l'élève (en comparant le "corrigé" et le texte "décodé"). La largeur de sortie sur l'imprimante a été fixée à 40 caractères pour tenir compte de certains matériels (exemple GP 50) disponibles sur le marché. Pour changer cela, il suffit de modifier le WIDTH 40 de la ligne 2610.

Écrit entièrement en BASIC, ceci étant rendu possible par les qualités de vitesse de l'AMSTRAD, le programme appelle peu de commentaires.

Les signes MORSE sont placés en DATA et rangés dans une table des codes COD\$ ainsi que leurs codes ASCII dans la table COD. La dictée est composée aléatoirement, en tenant compte des critères choisis, et rangée dans deux tables, COD\$ pour les formes MORSE, COD pour les formes ASCII.

Si l'option "Trafic" est choisie, les brouillages sont produits ainsi :

- Les variations de niveau (Fading) par variations lente de la sortie sonore, les différentes valeurs étant rangées dans une table QSB. A chaque signal sonore produit par l'AMSTRAD est associée une variation de volume prélevée dans cette table. Cette variation de volume agit sur la variable QRK modulant le niveau sonore.

- L'interférence avec une autre station est simulée par la production simultanée du texte de la dictée, lu à l'envers, sur un autre canal sonore et avec une tonalité différente.

Toutes les adaptations restent possibles...

```
1000 ? *****
1010 ? * *
1020 ? * MONIMORSE *
1030 ? * *
1040 ? * F1EZH F6GKQ *
1050 ? * *
1060 ? * (C) EDEN Software 85 *
1070 ? * *
1080 ? *****
1090 :
1100 :
1110 GOTO 1320
1120 REM Execution code MORSE
1130 IF fad=0 THEN 1190
1140 REM Produit le QSB si selectionne
```



```

1150 q=qsb(n):IF q=0 THEN qrk=qrk-1 ELSE
  qrk=qrk+1
1160 IF qrk<7 THEN qrk=7
1170 IF qrk>15 THEN qrk=15
1180 REM Sortie sonore
1190 IF car$=" "THEN SOUND 1,120,7*pt,0
1200 FOR k=1 TO LEN(car$):e$=MID$(car$,k
,1)
1210 IF e$="." THEN d=pt ELSE d=tt
1220 SOUND 1,ton,d,qrk: SOUND 1,120,pt,0
1230 NEXT k
1240 SOUND 1,120,tt,0:IF qrm=0 THEN RETU
RN
1250 REM Produit le QRM si selectionne
1260 IF qrm$=" "THEN SOUND 2,ton2,7*pt,0
1270 FOR k=1 TO LEN(qrm$):e$=MID$(qrm$,k
,1)
1280 IF e$="." THEN d=pt ELSE d=tt
1290 SOUND 2,ton2,d,6: SOUND 2,ton2,pt,0
1300 NEXT k
1310 SOUND 2,ton2,tt,0:RETURN
1320 DIM dic(255),dic$(255),qsb(255)
1330 qrk=11:ton=120:ton2=200
1340 REM remplissage tables des codes
1350 DIM cod$(45): FOR n=0 TO 45: READ c
od$(n): NEXT
1360 DIM cod(45): FOR n=2 TO 9: READ cod
(n):NEXT
1370 FOR n=10 TO 19:cod (n)=n+38:NEXT
1380 FOR n=20 TO 45:cod (n)=n+45:NEXT
1390 REM Presentation
1400 MODE 1:INK 0,1:PAPER 0:INK 1,24:PEN
  1:BORDER 1:WINDOW #1,8,30,7,17:INK 3,8:
PAPER #1,3:CLS#1:INK 2,0:PEN #1,2
1410 PRINT #1:PRINT #1,"          MONIMORSE
"
1420 PRINT #1:PRINT #1,"          Moniteur de M
ORSE"
1430 PRINT #1:PRINT #1,"          F1EZH ET F6G
KQ"
1440 PRINT #1:PRINT #1," ";CHR$(164);" E
DEN Software 1985"
1450 PRINT #1:PRINT #1,"          CPC 464 et 6
64"
1460 LOCATE #0,6,20:PRINT "Ajustez votre
volume sonore"
1470 LOCATE #0,7,21:PRINT "et appuyez su
r une touche"

```

```

1480 a$=INKEY$:WHILE a$="":a$=INKEY$:SOU
ND 1,120,5,12:WEND
1490 MODE 1:INK 0,13:PAPER 0:CLS:INK 1,0
:PEN 1:BORDER 13
1500 CLS:PRINT:PRINT:INPUT "Entrez la da
te JJ/MM/AA ";dat$
1510 CLS:PRINT:PRINT:PRINT "1- Pour appr
entissage des signes"
1520 PRINT "2- Pour dictees d'entraîneme
nt"
1530 qrm=0:fad=0: PRINT:PRINT
1540 INPUT "Votre choix ";cx
1550 IF cx <1 OR cx>2 THEN 1540
1560 ON cx GOTO 1570,2140
1570 REM Apprentissage des signes"
1580 qrk=12:tt=30:pt=10
1590 CLS:PRINT:PRINT "Apprentissage par
blocs de 6 signes":PRINT "-----
-----"
1600 PRINT:PRINT "1- ETANRK          4-
OVFHQP"
1610 PRINT "2- MDXIGC          5- JLO
123"
1620 PRINT "3- ZUBSVY          6- 456
789"
1630 PRINT "          7- .,?/+="
1640 PRINT "          8- VA AS"
1650 PRINT:PRINT
1660 :INPUT "          Votre choix ";cx: I
F cx<1 OR cx >8 THEN 1660
1670 ON cx GOTO 1690,1700,1710,1720,1730
,1740,1750,1760
1680 REM Aiguillage types de groupes
1690 lec$="ETANRK":GOTO 1770
1700 lec$="MDXIGC":GOTO 1770
1710 lec$="ZUBSVY":GOTO 1770
1720 lec$="OVFHQP":GOTO 1770
1730 lec$="JLO123":GOTO 1770
1740 lec$="456789":GOTO 1770
1750 lec$=".,?/+=":GOTO 1770
1760 lec$="<<<>>>"
1770 CLS:IF cx <>8 THEN 1900
1780 PRINT:PRINT "Vous allez apprendre V
A (>) et AS (<)"
1790 PRINT:PRINT "Ecoutez-les..."
1800 PRINT:PRINT "VA represente par >"

```

```

1810 FOR n= 1 TO 6:car$=cod$(8):GOSUB 11
90:FOR q=1 TO 1000:NEXT:NEXT
1820 PRINT:PRINT "AS represente par <"
1830 FOR n= 1 TO 6:car$=cod$(9):GOSUB 11
90:FOR q=1 TO 1000:NEXT:NEXT
1840 PRINT:PRINT "Apprenez a les reconna
itre melanges...":PRINT:PRINT
1850 lec$="<<<>><>><>><>><<<>><><<>"
1860 FOR n=1 TO 28:c#=MID$(lec$,n,1)
1870 IF c#="<" THEN car$=".-..." ELSE ca
r$="...-.-"
1880 GOSUB 1190:PRINT c#;:NEXT
1890 PRINT:PRINT:PRINT "Appuyez sur une
touche pour continuer"
1900 CLS:PRINT:PRINT "Nous allons appren
dre le groupe ";:PRINT lec$
1910 PRINT:PRINT "Ecoutez chaque signe d
u groupe":PRINT
1920 FOR n= 1 TO 6:c#=MID$(lec$,n,1)
1930 FOR j=2 TO 45: IF ASC(c#)<>cod(j) T
HEN 1940 ELSE c=j:j=45
1940 NEXT
1950 car$=cod$(c):FOR m=1 TO 6:GOSUB 119
0:FOR q=1 TO 1000:NEXT:PRINT c#;:NEXT:PR
INT:NEXT
1960 PRINT:PRINT "Apprenez a les reconna
itre melanges"
1970 PRINT:PRINT "Patientez quelques ins
tants S.V.P."
1980 REM Genere une dictee sur le groupe
1990 z=0
2000 FOR n=1 TO 23:FOR i=1 TO 6
2010 l=INT(RND(1)*6+1):c#=MID$(lec$,l,1)
2020 FOR j=2 TO 45:IF ASC(c#)<>cod(j) TH
EN 2030 ELSE c=j:j=45
2030 NEXT
2040 dic(z)=c:dic$(z)=CHR$(cod(c))
2050 z=z+1:NEXT
2060 dic(z)=0:dic$(z)=" "
2070 z=z+1:NEXT
2080 CLS:PRINT:PRINT "Appuyez sur une to
uche pour commencer ":PRINT:PRINT
2090 SOUND 1,120,100
2100 a$=INKEY$:IF a$="" THEN 2100
2110 REM Dictee du groupe

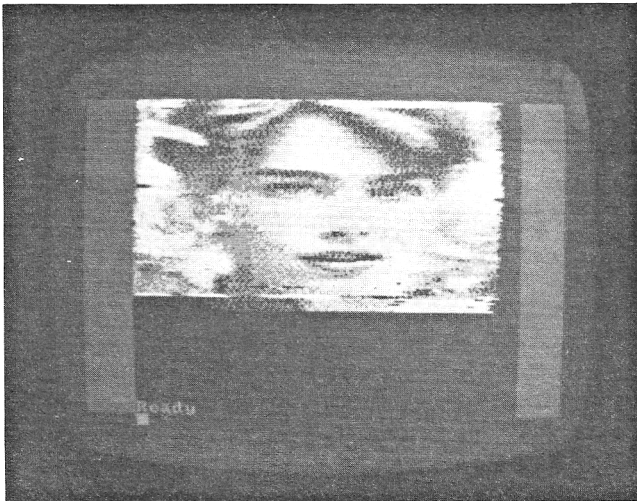
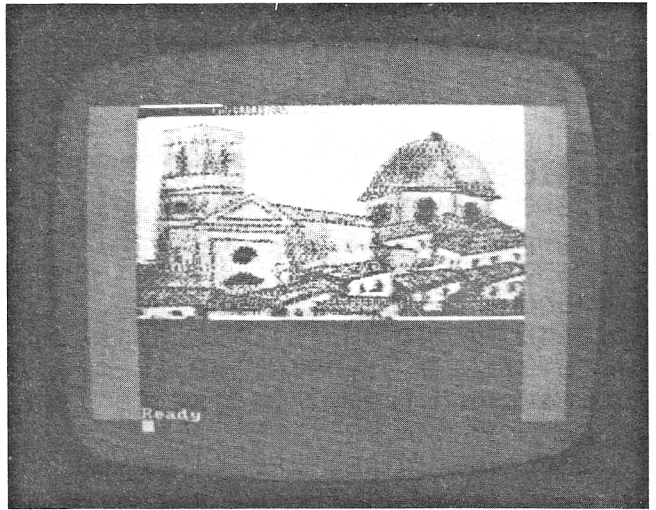
```

```

2120 FOR n=0 TO 180:x=dic(n):car#=cod$(x
):GOSUB 1190:PRINT dic$(n);:NEXT
2130 cx=0:GOTO 2570
2140 CLS:PRINT "SELECTION VITESSE"
2150 PRINT "-----"
2160 PRINT:PRINT:PRINT "1- LENTE":PRINT
"2- MOYENNE":PRINT "3- RAPIDE":PRINT
2170 INPUT "Votre choix ";vit
2180 IF vit <1 OR vit >3 THEN 2170
2190 qrk=11:ton=120:ton2=200
2200 tt=6*(6-vit)
2210 pt=tt/3
2220 PRINT:PRINT:PRINT "TYPE DE LA DICTE
E"
2230 PRINT "-----"
2240 PRINT"1- Dictee Chiffres seuls"
2250 PRINT"2- Dictee Lettres seules"
2260 PRINT"3- Dictee complete"
2270 PRINT"4- Dictee avec QRM et QSB"
2280 PRINT:PRINT
2290 INPUT "Votre choix ";cx
2300 IF cx<1 OR cx>4 THEN 2290
2310 ON cx GOTO 2320,2340,2390,2380
2320 REM Fonction Chiffres seuls
2330 DEF FNgenr=INT(RND(1)*10)+10:GOTO 2
410
2340 REM Fonction Lettres seules
2350 DEF FNgenr=INT(RND(1)*25)+20:GOTO 2
410
2360 REM Genere table de QSB
2370 FOR n=0 TO 255:qsb(n)=INT(RND(1)*2)
:NEXT
2380 fad=1:qrm=1
2390 REM Fonction Dictee complete
2400 DEF FNgenr=INT(RND(1)*44)+2
2410 REM Fonction Hasard espaces
2420 DEF FNespa=INT(RND(1)*6)+2
2430 REM Prepare la dictee
2440 FOR n=0 TO 255:x=FNgenr:dic(n)=x:di
c$(n)=CHR$(cod(x)):NEXT
2450 REM Met des espaces dans dictee
2460 n=0
2470 x=FNespa:n=n+x
2480 IF n>255 THEN 2490 ELSE dic(n)=0:di
c$(n)=" ":GOTO 2470
2490 CLS:PRINT " Dictee de type ";cx
;" le ";dat$:PRINT

```

```
2500 PRINT "APPUYEZ SUR UNE TOUCHE POUR  
COMMENCER..."  
2510 IF QRM=0 THEN 2520 ELSE PRINT:PRINT  
  "DECODEZ LE SIGNAL LE PLUS AIGU DES DEU  
X":PRINT  
2520 PRINT:PRINT "ESC Interrompt la dict  
ee, SPACE continue"  
2530 A$=INKEY$:IF A$=""THEN 2530  
2540 PRINT:PRINT:PRINT  
2550 FOR n=0 TO 255:x=dic(n):y=dic(255-n  
):car$=cod$(x):qrm$=cod$(y):GOSUB 1130  
2560 PRINT DIC$(N);:NEXT  
2570 PRINT:PRINT "Appuyer sur 'C' pour r  
ecopie imprimante"  
2580 a$=INKEY$:IF a$="" THEN 2580  
2590 IF a$ <> "C" AND a$ <> "c" THEN 151  
0  
2600 REM Module de recopie imprimante  
2610 WIDTH 40  
2620 CLS:PRINT:PRINT "Verifier connexion  
imprimante "  
2630 a$=INKEY$:IF a$="" THEN 2630  
2640 IF cx<>0 THEN 2660  
2650 PRINT #8,"DICTEE ENTRAINEMENT SUR L  
ES SIGNES ";:PRINT #8,lec$:GOTO 2670  
2660 PRINT #8, "DICTEE DE TYPE ";CX  
2670 PRINT #8  
2680 PRINT #8,"EFFECTUEE LE ";DAT$  
2690 PRINT #8:PRINT #8:PRINT #8  
2700 FOR N=0 TO 255: PRINT #8,DIC$(N);:N  
EXT  
2710 PRINT #8:PRINT #8  
2720 GOTO 1510  
2730 REM Signes de ponctuation (2 a 9)  
2740 DATA ,,.-.-.,-.---.,---.---.,---  
---.,---.  
---.,---.---.,---.---.,---  
2750 REM Chiffres (10 a 19)  
2760 DATA ---.---.,---.---.,---.---.,---.  
---.,---.---.,---.---.,---.---.  
2770 REM Lettres (20 a 45)  
2780 DATA -.-,-.---.,-.---.,---.,---.  
---.,---.---.,-.---.,---.,---.---.,  
---.,---.,---.,---.,---.,---.,---.  
2790 DATA 43,47,63,61,46,44,62,60
```



EMISSION ET RECEPTION SSTV

Le logiciel que nous proposons ici permet le décodage des émissions effectuées en SSTV (Slow Scan Television, ou encore Télévision à Balayage Lent). Le lecteur intéressé se reportera à l'annexe de ce livre pour un complément d'informations sur ce procédé de transmission.

Il nous est apparu intéressant d'adjoindre à la partie réception, un sous-programme permettant la transmission d'images créées à partir de l'ordinateur. Le logiciel est donc très complet comme le lecteur pourra le constater à la lecture de ce qui suit.

La réception permettra de mémoriser et de sauvegarder les images recues sur cassette ou disquette. L'émission est complétée d'un éditeur permettant de créer des images ou encore, de rappeler des images mémorisées antérieurement.

LA RECEPTION

Pour décoder un tel signal, il existe plusieurs solutions: la plus simple, mais aussi la moins performante, consiste à confier à l'ordinateur le travail d'échantillonnage du signal précédant la détermination des niveaux de gris. Le problème est que, le temps passé à échantillonner est perdu... Résultat, une définition moins bonne de l'image.

De plus, pour faire entrer le signal BF dans l'ordinateur, il faut une entrée 'analogique', ce que ne possède pas le CPC 464. Notons que, sur un 664 ou 6128, il aurait été possible d'utiliser l'entrée 'cassette'.

La solution que nous avons retenue passe par la réalisation d'une carte interface décrite plus loin dans ce livre. Précisons que cette carte pourra aussi servir au RTTY et à la CW... L'utilisateur pourra également l'employer à d'autres tâches.

En réception, le calage du récepteur est très important. Le logiciel vous aidera à régler votre poste sur la bonne fréquence grâce à un indicateur visuel. Ce 'curseur', situé au bas de l'écran indique à tout instant le type de signal reçu (blanc, gris clair, gris moyen, noir ou synchro). Pour que le décodage soit correct, il faut obtenir, en ajustant finement la fréquence de réception, un clignotement permanent du curseur sous la lettre 'S' symbolisant la synchro. Si ce clignotement se fait plus à gauche, retouchez légèrement le réglage du récepteur. Cela demande, au début, un peu d'oreille et de doigté.

Le décodage se fait sur 4 niveaux, dont le blanc et le noir et ce, avec une définition d'environ 128 x 128. Tout dépend en fait du soin apporté à la réalisation et au réglage de l'interface.

L'arrêt de la réception se fait en appuyant sur la touche 'SHIFT', par retour au menu. ATTENTION ! cet arrêt ne sera possible que si l'interface est connectée et si un signal est présent à l'entrée (signal SSTV, QRM etc.) pour débloquer le système.

L'EMISSION

Elle s'effectue directement à partir de l'AMSTRAD, sans l'aide d'interface. On y accède par l'appui sur la touche 'E'. La génération des tonalités est accomplie par l'ordinateur, le signal sortant sur la prise 'jack stéréo' est envoyé vers l'émetteur. Si votre émetteur demande un signal important sur son entrée 'Micro', il sera peut-être nécessaire d'intercaler un préamplificateur entre lui et l'AMSTRAD.

Les images transmises auront été, au préalable, créées par l'éditeur. Vous pourrez vous en confectionner une collection que vous sauvegarderez sur cassette ou disquette.

Les images sont chargées et sauvegardées par groupe de 4. Pendant la transmission, il est possible de changer d'image en appuyant sur 'CTRL'. Le trafic reste donc confortable...

Pour quitter l'émission il suffit, comme pour la réception, d'appuyer sur la touche 'SHIFT'.

Outre ces deux options émission-réception, le menu vous propose:

EFFACEMENT

L'action sur la touche 'V' efface l'image qui est présente à l'écran.

MEMORISATION ET SAUVEGARDE

Les images, recues ou créées, peuvent être mémorisées. Quatre zones mémoire sont disponibles à cet effet. L'appui sur 'P' permet la mémorisation.

Lorsque 4 images ont été mémorisées (ou avant, mais là, il y a perte de place) il est possible de les sauvegarder sur cassette ou disquette en actionnant la touche 'S'.

MIRES

3 mires sont prévues dans le programme: mire de barres, mire de carreaux et mire combinant les deux. On y accède rapidement grace à la touche 'M'.

EDITION

La touche 'T' donne accès à un éditeur plein écran permettant de créer des images ou d'ajouter des informations sur des images existantes. Un curseur apparait en haut et à gauche de l'écran. Il peut être déplacé grace aux touches curseur sans détériorer l'image. Seuls les caractères graphiques et alphanumériques peuvent être utilisés par appui sur les touches correspondantes.

Précisons que, à l'émission, il est préférable de créer de gros caractères grace aux symboles graphiques afin que le correspondant puisse les lire quelles que soient ses conditions de trafic.

Le retour au menu se fait par appui sur la touche 'COPY'.

SAUVEGARDE

Commandée par l'appui sur la touche 'S', la sauvegarde permet de préserver sur disque ou cassette (prévoir un support différent de celui qui contient le programme) la mémoire d'images (4 images) pour une lecture future.

CHARGEMENT

La touche 'C' permet de charger en mémoire un bloc d'images sauvegardées auparavant sur cassette ou disquette. A l'issue du chargement, les 4 images défilent à l'écran, renseignant rapidement du bon fonctionnement de l'opération.

Cette option permet d'appeler à l'écran l'image voulue parmi les 4 en mémoire. Elle est commandée par la touche 'A'.

Pour terminer cette description, signalons qu'un curseur à gauche de l'écran renseigne l'opérateur, pendant l'émission, sur la ligne SSTV émise à tout instant.

CHARGEMENT DU PROGRAMME

Pour préserver un maximum de mémoire, la saisie de ce programme sera un peu plus fastidieuse que pour les autres.

Voici comment vous devrez procéder:

1- Taper le programme BASIC principal.
2- Sans l'exécuter, le sauvegarder sur cassette ou disquette sous le nom 'SSTV'.

3- Faire NEW et taper ensuite le programme 'Chargeur langage machine'.

4- Faire RUN et entrer successivement les codes des tables 1,2,3,4,5 et quand une table est entrée, taper sur la touche arobas (@) pour passer à la suivante. Une erreur peut être corrigée durant chaque entrée de ligne en appuyant sur la touche 'DEL' et en la retapant entièrement.

IL VA SANS DIRE QUE L'ENSEMBLE DU FONCTIONNEMENT DU PROGRAMME SSTV DEPEND DU SOIN APORTE A LA SAISIE DE CES CODES.

5- Quand l'ordinateur affiche, à la fin du travail précédent, 'SAUVEGARDE', mettre le magnétophone en enregistrement (si vous avez un 464) et appuyer sur une touche pour sauvegarder le langage machine à la suite du programme BASIC principal.

Recharger ensuite l'ensemble en faisant RUN 'SSTV' et vérifier le fonctionnement du programme. ATTENTION ! Ne pas passer en 'Réception' si l'interface n'est pas reliée, pour éviter le blocage du programme.

Un large espace mémoire allant de &8300 à &9D00 a été laissé libre pour permettre d'inclure une routine de recopie d'écran sur imprimante. Devant le nombre important de différents types d'imprimantes, nous n'avons pas jugé bon de décrire, dans ce livre, une telle routine. Nous laissons au lecteur le soin d'en choisir une adaptée à son matériel.

QUELQUES COMMENTAIRES SUR LA PARTIE BASIC

Lignes 10 à 50	Initialisations
Lignes 60 +	Dessin du cercle
100-200	Affichage du menu
130-230	Sélection de l'option
240-250	Emission
260-290	Chargement des images
300-320	Sauvegarde des images
330-460	Génération des mires
480-700	Edition d'écrans
710-740	Mémorisation d'images
750-810	Affichage des images
820-850	Réception

PROGRAMME SSTV

CE LISTING NE DOIT PAS ETRE SAISI

```

ORG 7E04H
LOAD 7E04H
MEMO1:EQU 7E00H
MEMO2:EQU 7E02H
LD IX,9E00H           Adresse debut table d'adresses écran
LD B,80H              Nb de lignes = 128
LD DE,(MEMO1)         Adresse de rangement en mémoire
LB1:LD L,(IX+0)
LD H,(IX+1)           HL adresse de début de ligne d'écran
PUSH DE
LD DE,10H
ADD HL,DE             HL adresse début ligne écran utile
POP DE
LD C,30H              C nombre d'octets utiles
LB2:LD A,(HL)
LD (DE),A             Transfert
INC HL
INC DE
DEC C
JR NZ,LB2
INC IX
INC IX
DJNZ LB1              Boucle
RET

```

LD IX,9E00H	Transfert mémoire écran
LD B,80H	
LD DE,(MEM02)	
LB3:LD L,(IX+0)	Adresse de lecture mémoire
LD H,(IX+1)	
PUSH DE	
LD DE,10H	
ADD HL,DE	
POP DE	
LD C,30H	
LB4:LD A,(DE)	
LD (HL),A	
INC HL	
INC DE	
DEC C	
JR NZ,LB4	
INC IX	
INC IX	
DJNZ LB3	Boucle
RET	
	RECEPTION
ORG 8000H	
LOAD 8000H	
LD HL,OFFCOH	Adresse début affichage curseur de calage
LD (0A000H),HL	Mémorisation
DI	
LB001:LD HL,9E00H	Adresse début table des adresses écran
LD B,80H	B = nombre de lignes (128)
LB006:DI	
LD E,(HL)	
INC HL	
LD D,(HL)	DE Adresse de début ligne écran
PUSH HL	
LD HL,0010H	
ADD HL,DE	HL Adresse début écran utile
LD C,48	C nombre d'octets utiles
LB011:LD DE,0488H	D nombre de bits / octets & E masque de lecture
LD A,0	
LD (9DEFH),A	Met la mémoire temporaire à zéro
LB019:PUSH BC	
LD BC,0FBFFH	
IN A,(C)	Lecture port FBFF (interface SSTV)
POP BC	
AND 0FH	Conserve les 4 bits utiles
PUSH HL	
LD HL,(0A000H)	Récupère la position actuelle du curseur
PUSH AF	de calage

XOR A	
LD (HL),A	Efface
POP AF	
PUSH AF	
LD HL,0FFCDH	
ADD A,L	
LD L,A	Affiche le curseur à sa nouvelle position
LD (0A000H),HL	Mémoire cette position
LD (HL),0FFH	
POP AF	
POP HL	
CP.1	Teste si c'est synchro
JR Z,L804C	
PUSH DE	
LD DE,9DF0H	Va chercher la configuration couleur qui
ADD A,E	correspond au signal reçu (recherche dans table)
LD E,A	
LD A,(DE)	
POP DE	
AND E	Masque avec la position du bit
PUSH DE	
LD D,A	
LD A,(9DEFH)	
OR D	
LD (9DEFH),A	Mémoire dans l'octet temporaire
POP DE	
RRC E	Prépare le masque pour bit suivant
CALL L8074	Temporisation
DEC D	
JR NZ,L8019	Boucle tant que les 4 bits image n'ont pas été lus
LD A,(9DEFH)	
LD (HL),A	Affiche l'octet complet (4 bits image)
CALL L8074	Temporisation
INC HL	Octet ligne suivant
DEC C	
JR NZ,L8011	Boucle si ce n'est pas le dernier
L804C:CALL L8085	Attente de synchro
LD A,H	
CP B	Test synchro image
JR C,L804F	
POP HL	
JR L8001	Image suivante
L804F:POP HL	Ligne suivante
INC HL	
PUSH BC	
CALL KEY	Lecture clavier

POP BC	
CP 223	
JR Z,L807F	Arret si appui sur SHIFT
DJNZ L8006	Sinon boucle...
CALL L809C	Attente synchro image
JR L8001	
L8074:PUSH BC	Temporisation entre bits image
LD BC,0010H	
L8078:DEC BC	
LD A,B	
OR C	
JR NZ,L8078	
POP BC	
RET	
L807F:NOP	Arret du programme réception
RET NZ	
POP HL	
EI	
RET	
L8085:PUSH BC	Attente synchro et mesure la durée
CALL KEY	
CP 223	
JR Z,L809A	
LD BC,0FBFFH	
LD HL,0	
L808E:IN A,(C)	
AND 1	
JR Z,L808E	
L8094:IN A,(C)	
INC HL	HL contient la durée de la synchro
AND 1	
JR NZ,L8094	
L809A:POP BC	
RET	
L809C:PUSH BC	Attente synchro image
L809D:LD BC,0FBFFH	
PUSH BC	
CALL KEY	Test clavier
POP BC	
CP 223	
JR Z,L80BC	
IN A,(C)	
AND 1	
JR Z,L809D	
LD BC,08FFH	
L80AE:DEC BC	

LD A,B
OR C
JR NZ,LBOAE
LD BC,0FBFFH
IN A,(C)
AND 1
JR Z,LBD9D
LBOBC:POP BC
RET

EMISSION

ORG 8100H
LOAD 8100H
DI
LD C,0
LD A,0
CALL SOUND Met à 0 le registre 0 du AY-3-8912
LD C,0
LD A,1 Met à 0 le registre 1
CALL SOUND
LD C,15
LD A,8 Met le volume à 15
CALL SOUND
LD C,62
LD A,7 Met en route le canal A
CALL SOUND
TRAME:LD C,52 Synchro image
LD A,0 Génère le 1200 Hz
CALL SOUND
DI
LD BC,0F00H
T1:DEC BC Pendant 30 ms
LD A,C
OR B
JR NZ,T1
LD HL,9E00H HL adresse de la table des adresses écran
LD B,80H B nombre de lignes (128)
LIGNE:LD E,(HL)
INC HL
LD D,(HL)
LD A,33H
LD (DE),A Affichage du curseur
PUSH DE
PUSH HL
LD HL,0010H
ADD HL,DE HL adresse début de ligne utile
LD C,4B C nombre d'octets utiles

LD B,26H	
T2:DJNZ T2	
POP BC	
RET	
SYNC:PUSH BC	Synchro ligne
LD A,0	
LD C,52	
CALL SOUND	Génère le 1200 Hz
DI	
CALL KEY	Lit le clavier
CP 223	
JR Z,STOP	Arret si appui sur SHIFT
CP 127	
JR NZ,SUITE	
LD A,(7E03H)	Si appui sur CTRL, affichage de
CP 66H	l'image suivante
JR Z,IMAGO	
ADD A,18H	
IMAG1:LD (7E03H),A	
PUSH HL	
PUSH DE	
PUSH BC	
CALL 7E2AH	Transfert mémoire-écran
POP BC	
POP DE	
POP HL	
JR SUITE	
IMAGO:LD A,1EH	Si c'est la 5ème, affichage de la 1ère
JR IMAG1	
SUITE:DI	
LD RC,02FFH	Temporisation finale de la synchro ligne (5 ms)
T3:DEC BC	
LD A,C	
OR B	
JR NZ,T3	
POP BC	
RET	
SOUND:LD B,0F4H	Routine d'écriture dans les registres
OUT (C),A	du générateur sonore
LD B,0F6H	
IN A,(C)	
OR 0CDH	
OUT (C),A	
AND 3FH	
OUT (C),A	
LD B,0F4H	

OCTET:LD DE,0488H	D nombre de bits utiles & E masque de lecture
PIXEL:LD A,(HL)	
AND E	Lit le bit image
PUSH HL	
LD HL,9000H	HL adresse début table des fréquences
LD L,A	
LD A,(HL)	
POP HL	
CALL GEN	Génère la fréquence qui correspond à la couleur
CALL TEMPO	
RRC E	Prépare le masque pour le bit suivant
DEC D	
JR NZ,COMPL	Boucle tant que l'octet n'a pas été complètement lu
INC HL	Octet suivant
DEC C	
JR NZ,OCTET	Boucle si ce n'est pas le dernier
CALL SYNC	Synchro ligne
POP HL	
INC HL	
POP DE	Récupère la position du curseur
XOR A	
LD (DE),A	Efface le curseur
DJNZ LIGNE	Boucle si ce n'est pas la dernière ligne
JR TRAME	Image suivante
STOP:POP HL	Arrêt du programme
POP HL	
POP HL	
POP HL	
LD C,D	Coupe le son
LD A,D	
CALL SOUND	
EI	
RET	
COMPL:NOPI	Complément d'instructions
XOR A	servant de
JR NZ,COMPL	temporisation
JR PIXEL	
GEN:PUSH BC	
LD C,A	
LD A,D	
CALL SOUND	Génère la tonalité
DI	
POP BC	
RET	
TEMPO:PUSH BC	Temporisation entre bits

```
OUT (C),C
LD B,0F6H
LD C,A
OR 80H
OUT (C),A
OUT (C),C
RET
```

```
KEY:LD BC,0F40EH
```

Routine de lecture du clavier

```
OUT (C),C
LD B,0F6H
IN A,(C)
AND 30H
LD C,A
OR 0C0H
OUT (C),A
OUT (C),C
INC B
LD A,92H
OUT (C),A
PUSH BC
LD C,042H
LD B,0F6H
OUT (C),C
LD B,0F4H
IN A,(C)
POP BC
PUSH AF
LD A,B2H
OUT (C),A
DEC B
OUT (C),C
POP AF
RET
END
```

```

10 ' E/R SSTV par FC1EZH & F6GKQ
20 ON BREAK GOSUB 500
30 MODE 1:MEMORY &1DFF:PEN 2
40 INK 0,0:INK 1,12:INK 2,19:INK 3,26
50 LOAD"!SSTV2.bin"
60 ORIGIN 124,399:DRAWR 0,-258,3:DRAWR 3
90,0,3:DRAWR 0,258,3
70 ORIGIN 0,0
80 ON BREAK GOSUB 500
90 WINDOW#0,1,40,18,25: WINDOW#1,1,4,1,1
6:WINDOW#2,9,32,1,16:CLS#2
100 CLS: PRINT CHR$(22);CHR$(0);: PRINT"
E-EMISSION":PRINT" R-RECEPTION":PRINT"
S-SAUVEGARDE":PRINT" C-CHARGEMENT":PRINT
" T-EDITION":PRINT" M-MIRE"
110 PRINT" P-MEMORISATION":PRINT" A-AFFI
CHAGE";
120 LOCATE 27,1:PRINT"V-EFFACEMENT"
130 K$=INKEY$:IF K$="" THEN 130
140 IF UPPER$(K$)="E" THEN 240
150 IF UPPER$(K$)="R" THEN 850
160 IF UPPER$(K$)="C" THEN 260
170 IF UPPER$(K$)="M" THEN 360
180 IF UPPER$(K$)="T" THEN 510
190 IF UPPER$(K$)="P" THEN 740
200 IF UPPER$(K$)="A" THEN 780
210 IF UPPER$(K$)="V" THEN CLS#2
220 IF UPPER$(K$)="S" THEN 320
230 GOTO 130
240 CLS:PRINT F$:PRINT:PRINT"
EMISSION":PRINT:PRINT"Pressez 'SHIFT'
pour arreter":PRINT"Pressez 'CTRL' pour
l'image suivante":CALL &8100:CLS
250 LOCATE#1,1,1:FOR N=1 TO 16:PRINT#1,"
":NEXT:GOTO 100
260 CLS:PRINT"CHARGEMENT DE 4 IMAGES"
270 PRINT:INPUT "Nom du fichier a charge
r (8 car. max) ";fi$
280 fi$=fi$+".bin"
290 LOAD fi$:FOR N=0 TO 3
300 POKE &7E03,&1E+&18*N:CALL &7E2A
310 FOR T=1 TO 100:NEXT T,N:GOTO 100
320 CLS:PRINT"SAUVEGARDE DE 4 IMAGES"
330 PRINT:INPUT "Nom du fichier a sauver
(8 car. max) ";fi$

```

```

340 SAVE fi$,B,&1E00,&6000
350 GOTO 100
360 CLS:PRINT:PRINT" B-MIRE DE BARRES":P
RINT" C-MIRE DE CARREAUX":PRINT" D-CARRE
AUX+BARRES"
370 K#=UPPER$(INKEY#):IF K#="" THEN 370
380 IF K#="B" THEN GOSUB 420:GOTO 100
390 IF K#="C" THEN 450
400 IF K#="D"THEN GOSUB 420:C=0:GOTO 460
410 GOTO 370
420 CLS#2:FOR L=1 TO 16:FOR C=0 TO 3:FOR
G=1 TO 6:PEN#2, C:PRINT#2,CHR$(143);:NE
XT G,C
430 NEXT L
440 RETURN
450 PAPER#2,0:PEN#2,3:CLS#2:C=3
460 FOR x=128 TO 508 STEP 48:ORIGIN x,39
9:DRAWR 0,-254,C:ORIGIN x+2,399:DRAWR 0,
-254,C:NEXT
470 FOR y=399 TO 145 STEP -32:ORIGIN 128
,y:DRAWR 382,0,C:ORIGIN 128,y-2:DRAWR 38
2,0,C:NEXT
480 C=3
490 GOTO 100
500 MODE 1:STOP
510 CLS:PRINT"COPY pour arreter"
520 M=1:FOR N=80 TO 66 STEP -1:PRINT CHR
$(24);"CTRL";CHR$(24);CHR$(N);"-";CHR$(N
+63);" ";M=M+1
530 IF M=6 THEN M=1:PRINT
540 NEXT N
550 D=&C010:S=&7FF0:PEN#2,3:R=0
560 X=1:Y=1
570 P=D+(X-1)*2+(Y-1)*&50
580 FOR C=1 TO 2:FOR L=1 TO 8:POKE S,PEE
K(P):P=P+&800:S=S+1:NEXT L:P=P-&3FFF:NEX
T C
590 H=TEST (120+X*16,400-Y*16)
600 LOCATE#2,X,Y:PEN#2,3-H:PRINT #2,CHR#
(22);CHR$(1);CHR$(218);CHR$(22);CHR$(0);

610 K#=INKEY#
620 IF K#="" THEN 610
630 K=ASC(K#)
640 PEN#2,3
650 IF K<32 THEN K=K+127

```

```

660 IF K>31 AND K<&EO THEN LOCATE#2,X,Y:
PRINT#2, CHR$(K);S=&7FFD:GOTO 570
670 IF K=&EO THEN R=1
680 S=&7FFD: P=D+(X-1)*2+(Y-1)*&50
690 FOR C=1 TO 2:FOR L=1 TO 8:POKE P,PEE
K(S):P=P+&800:S=S+1:NEXT L:P=P-&3FFF:NEX
T C
700 IF R=1 THEN 100
710 X=X-1*(K=&F3 AND X<24)+1*(K=&F2 AND
X>1)
720 Y=Y-1*(K=&F1 AND Y<16)+1*(K=&F0 AND
Y>1)
730 S=&7FFD:GOTO 570
740 CLS:PRINT:PRINT"NUMERO D'ORDRE DE L'
IMAGE ?(1 a 4)";
750 INPUT N:IF N>4 OR N<1 THEN 740
760 POKE &7E01,&1E+&18*(N-1):CALL &7E04
770 GOTO 100
780 CLS:N=1:PRINT"ESPACE POUR IMAGE SUIV
ANTE":PRINT"ENTER POUR ARRETER"
790 LOCATE 1,4:PRINT"IMAGE No";N
800 POKE &7E03,&1E+&18*(N-1):CALL &7E2A
810 T$=INKEY$:IF T$="" THEN 810
820 IF ASC(T$)=13 THEN 100
830 N=N+1:IF N=5 THEN N=1
840 GOTO 790
850 CLS:CLS#2:PRINT:PRINT" R
ECEPTION"
860 PRINT:PRINT"APPUYEZ SUR 'SHIFT' POUR
ARRETER"
870 LOCATE 33,7:PRINT"B NS";
880 CALL &8000:GOTO 100

10 DATA 41,51,37,39,34,36,32,34
15 CLS
20 FOR n=1 TO 4
30 READ a,b
40 c=a
50 SOUND 1,c,15,15
60 LOCATE 1,1
70 PRINT BIN$(INP(&FBFF)):PRINT:PRINT:PR
INT"P";n
80 IF c=a THEN c=b ELSE c=a
90 IF INKEY$<>"" THEN 110
100 GOTO 50
110 NEXT n

```

```

10 REM*****CHARGEUR LANG. MACHINE*****
20 MEMORY &7DFF
30 L=&7E00:GOSUB 120
40 L=&8000:GOSUB 120
50 L=&8100:GOSUB 120
60 L=&9D00:GOSUB 120
70 L=&9DF0:GOSUB 120
80 CLS:PRINT"SAUVEGARDE":PRINT:PRINT"APP
UYEZ SUR UNE TOUCHE"
90 IF INKEY#="" THEN 90
100 SAVE"SSTV2.BIN",B,&7E00,&2200
110 END
120 CLS
125 J=L
130 PRINT HEX$(L);". ";CHR$(8);
140 FOR T=1 TO 8
150 N=1:A#=""
160 K#=INKEY$:IF K#="" THEN 160
170 K#=UPPER$(K#)
180 IF ASC(K#)=127 THEN PRINT CHR$(17):P
RINT CHR$(11);:L=J:GOTO 130
190 IF K#<"0" OR K#>"F" THEN 160
200 IF K#>"9" AND K#<"@" THEN 160
210 IF K#="@" THEN RETURN
220 PRINT K#;". ";CHR$(8);
230 A#=A#+K#
240 N=N+1:IF N<>3 THEN 160
250 A=VAL("&"+A#):POKE L,A:L=L+1
260 PRINT". ";CHR$(8);:NEXT T
270 PRINT":GOTO 125

```

TABLE 1

7E00	00	1E	00	66	DD	21	00	9E
7E08	06	80	ED	5B	00	7E	DD	6E
7E10	00	DD	66	01	D5	11	10	00
7E18	19	D1	0E	30	7E	12	23	13
7E20	0D	20	F9	DD	23	DD	23	10
7E28	E5	C9	DD	21	00	9E	06	80
7E30	ED	5B	02	7E	DD	6E	00	DD
7E38	66	01	D5	11	10	00	19	D1
7E40	0E	30	1A	77	23	13	0D	20
7E48	F9	DD	23	DD	23	10	E5	C9

TABLE 2

8000	21	C0	FF	22	00	A0	F3	21
8008	00	9E	06	80	F3	5E	23	56
8010	E5	21	10	00	19	0E	30	11
8018	88	04	3E	00	32	EF	9D	C5
8020	01	FF	FB	ED	78	C1	E6	0F
8028	E5	2A	00	A0	F5	AF	77	F1
8030	F5	21	C0	FF	85	6F	22	00
8038	A0	36	FF	F1	E1	FE	01	28
8040	26	D5	11	F0	9D	83	5F	1A
8048	D1	A3	D5	57	3A	EF	9D	B2
8050	32	EF	9D	D1	CB	0B	CD	84
8058	80	15	20	C3	3A	EF	9D	77
8060	CD	84	80	23	0D	20	B0	CD
8068	94	80	7C	FE	08	38	03	E1
8070	18	95	E1	23	C5	CD	E3	81
8078	C1	FE	DF	28	12	10	8D	CD
8080	B1	80	18	83	C5	01	10	00
8088	0B	78	B1	20	FB	C1	C9	00
8090	C0	E1	FB	C9	C5	CD	E3	81
8098	FE	DF	28	13	01	FF	FB	21
80A0	00	00	ED	78	E6	01	28	FA
80A8	ED	78	23	E6	01	20	F9	C1
80B0	C9	C5	01	FF	FB	C5	CD	E3
80B8	81	C1	FE	DF	28	17	ED	78
80C0	E6	01	28	EE	01	FF	08	0B
80C8	78	B1	20	FB	01	FF	FB	ED
80D0	78	E6	01	28	DD	C1	C9	C9

8100	F3	0E	00	3E	00	CD	C5	81
8108	0E	00	3E	01	CD	C5	81	0E
8110	0F	3E	08	CD	C5	81	0E	3E
8118	3E	07	CD	C5	81	0E	34	3E
8120	00	CD	C5	81	F3	01	00	0F
8128	0B	79	B0	20	FB	21	00	9E
8130	06	80	5E	23	56	3E	33	12
8138	D5	E5	21	10	00	19	0E	30
8140	11	88	04	7E	A3	E5	21	00
8148	9D	6F	7E	E1	CD	7A	81	CD
8150	84	81	CB	0B	15	20	1D	23
8158	0D	20	E5	CD	8B	81	E1	23
8160	D1	AF	12	10	CD	18	B6	E1
8168	E1	E1	E1	0E	00	3E	00	CD
8170	C5	81	FB	C9	00	AF	20	FC
8178	18	C9	C5	4F	3E	00	CD	C5
8180	81	F3	C1	C9	C5	06	26	10
8188	FE	C1	C9	C5	3E	00	0E	34
8190	CD	C5	81	F3	CD	E3	81	FE
8198	DF	28	CC	FE	7F	20	1B	3A
81A0	03	7E	FE	66	28	10	C6	18
81A8	32	03	7E	E5	D5	C5	CD	2A
81B0	7E	C1	D1	E1	18	04	3E	1E
81B8	18	EE	F3	01	FF	02	0B	79
81C0	B0	20	FB	C1	C9	06	F4	ED
81C8	79	06	F6	ED	78	F6	C0	ED
81D0	79	E6	3F	ED	79	06	F4	ED
81D8	49	06	F6	4F	F6	80	ED	79
81E0	ED	49	C9	01	0E	F4	ED	49
81E8	06	F6	ED	78	E6	30	4F	F6
81F0	C0	ED	79	ED	49	04	3E	92
81F8	ED	79	C5	0E	42	06	F6	ED
8200	49	06	F4	ED	78	C1	F5	3E
8208	82	ED	79	05	ED	49	F1	C9
8210	79	05	ED	49	FB	C9	FF	00

TABLE 3

9D00	28	1F	1F	00	1F	00	00	00
9D08	1F	00	00	00	00	00	00	00
9D10	24	1B	00	00	00	00	00	00
9D18	00	00	00	00	00	00	00	00
9D20	24	00	1B	00	00	00	00	00
9D28	00	00	00	00	00	00	00	00
9D30	00	00	00	00	00	00	00	00
9D38	00	00	00	00	00	00	00	00
9D40	24	00	00	00	1B	00	00	00
9D48	00	00	00	00	00	00	00	00
9D50	00	00	00	00	00	00	00	00
9D58	00	00	00	00	00	00	00	00
9D60	00	00	00	00	00	00	00	00
9D68	00	00	00	00	00	00	00	00
9D70	00	00	00	00	00	00	00	00
9D78	00	00	00	00	00	00	00	00
9D80	24	00	00	00	00	00	00	00
9D88	1B	00	00	00	00	00	00	00

TABLE 4

9DF0	FF	FF	FF	FF	FF	FF	FF	FF
9DF8	0F	0F	0F	0F	F0	F0	00	00
9E00	00	C0	00	C8	00	D0	00	D8
9E08	00	E0	00	E8	00	F0	00	F8
9E10	50	C0	50	C8	50	D0	50	D8
9E18	50	E0	50	E8	50	F0	50	F8
9E20	A0	C0	A0	C8	A0	D0	A0	D8
9E28	A0	E0	A0	E8	A0	F0	A0	F8
9E30	F0	C0	F0	C8	F0	D0	F0	D8
9E38	F0	E0	F0	E8	F0	F0	F0	F8
9E40	40	C1	40	C9	40	D1	40	D9
9E48	40	E1	40	E9	40	F1	40	F9
9E50	90	C1	90	C9	90	D1	90	D9
9E58	90	E1	90	E9	90	F1	90	F9
9E60	E0	C1	E0	C9	E0	D1	E0	D9
9E68	E0	E1	E0	E9	E0	F1	E0	F9
9E70	30	C2	30	CA	30	D2	30	DA
9E78	30	E2	30	EA	30	F2	30	FA
9E80	80	C2	80	CA	80	D2	80	DA
9E88	80	E2	80	EA	80	F2	80	FA
9E90	D0	C2	D0	CA	D0	D2	D0	DA
9E98	D0	E2	D0	EA	D0	F2	D0	FA

TABLE 5

9EA0	20	C3	20	CB	20	D3	20	DB
9EAS	20	E3	20	EB	20	F3	20	FB
9EB0	70	C3	70	CB	70	D3	70	DB
9EB8	70	E3	70	EB	70	F3	70	FB
9EC0	C0	C3	C0	CB	C0	D3	C0	DB
9ECS	C0	E3	C0	EB	C0	F3	C0	FB
9ED0	10	C4	10	CC	10	D4	10	DC
9ED8	10	E4	10	EC	10	F4	10	FC
9EE0	60	C4	60	CC	60	D4	60	DC
9EE8	60	E4	60	EC	60	F4	60	FC
9EF0	B0	C4	B0	CC	B0	D4	B0	DC
9EF8	B0	E4	B0	EC	B0	F4	B0	FC
9F00	00	C5	00	CD	00	D5	00	DD
9F08	00	E5	00	ED	00	F5	00	FD
9F10	50	C5	50	CD	50	D5	50	DD
9F18	50	E5	50	ED	50	F5	50	FD
9F20	A0	C5	A0	CD	A0	D5	A0	DD
9F28	A0	E5	A0	ED	A0	F5	A0	FD
9F30	F0	C5	F0	CD	F0	D5	F0	DD
9F38	F0	E5	F0	ED	F0	F5	F0	FD
9F40	40	C6	40	CE	40	D6	40	DE
9F48	40	E6	40	EE	40	F6	40	FE
9F50	90	C6	90	CE	90	D6	90	DE
9F58	90	E6	90	EE	90	F6	90	FE
9F60	E0	C6	E0	CE	E0	D6	E0	DE
9F68	E0	E6	E0	EE	E0	F6	E0	FE
9F70	30	C7	30	CF	30	D7	30	DF
9F78	30	E7	30	EF	30	F7	30	FF
9F80	80	C7	80	CF	80	D7	80	DF
9F88	80	E7	80	EF	80	F7	80	FF

EMISSION RECEPTION RTTY

Le logiciel d'émission-réception RTTY proposé ici vous demandera quelques heures de saisie, mais vous disposerez par la suite de la possibilité de trafiquer en radio-télétype.

Le programme est long et fait appel à un savant mélange de BASIC et de langage machine. Lors de son élaboration, nous avons découvert une des limites du BASIC de l'AMSTRAD: sa lenteur d'affichage.

La composition de la mémoire écran de notre machine est très particulière, nous l'avons déjà écrit, et sa gestion ralentit considérablement l'affichage. Nous aurions souhaité proposer un logiciel bénéficiant des possibilités de fenêtrage de l'AMSTRAD, ce qui aurait permis de disposer d'une sorte de 'tableau de bord' indiquant divers états tels que vitesse, shift, position mémoire texte etc. Nous avons dû renoncer au multi-fenêtrage pour pouvoir décoder jusqu'à 110 bauds sans problème. Il est bien entendu que, si nous avons utilisé en interface un UART, l'AMSTRAD aurait été employé comme un vulgaire terminal et le problème résolu ! Nous avons préféré proposer au lecteur un interfacage simple.

PRINCIPALES CARACTERISTIQUES DU PROGRAMME

Il permet de décoder aux vitesses standard de 45.45, 50, 75 et 110 Bds, les messages émis en code Baudot. Le décodage s'effectue via un démodulateur classique en RTTY, la sortie de ce dernier étant reliée à l'entrée imprimante de l'AMSTRAD. Ce choix, criticable nous le savons, a pour but d'éviter la réalisation d'une interface d'entrée. Il est bien entendu que, si le lecteur possède une telle interface, il aura tout intérêt à l'utiliser, les modifications à effectuer sur le logiciel étant minimales. La carte SSTV décrite dans ce livre, pourra aussi être utilisée, le démodulateur RTTY y sera couplé avec avantage. Dans l'hypothèse contraire, la sortie du démodulateur sera reliée aux points 11 (point 'chaud', ligne 'BUSY') et 14 pour la masse.

La réception est complétée d'une mémorisation des caractères recus (les 4095 derniers). Cette mémoire peut être relue dans son entier, ou par 'pages' de 512 caractères. Il est prévu également de pouvoir lister ces pages sur imprimante. Les forçages en mode 'CHIFFRE' ou 'LETTRE' permettent de limiter les effets de brouillages intempestifs. L'écran peut être effacé. Le passage en émission se fera avec les paramètres vitesse et shift déjà choisis, ou encore en modifiant ces paramètres.

Toujours à cause de la lenteur d'affichage, il n'est pas possible, dans cette version du programme, de préparer le texte qui sera émis tout en recevant.

L'émission permet le fonctionnement aux vitesses standards énoncées ci-dessus, le shift étant choisi par l'utilisateur. C'est le générateur sonore de l'AMSTRAD qui produit les tonalités MARK et SPACE, ce qui fera peut-être frémir les puristes mais réjouira tous les autres: pas besoin d'interface AFSK.

Plusieurs messages mémorisés sont prévus. L'utilisateur les définira à sa guise. Ils pourront être interrompus en cours d'émission. Ces messages peuvent être mélangés sans problème avec les textes tapés au clavier. Ceci facilite grandement le trafic, en permettant l'insertion des 'procédures standards'. Il y a ainsi jusqu'à 10 messages mémorisés pouvant atteindre chacun 240 caractères.

Les messages mémorisés se chargent à partir d'un fichier 'à part', lu sur la cassette ou la disquette. Ils peuvent être modifiés ou complétés. Il faut alors sauvegarder la nouvelle version si on désire la conserver.

À l'émission, la sortie BF de l'AMSTRAD (jack stéréo) sera connectée à l'entrée 'Micro' du transceiver. Dans certains cas, il sera peut-être nécessaire d'intercaler un petit préamplificateur: tout dépend de la sensibilité du modulateur de l'émetteur. La commutation émission devra être assurée manuellement... Le relai de commutation des CPC 664 et 6128 (sortant sur la prise magnétophone) auraient pu être utilisés mais la compatibilité avec le 464 n'était plus assurée.

La mémoire de texte atteint 4095 caractères. Il faut l'imaginer comme un 'rouleau'. Lorsque la dernière position mémoire disponible a été remplie, le prochain caractère reçu sera stocké au début de la mémoire, écrasant petit à petit les caractères recus antérieurement. Quand un texte important a été reçu, et que l'utilisateur désire le conserver, il est possible d'effectuer une sauvegarde de toute la mémoire texte sur disque (ou cassette). L'ordinateur demande alors sous quel nom doit se faire la sauvegarde. Il sera possible de récupérer ensuite ce fichier. On le désignera alors par le nom sous lequel il avait été sauvegardé. Les noms seront donnés sur 1 à 8 caractères, l'extension '.BIN' étant attribuée automatiquement par le logiciel, pour les possesseurs de disques.

L'édition de la mémoire peut se faire 'en continu', de la première adresse jusqu'à la dernière, 'partiellement' c'est-à-dire jusqu'au dernier caractère reçu, ou encore en mode 'pages', listant 512 caractères. Les emplacements vides de la mémoire sont remplis par des points.

La position du dernier caractère reçu est indiquée. Bien sur, on peut avoir déjà 'fait un tour', c'est-à-dire que la mémoire a été écrite au moins une fois. Le fait d'indiquer à l'utilisateur la position du dernier caractère mis en mémoire lui permettra de savoir quelles pages il doit lister. Prenons un exemple. Le pointeur indique 184. L'habitude, durée du décodage déjà effectué, ou l'examen de la mémoire en mode 'continu', montrent que celle-ci se remplit pour la énième fois. On listera donc la page 0 et on saura que, pour trouver la partie précédente du texte, il faudra lister les pages 7, 6, etc. Pas facile à expliquer mais vite compris lors de l'utilisation.

La sortie d'un texte sur l'imprimante a été prévue. Bien entendu, cela suppose que l'imprimante soit connectée et en service. C'est là tout l'intérêt de posséder une carte d'entrée, pour libérer le connecteur imprimante de l'AMSTRAD, sinon il faudra 'brancher et débrancher'. Cela peut se faire sans risque, car la sortie imprimante de l'AMSTRAD est 'bufferisée', mais il faut quand même faire extrêmement attention à bien présenter et positionner le connecteur. Une solution de fortune consiste à monter un inverseur à 2 positions sur le fil 'BUSY' du câble imprimante, le commun vers l'entrée de l'AMSTRAD, un point vers la sortie du démodulateur RTTY, l'autre point vers l'imprimante. On basculera ainsi du décodage vers l'impression. L'édition du texte sur imprimante se fait par 'pages' de 512 caractères. Le numéro de la page (0 à 7) est indiqué sur le papier.

LES DIFFERENTES COMMANDES

RECEPTION:

- C force le mode CHIFFRES.
- L force le mode LETTRES.
- M permet d'afficher le contenu de la mémoire.
- E passage en émission.
- T retour au menu pour modifier les paramètres sélectionnés.
- S efface le contenu de l'écran.

EMISSION:

- Les commandes passent par l'appui simultané sur 'CTRL'.
- CTRL R passage en réception.
- CTRL O modification des paramètres choisis.
- CTRL A à J choix de l'un des 10 messages mémorisés.

MEMOIRE:

Un menu particulier est présenté à l'écran pour:
Sauvegarde de la zone mémoire.
Chargement de la zone mémoire.
Consultation complète ou partielle.
Possibilité d'édition sur papier.

QUELQUES COMMENTAIRES SUR LE PROGRAMME

Les listings de la partie 'assembleur' sont commentés pour permettre au lecteur de procéder à d'éventuelles modifications. Le BASIC est composé de plusieurs parties dont nous donnons le role ci-après.

110 Si le programme a déjà été exécuté une fois, on ne perd pas de temps à recharger les messages mémorisés et le langage machine.

120 Entrée du fichier messages mémorisés.

170 Implantation du langage machine à partir des DATAS

230 Initialise la mémoire texte réception.

250 Affiche les différents messages mémorisés et propose leur éventuelle modification.

510 Implante en mémoire, à leur emplacement définitif, les différents messages mémorisés.

590 Pret à commencer. Choix des paramètres vitesse, shift, et nombre de colonnes avant émission auto d'un CR/LF.

610 Aiguillage émission, réception, affichage contenu mémoire.

650 Emission.

760 Réception.

770 Détermine le motif du retour au BASIC: pour lire mémoire, pour modifier les paramètres ou pour passer en émission ?

800 Modification des paramètres et transmission aux routines en langage machine de ces paramètres.

1050 DATAS contenant le langage machine.

1770 Sauvegarde des messages mémorisés.

1790 Traitements de la mémoire texte reçu.

1830 Menu et aiguillage.

1930 Sauvegarde du contenu de la mémoire texte vers le disque ou la cassette.

2010 Chargement de la mémoire à partir d'un texte déjà sauvegardé.

2100 Edition de la mémoire. Calcul de la position du dernier caractère reçu et mémorisation. Menu édition. Dans le mode édition par 'pages', il est possible de sortir les textes sur l'imprimante. La variable 'PRT' renseigne de ce choix.

2530 Effacement complet de la mémoire texte.

2610 Sous-aiguillage 'menu mémoire' ou 'menu principal'.

```

10 ' ***** E/R RTTY *****
20 ' * *
30 ' * (c) E.DUTERTRE FC1EZH *
40 ' * & D.BONOMO F 6GKQ *
50 ' * CPC 464 664 6128 *
60 ' * V.02 27/11/1985 *
70 ' * *
80 ' *****
90 '
100 MEMORY &7FF0:CLS
110 IF PEEK(&9E60)=&26 THEN 590:' IL A D
EJA TOURNE...
120 MODE 1:PRINT"CHARGEMENT DE LA MEMOIR
E 'MESSAGES'"
130 OPENIN"MESSAGE"
140 FOR N=1 TO 10:INPUT#9,ME$(N):NEXT
150 CLOSEIN
160 CLS
170 PRINT:PRINT:PRINT"PATIENTEZ QUELQUES
INSTANTS "
180 RESTORE 1060
190 FOR N=&A100 TO &A290:READ A$:A=VAL("
&" +A$)
200 POKE N,A:NEXT:RESTORE 1480
210 FOR N=&9E00 TO &9F0F:READ A$:A=VAL("
&" +A$)
220 POKE N,A:NEXT
230 ' MISE A ZERO DE LA MEMOIRE TEXTE
240 POKE &9E06,0:POKE &9E07,&80:FOR N=&8
000 TO &8FFF:POKE N,46:NEXT:PRINT CHR$(7
)
250 ' MESSAGES MEMORISES
260 MODE 2:PRINT CHR$(24);"LES MESSAGES
MEMORISES SONT:";CHR$(24)
270 PRINT:FOR N=1 TO 10:PRINT CHR$(24);"
MESSAGE ";CHR$(64+N);CHR$(24);"-";ME$(N)
:FOR M=1 TO 1000:NEXT M,N
280 PRINT:PRINT"VOULEZ--VOUS Y FAIRE DES
MODIFICATIONS (O/N) ?"
290 R$=INKEY$:R$=UPPER$(R$):IF R$="" OR
(R$<>"O" AND R$<>"N") THEN 290
300 IF R$="N" AND T<>0 THEN 430
310 IF R$="N" AND T=0 THEN 490
320 INPUT"LETTRE DU MESSAGE A MODIFIER (
S POUR ARRETER)",NU$
330 NU=ASC(UPPER$(NU$))-64

```

```

340 IF NU=19 THEN 430
350 IF NU<0 OR NU>10 THEN 320
360 CLS:PRINT"ENTREZ LE MESSAGE DE 240 C
ARACTERES MAXIMUM (ENVIRON 3 LIGNES)"
370 PRINT"          Le caractere ^ symbo
lise le CR/LF"
380 INPUT ME$(NU)
390 ME$(NU)=UPPER$(ME$(NU))
400 ME$(NU)=ME$(NU)+CHR$(255)
410 T=T+1
420 GOTO 260
430 CLS:PRINT CHR$(24);"ATTENTION";CHR$(
24);" Si vous voulez conserver ces messa
ges, vous devez les resauvegarder sur di
sque"
440 PRINT:PRINT"SAUVEGARDE (O/N) ?"
450 R$=INKEY$:R$=UPPER$(R$):IF R$="" OR
(R$<>"O" AND R$<>"N") THEN 450
460 IF R$="O" THEN GOSUB 1760
470 PRINT:PRINT:PRINT:PRINT"APPUYEZ SUR
UNE TOUCHE....."
480 IF INKEY$="" THEN 480
490 CLS:PRINT"PATIENTEZ....."
500 FOR I=1 TO 10
510 A=&9000+I*256
520 J=1:C$=""
530 WHILE C$<>CHR$(255)
540 C$=MID$(ME$(I),J,1)
550 IF C$="^" THEN C$=CHR$(13)
560 POKE (A+J-1),ASC(C$)
570 J=J+1
580 WEND:NEXT
590 MODE 2:PRINT CHR$(24);"OPTIONS";CHR$(
24):GOSUB 820
600 CLS#0:PRINT"PARAMETRES : shift";s;"-
RC";col;"col. - vitesse ";v$;" bds"
610 PRINT:PRINT "E pour Emettre":PRINT "
R pour Recevoir":PRINT "M pour Modifier
parametres"
620 K$=UPPER$(INKEY$):IF K$<>"E" AND K$<
>"R" AND K$<>"M" THEN 620
630 IF K$="R" THEN 760
640 IF K$="M" THEN GOSUB 800:MODE 2:GOTO
600:' MODIF. OPTIONS
650 MODE 2:WINDOW #1,1,80,19,25:WINDOW#0
,1,80,1,18

```



```

660 PEN#1,1
670 CLS #0:LOCATE#1,1,1:FOR N=1 TO 80:PR
INT#1, CHR$(154);:NEXT
680 PRINT#1:C$=CHR$(24)+"CTRL"+CHR$(24)
690 PRINT#1,C$;" R-RECEPTION":PRINT#1,C$
;" O-CHANGEMENT OPTIONS":PRINT#1,C$;" A
a J-MESSAGES MEMORISES "
700 LOCATE#1,40,3:PRINT#1,CHR$(24);"OPTI
ONS";CHR$(24)
710 LOCATE#1,40,4:PRINT#1,"VITESSE:";V;"
BDS"
720 LOCATE#1,40,5:PRINT#1,"SHIFT  :";S;"
HZ"
730 CALL &A100
740 CH=PEEK(&ADFA):IF CH=&F THEN GOSUB 8
00:MODE 2:GOTO 600
750 IF CH=13 THEN 1760
760 MODE 2:CALL &9E60
770 IF PEEK(&9E00)=2 THEN 1810:' OPTION
MEMOIRE
780 IF PEEK(&9E00)=0 THEN 650:' EMISSIO
N
790 IF PEEK(&9E00)=1 THEN GOSUB 800:MODE
 2:GOTO 600:' MODIF. OPTIONS
800 MODE 2:CLS
810 PRINT CHR$(24);"          CHANGEMEN
T OPTIONS          ";CHR$(24)
820 PRINT:PRINT:PRINT
830 INPUT"Shift(+/-...Hz)";I
840 IF S=0 AND I=0 THEN 800
850 IF I<>0 THEN S=I
860 PRINT:PRINT
870 PRINT"shift";s;"Hz":PRINT
880 INPUT"Nb de colonnes pour le RC auto
matique (1 a 80)";I
890 IF I=0 AND COL=0 THEN 880 ELSE IF I<
>0 THEN COL=I
900 IF COL>80 THEN 880 ELSE PRINT:PRINT
910 PRINT "RC";COL;"colonnes":PRINT
920 PRINT"Vitesse: 1- 45 Bds":PRINT"
      2- 50 Bds":PRINT"          3- 75
Bds":PRINT"          4-110 Bds":PRINT:PR
INT"votre choix 1,2,3 ou 4 svp ";:INPUT
V
930 TRM1=&9E03:TRL1=&9E02:TRM2=&9E05:TRL
2=&9E04

```

```

940 TEM1=&A235:TEL1=&A234:TEM2=&A22C:TEL
2=&A22B
950 IF V>4 THEN 920
960 ON V GOTO 970,980,990,1000
970 V=45:V$="45":POKE TRM1,5:POKE TRL1,&
A3:POKE TRM2,&B:POKE TRL2,&46:POKE TEM1,
&B:POKE TEL1,&46:POKE TEM2,&10:POKE TEL2
,&E3:GOTO 1010
980 V=50:V$="50":POKE TRM1,5:POKE TRL1,&
20:POKE TRM2,&A:POKE TRL2,&40:POKE TEM1,
&A:POKE TEL1,&40:POKE TEM2,&F:POKE TEL2,
&60:GOTO 1010
990 V=75:V$="75":POKE TRM1,3:POKE TRL1,&
6A:POKE TRM2,&6:POKE TRL2,&D4:POKE TEM1,
6:POKE TEL1,&D4:POKE TEM2,&A:POKE TEL2,&
3E:GOTO 1010
1000 V=110:V$="110":POKE TRM1,2:POKE TRL
1,&17:POKE TRM2,4:POKE TRL2,&2E:POKE TEM
1,4:POKE TEL1,&2C:POKE TEM2,&6:POKE TEL2
,&43:GOTO 1010
1010 POKE &A0FD,COL
1020 IF S>0 THEN POKE &A0FE,INT(62500/12
75):POKE &A0FF,INT(62500/(1275+S)):GOTO
1040
1030 POKE &A0FE,INT(62500/(1275-S)):POKE
&A0FF,INT(62500/1275)
1040 RETURN

1050 REM *****EMISSION*****
1060 DATA CD,8A,BB,AF,32,F9,A0,3A,FD,A0
1070 DATA 32,FC,A0,00,00,00,00,00,00,00
1080 DATA 00,00,00,3E,08,0E,08,CD,34,BD
1090 DATA 3A,FE,A0,4F,3E,00,CD,34,BD,3E
1100 DATA 01,0E,00,CD,34,BD,3E,07,0E,3E
1110 DATA CD,34,BD,3A,F9,A0,A7,CD,CD,06
1120 DATA BB,FE,61,38,02,D6,20,FE,20,DA
1130 DATA D1,A1,F5,3E,00,CD,B4,BB,F1,CD
1140 DATA 5A,BB,F5,CD,8A,BB,F1,FE,20,28
1150 DATA 2B,FE,41,30,13,F5,3A,FB,A0,A7
1160 DATA 20,1F,3E,01,32,FB,A0,1E,1B,CD
1170 DATA A1,A1,18,13,F5,3A,FB,A0,A7,2B
1180 DATA 0C,3E,00,32,FB,A0,1E,1F,CD,A1
1190 DATA A1,18,00,F1,D6,20,21,56,A2,06
1200 DATA 00,4F,09,5E,CD,A1,A1,3A,FC,A0
1210 DATA 3D,32,FC,A0,20,99,3E,0D,C3,02
1220 DATA A2,F3,3A,FF,A0,4F,3E,00,CD,34

```

1230 DATA BD, CD, 33, A2, 16, 05, CB, 1B, 38, 71
1240 DATA 3A, FF, A0, 4F, 3E, 00, CD, 34, BD, CD
1250 DATA 33, A2, 15, 20, ED, 3A, FE, A0, 4F, 3E
1260 DATA 00, CD, 34, BD, CD, 2A, A2, FB, C9, FE
1270 DATA 0D, 28, 2D, FE, 12, 28, 1E, FE, 0F, 28
1280 DATA 1A, FE, 01, DA, 35, A1, FE, 0B, D2, 35
1290 DATA A1, 57, 3E, 01, 32, F9, A0, CD, 3C, A2
1300 DATA AF, 32, F9, A0, C3, 35, A1, 32, FA, A0
1310 DATA 3E, 07, 0E, 3F, CD, 34, BD, C9, F5, CD
1320 DATA 8D, BB, F1, CD, 5A, BB, 3E, 0A, CD, 5A
1330 DATA BB, CD, 8A, BB, 3A, FD, A0, 32, FC, A0
1340 DATA 1E, 08, CD, A1, A1, 1E, 02, CD, A1, A1
1350 DATA C3, 35, A1, 3A, FE, A0, 18, 8D, 01, 60
1360 DATA 0F, 0B, 78, B1, 20, FB, C9, 01, 40, 0A
1370 DATA 0B, 78, B1, 20, FB, C9, 21, 00, 90, 7A
1380 DATA 84, 67, 7E, FE, FF, C8, E5, CD, 3D, A1
1390 DATA CD, 1B, BB, E1, D8, 23, 7D, FE, FF, C8
1400 DATA 18, EC, 04, 00, 05, 00, 00, 1A, 00, 05
1410 DATA 0F, 12, 09, 11, 0C, 03, 1C, 1D, 16, 17
1420 DATA 13, 01, 0A, 10, 15, 07, 06, 18, 0E, 00
1430 DATA 00, 1E, 00, 19, 00, 03, 19, 0E, 09, 01
1440 DATA 0D, 1A, 14, 06, 0B, 0F, 12, 1C, 0C, 18
1450 DATA 16, 17, 0A, 05, 10, 07, 1E, 13, 1D, 15
1460 DATA 11
1470 REM *****RECEPTION*****
1480 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
1490 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
1500 DATA 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
1510 DATA 00, 00, 20, 54, 0A, 4F, 20, 48, 4E, 4D
1520 DATA 0D, 4C, 52, 47, 49, 50, 43, 56, 45, 5A
1530 DATA 44, 42, 53, 59, 46, 58, 41, 57, 4A, FF
1540 DATA 55, 51, 4B, FE, 00, 35, 0A, 39, 20, 48
1550 DATA 2C, 2E, 0D, 29, 34, 25, 38, 30, 3A, 3D
1560 DATA 33, 2B, 2A, 3F, 22, 36, 45, 2F, 2D, 32
1570 DATA 07, FF, 37, 31, 28, FE, 26, 00, CD, 1B
1580 DATA BB, DC, BA, 9E, CD, F4, 9E, 30, F5, CD
1590 DATA 1B, BB, DC, BA, 9E, CD, F4, 9E, 38, F5
1600 DATA 16, 05, ED, 4B, 02, 9E, 0B, 78, B1, 20
1610 DATA FB, 2E, 00, ED, 4B, 04, 9E, 0B, 78, B1
1620 DATA 20, FB, CD, F4, 9E, CB, 15, 15, 20, EF
1630 DATA 11, 20, 9E, 7D, 84, E5, 6F, 26, 00, 19
1640 DATA 7E, E1, FE, FF, 20, 04, 26, 20, 18, B8
1650 DATA FE, FE, 20, 04, 26, 00, 18, B0, CD, 5A
1660 DATA BB, CD, FC, 9E, 18, A8, FE, 61, 38, 02
1670 DATA D6, 20, FE, 43, 20, 03, 26, 20, C9, FE
1680 DATA 4C, 20, 03, 26, 00, C9, FE, 53, 20, 06

```

1690 DATA 3E,0C,CD,5A,BB,C9,FE,4D,20,05
1700 DATA 3E,02,C3,EF,9E,FE,45,20,05,3E
1710 DATA 00,C3,EF,9E,FE,54,CO,3E,01,32
1720 DATA 00,9E,E1,C9,01,32,F5,ED,78,17
1730 DATA 17,C9,E5,2A,06,9E,77,23,7C,FE
1740 DATA 90,20,03,21,00,80,22,06,9E,E1
1750 DATA C9,00,00,00,00,00,00,00,00,00
1760 REM
1770 OPENOUT"MESSAGE":FOR N=1 TO 10:WRIT
E#9,ME$(N):NEXT:CLOSEOUT
1780 RETURN
1790 'MANIPULATIONS MEMOIRE
1800 '
1810 FOR n=1 TO 7:PRINT :NEXT:WINDOW #1,
1,80,19,25:WINDOW #0,1,80,1,18
1820 '
1830 CLS #1
1840 LOCATE #1,30,1:PRINT #1,"Options me
moire"
1850 LOCATE #1,30,2:PRINT #1,"-----
-----"
1860 LOCATE #1,10,3: PRINT #1," 1- Sauve
garder la memoire          2- Charger la m
emoire"
1870 LOCATE #1,10,4:PRINT #1," 3- Lire /
Imprimer la memoire      4- Effacer la me
moire"
1880 LOCATE #1,30,7:PRINT #1,"Votre choi
x ->?"
1890 t$="":WHILE t$="":t$=INKEY$:WEND
1900 IF ASC(t$)<49 OR ASC(t$)>52 THEN 18
90
1910 LOCATE #1,44,7:PRINT #1,t$
1920 t=VAL(t$):ON t GOTO 1930,2010,2100,
2510
1930 'SAUVEGARDE DE LA MEMOIRE
1940 '
1950 CLS #1:LOCATE #1,30,1:PRINT #1,"Sau
vegarde de la memoire"
1960 PRINT CHR$(7):LOCATE #1,20,3: INPUT
#1,"Titre de la sauvegarde (8 car. max.
) ";t$
1970 IF LEN(t$)>8 THEN 1960
1980 LOCATE #1,20,5:PRINT #1,"Preparer l
a disquette et presser une touche"

```

```

1990 WHILE INKEY$="":WEND:POKE &7FFE,PEE
K(&9E06):POKE &7FFF,PEEK(&9E07):SAVE t$,
b,&7FFE,&1001
2000 GOSUB 2610:ON t GOTO 2600,1830
2010 'CHARGEMENT DE LA MEMOIRE
2020 '
2030 CLS #1:LOCATE #1,30,1:PRINT #1,"Cha
rgement de la memoire"
2040 PRINT CHR$(7):LOCATE #1,20,3: INPUT
#1,"Titre du fichier a charger (8 car.
max.) ";t$
2050 IF LEN(t$)>8 THEN 2040
2060 LOCATE #1,20,5:PRINT #1,"Preparer l
a disquette et presser une touche"
2070 WHILE INKEY$="":WEND:t$=t$+".bin":L
OAD t$
2080 POKE &9E06,PEEK(&7FFE):POKE &9E07,P
EEK(&7FFF)
2090 GOSUB 2610:ON t GOTO 2600,1830
2100 'LECTURE DE LA MEMOIRE
2110 '
2120 debmem=32768:finmem=PEEK(&9E06)+256
*PEEK(&9E07)
2130 dercar=finmem-debmem
2140 CLS #1:LOCATE #1,30,1:PRINT #1,"Lec
ture de la memoire"
2150 LOCATE #1,20,2:PRINT #1,"Le dernier
caractere est en";dercar;"eme position"

2160 PRINT #1,"1- Pour lister jusqu'au d
ernier caractere"
2170 PRINT #1,"2- Pour lister toute la m
emoire"
2180 PRINT #1,"3- Pour lister par pages
de 512 caracteres"
2190 '
2200 t$="":WHILE t$="":t$=INKEY$:WEND
2210 t=VAL(t$):IF t<1 OR t>3 THEN 2200
2220 IF t=1 THEN dercar=32768+dercar
2230 IF t=2 THEN dercar=36863
2240 IF t<3 THEN 2250 ELSE 2330
2250 'AFFICHE LE CONTENU MEMOIRE
2260 '
2270 LOCATE #1,20,6:PRINT #1,"Appuyer su
r une touche pour quitter l'option"
2280 car=0:FOR n=debmem TO dercar

```

```

2290 LOCATE #1,38,7:PRINT #1,car:car=car
+1:PRINT #0,CHR$(PEEK(n));
2300 IF INKEY$<>" THEN n=der car
2310 NEXT:PRINT #0:IF prt THEN PRINT #8
2320 GOSUB 2610:ON t GOTO 2600,1830
2330 'AFFICHE LES n DERNIERS CARACTERES
2340 '
2350 PRINT CHR$(7):CLS #1:LOCATE #1,25,1
:PRINT #1,"AFFICHAGE MEMOIRE PAR PAGES"

2360 LOCATE #1,10,3:INPUT #1,"Choisissez
la page memoire a afficher (0 a 7)";pm
2370 IF pm <0 OR pm>7 THEN 2350
2380 LOCATE #1,10,5:PRINT #1,"Voulez-vous
une copie sur imprimante (O/N) ?"
2390 FOR x=1 TO 35:PRINT "-";NEXT:PRINT
" Page ";pm;:FOR x=1 TO 35:PRINT "-";:
NEXT
2400 t$="":WHILE t$="":t$=INKEY$:WEND:t$
=UPPER$(t$):IF t$<>"0" THEN prt=0 ELSE p
rt=1
2410 liminf=debmem+pm*512:limsup=liminf+
512
2420 IF prt=0 THEN 2430 ELSE PRINT #8,"P
age ";pm:PRINT #8
2430 FOR n=liminf TO limsup
2440 t$=INKEY$:IF t$<>" THEN n=limsup

2450 PRINT #0,CHR$(PEEK(n));:IF prt THEN
PRINT #8,CHR$(PEEK(n));
2460 NEXT:PRINT #0:IF prt THEN PRINT #8

2470 PRINT CHR$(7):CLS #1:LOCATE #1,10,2
:PRINT #1,"Donner le no de la prochaine
page (0 a 7) ou Q pour quitter"
2480 t$="":WHILE t$="":t$=INKEY$:WEND:t$
=UPPER$(t$): IF t$="Q" THEN 2500
2490 pm=VAL(t$):GOTO 2370
2500 GOSUB 2610:ON t GOTO 2600,1830
2510 'EFFACEMENT DE LA MEMOIRE
2520 '
2530 CLS #1:LOCATE #1,30,1:PRINT #1,"EFF
ACEMENT MEMOIRE"
2540 LOCATE #1,5 ,3:PRINT #1,"ATTENTION,
cette option detruit le contenu de la m
emoire reception..."

```

```

2550 LOCATE #1,26,5:INPUT #1,"Confirmez
votre choix (O/N) ";t$:t%=UPPER$(t%)
2560 IF t%<>"O" THEN 1830
2570 PRINT CHR$(7):LOCATE #1,26,5:PRINT
#1,"Effacement en cours. Patientez !"
2580 FOR n=&8000 TO &8FFF:POKE n,46:NEXT
:dercar=0:POKE &9E06,0:POKE &9E07,&80
2590 GOSUB 2610:ON t GOTO 2600,1830
2600 MODE 2:GOTO 600
2610 CLS #1:LOCATE #1,17,4:PRINT #1,"1-
Retour menu principal      2- Retour menu
memoire"
2620 t%="":WHILE t%="":t%=INKEY$:WEND
2630 IF ASC(t%)<>49 AND ASC(t%)<>50 THEN
  2620 ELSE t=VAL(t%)
2640 RETURN

```

ADDITIF AU PROGRAMME RTTY

Il est possible d'ajouter facilement au programme RTTY un sous-programme, permettant l'émission automatique de l'indicatif de la station, en télégraphie, à chaque fin de transmission. Rappelons que le passage de l'indicatif en graphie, à périodes régulières, est obligatoire...

La modification est minime; il suffit d'ajouter au programme BASIC les quelques lignes qui suivent, l'exemple étant donné avec F6GKQ :

AJOUTER UNE LIGNE : 735 GOSUB 3000

```

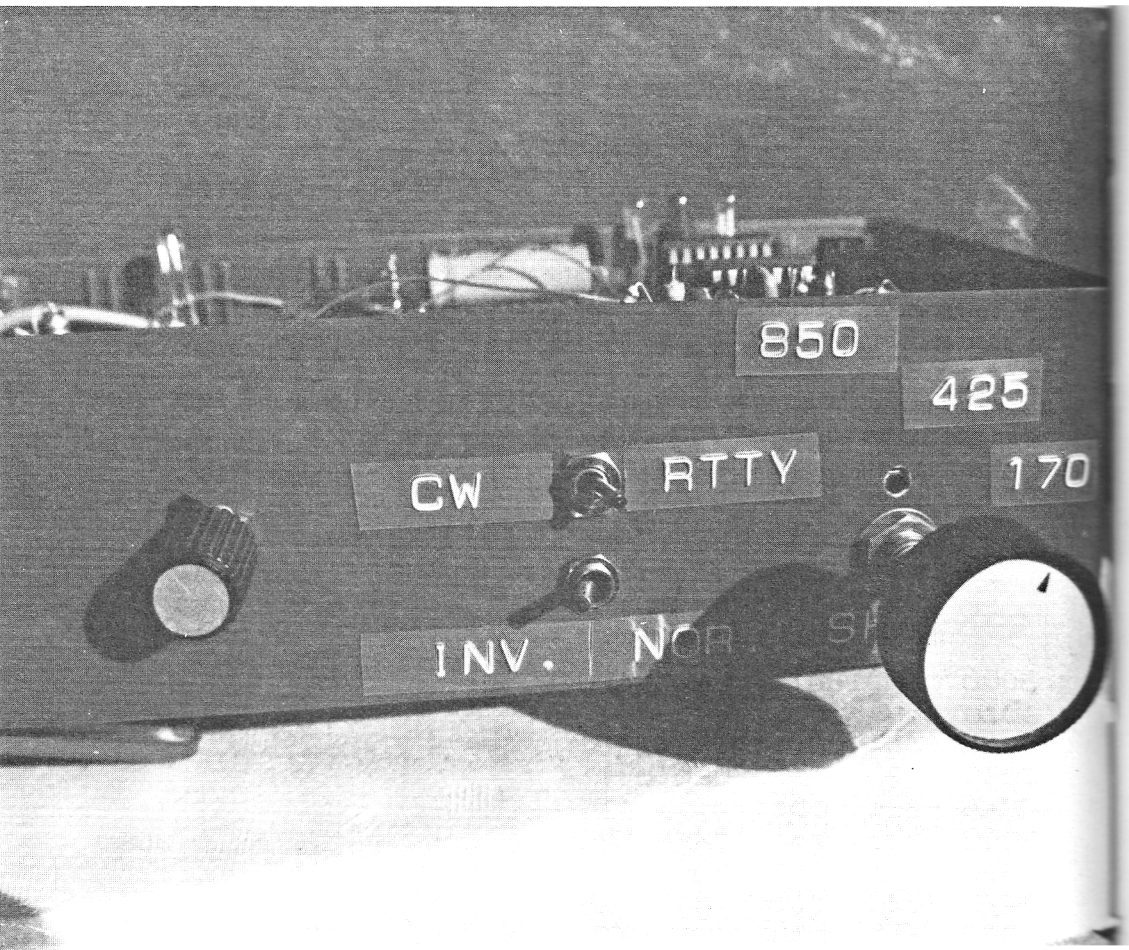
3000 'INDICATIF EN GRAPHIE
3010 cc$="...- . . . . . --. -.- ---.-"
3020 ton=120:tt=15:pt=5
3030 FOR n=1 TO LEN(cc$)
3040 car%=MID$(cc$,n,1)
3050 IF car%=" " THEN SOUND 1,ton,3*pt,0:GOTO 3080
3060 IF car%="." THEN d=pt ELSE d=tt
3070 SOUND 1,ton,d,10:SOUND 1,ton,pt,0
3080 NEXT:RETURN

```

DUE TO THE SIMPLE FACT THAT OUR COUNTRIES ARE NEIGHBOURS.

THE SOVIET LEADER'S REMARKS WERE POSITIVELY ASSESSED IN TOKYO. THE JAPANESE FOREIGN MINISTRY, THE NEWSPAPER YOMIURI REPORTED, +WELCOMED THEM IN PRINCIPLE+. PRIME MINISTER YASUHIRO NAKASONE, AT A PRESS CONFERENCE IN THE NATIONAL PRESS CLUB, ON PROSPECTS OF SOVIET-JAPANESE RELATIONS IN THE LIGHT OF THE SOVIET-AMERICAN SUMMIT IN GENEVA, FOR HIS PART, VOICED THE OPINION THAT +THERE NOW EMERGED A CHANCE OF SETTLING JAPANESE-SOVIET R

EXEMPLE DE RECEPTION ET IMPRESSION D'UNE 'PAGE' MEMOIRE



RTTY PARTIE EMISSION

```

ORG 0A100H
LOAD 0A100H
CALL 0BBBAH      ;Affiche le curseur
XOR A
LD (0A0F9H),A
LD A,(0A0FDH)
LD (0A0FCH),A
NOP              ;On avait prévu des modifs !
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
LD A,B
LD C,B          ;Fixe le volume BF à moitié
CALL 0BD34H
LD A,(0A0FEH)
LD C,A
LD A,0
CALL 0BD34H
LD A,1
LD C,0
CALL 0BD34H     ;Programme le canal A du génér. sonore
LD A,7          ;avec la fréquence du MARK
LD C,3EH
CALL 0BD34H     ;Met en route le canal A
LA135:LD A,(0A0F9H)
AND A
RET NZ
CALL 0BB06H     ;Lit le clavier
LA13D:CP 'a'    ;S'affranchit de l'état de CAPS LOCK
JR C,LA143     ;Si en minuscules, on soustrait
SUB 20H        ;la valeur 20H
LA143:CP 20H   ;Teste si c'est un code de controle
JP C,LA1D1     ;Oui, on saute !
PUSH AF
LD A,0
CALL 0BBR4H     ;Efface, le curseur
POP AF
CALL 0BB5AH     ;Affiche le caractère saisi au clavier

```

```

PUSH AF
CALL DBBBAH      ;Puis, réaffiche le curseur
POP AF
CP 20H          ;Traitement de l'émission d'un ESPACE
JR Z,LA186
CP 'A'
JR NC,LA172     ;Saut à l'émission du code LETTRES
PUSH AF
LD A,(0A0FBH)
AND A
JR NZ,LA185    ;Est-on actuellement en mode CHIFFRES ?
LD A,1         ;Si oui, il est inutile d'émettre le préfixe CHIFFRES
LD (0A0FBH),A
LD E,1BH
CALL LA1A1     ;Mémorisation du mode CHIFFRE et émission du
JR LA185      ;code CHIFFRE
LA172:PUSH AF
LD A,(0A0FBH)
AND A         ;Teste si on est en mode LETTRE
JR Z,LA185    ;Oui, pas la peine d'émettre le préfixe LETTRE
LD A,0
LD (0A0FBH),A
LD E,1FH     ;Mémorisation du mode LETTRE
CALL LA1A1   ;et émission du code LETTRE
JR LA185
LA185:POP AF ;Emission du caractère
LA186:SUB 20H
LD HL,0A256H ;Trouve dans la table de transcodage la correspondance
LD B,0       ;en code Baudot
LD C,A
ADD HL,BC
LD E,(HL)   ;E contient le code Baudot du caractère à émettre
CALL LA1A1  ;On émet !
LD A,(0A0FCH) ;Tient à jour la position colonne
DEC A
LD (0A0FCH),A ;Est-on rendu à la dernière colonne ?
JR NZ,LA135
LD A,0DH    ;Oui, alors on traite le CR/LF
JP LA202
LA1A1:DI    ;Plus d'interruption S.V.P
LD A,(0A0FFH) ;pendant l'émission des tonalités
LD C,A
LD A,0
CALL 0BD34H ;Génère le SPACE
CALL LA233  ;Temporisation
LD D,5     ;Initialise le nombre de moments (5)

```

```

LA1B0:RR E           ;Moment suivant
JR C,LA225
LD A,(0A0FFH)
LA1B7:LD C,A
LD A,0
CALL 0BD34H         ;Emission du SPACE
CALL LA233
DEC D
JR NZ,LA1B0
LD A,(0A0FEH)
LD C,A
LD A,0
CALL 0BD34H         ;Emission du MARK
CALL LA22A
EI                 ;Les interruptions ont droit de parole...
RET
LA1D1:CP ODH        ;Traitement des codes de controle
JR Z,LA202
CP 12H
JR Z,LA1F7
CP 0FH
JR Z,LA1F7
CP 1
JP C,LA135
CP 0BH
JP NC,LA135
LD D,A
LD A,1
LD (0A0F9H),A
CALL LA23C         ;Traitement des messages mémorisés
XOR A
LD (0A0F9H),A
JP LA135
LA1F7:LD (0A0FAH),A
LD A,7
LD C,3FH           ;Coupe le générateur sonore
CALL 0BD34H
RET                ;Sortie.
LA202:PUSH AF       ;C'est un CR/LF
CALL 0BB8DH        ;on le traite
POP AF
CALL 0BB5AH
LD A,0AH           ;on l'affiche
CALL 0BB5AH
CALL 0BB8AH        ;on met curseur
LD A,(0A0FDH)

```

```

LD (0A0FCH),A
LD E,B
CALL LA1A1 ;on émet le CR/LF
LD E,2
CALL LA1A1
JP LA135
LA225:LD A,(0A0FEH)
JR LA1B7
LA22A:LD BC,0F60H ;Temporisation 1/2 moment
LA22D:DEC BC
LD A,B
OR C
JR NZ,LA22D
RET
LA233:LD BC,0A40H ;Temporisation 1 moment
LA236:DEC BC
LD A,B
OR C
JR NZ,LA236
RET
LA23C:LD HL,9000H ;Récupère le message mémorisé
LD A,D
ADD A,H
LD H,A
LA242:LD A,(HL)
CP 0FFH ;Teste si fin de message mémorisé
RET Z
PUSH HL ;Emission de la chaine de caractères
CALL LA13D ;du message
CALL 0BB1BH ;Test du clavier pour pouvoir interrompre le
POP HL ;message mémorisé
RET C
INC HL
LD A,L
CP 0FFH
RET Z
JR LA242
END

```

;--- RECEPTION RTTY ---

;

ORG 9E00H Adresse d'implantation

LOAD 9E00H

TABLE: EQU 9E20H Adresse de la table de transcodage

TXPARA: EQU 9E00H Utilisée lors du retour au BASIC

TRM1: EQU 9E02H Temporisations fonction de

TRM2: EQU 9E04H la vitesse de décodage

MEMTXT: EQU 9E06H Pointeur mémoire texte reçu

;

;

SETSTR: EQU 08BB4H Installation d'une fenetre

CLAVIE: EQU 08B1BH Lecture du clavier

OUTVDU: EQU 08B5AH Sortie d'un caractère vers l'écran

;

;

PORTIN: EQU 0F532H Adresse du PORT d'entrée (ici, port imprimante)

;

DS 20H Réserve d'octets (variables de travail)

;

;

;--- TABLE DE TRANSCODAGE ---

;

DB 20H,54H,0AH,4FH,20H,48H,4EH,4DH	Table de transcodage
DB 0DH,4CH,52H,47H,49H,50H,43H,56H	établissant la
DB 45H,5AH,44H,42H,53H,59H,46H,58H	relation entre le
DB 41H,57H,4AH,255,55H,51H,4BH,254	code Baudot reçu et le
DB 0DH,35H,0AH,39H,20H,48H,2CH,2EH	code ASCII correspondant
DB 0DH,29H,34H,25H,38H,30H,3AH,3DH	au caractère.
DB 33H,2BH,2AH,3FH,22H,36H,45H,2FH	Cette table sert au
DB 2DH,32H,07H,255,37H,31H,28H,254	moment de l'affichage.

;

;

;--- DEBUT DU PROGRAMME RX ---

;

LD H,0 Force arbitrairement le mode 'LETTRES'

ATTSTP: CALL CLAVIE Test du clavier pendant

CALL C,DECOKB l'attente du STOP

CALL LECINP

JR NC,ATTSTP

ATTSTR: CALL CLAVIE Test du clavier pendant

CALL C,DECOKB l'attente du START

CALL LECINP

JR C,ATTSTR

;

;

LD D,5	Nombre de moments
LD BC,(TRM1)	Initialise valeur de la tempo.
TPOHLF: DEC BC	Attente d'un 1/2 moment
LD A,B	pour centrage sur le bit.
OR C	
JR NZ,TPOHLF	
;	
LD L,D	L contiendra le profil du caractère reçu
CHRTPO: LD BC,(TRM2)	Initialise valeur de la tempo.
TPOINT: DEC BC	Attente d'un moment
LD A,B	pour se centrer.
OR C	
JR NZ,TPOINT	
CALL LECINP	Lecture de l'entrée.
RL L	Mémorise le bit reçu.
DEC D	Décrémente le compteur de moments.
JR NZ,CHRTPO	Si non nul, saut au suivant.
LD DE,TABLE	C'était le dernier, pointe sur la table.
LD A,L	
ADD A,H	Tient compte de l'offset CHIFFRE/LETTRE
PUSH HL	
LD L,A	Transformation Baudot/ASCII
LD H,D	
ADD HL,DE	
LD A,(HL)	Récupère le caractère ASCII
POP HL	
CP OFFH	Teste si code préfixe CHIFFRE
JR NZ,ISPLET	
LD H,20H	Si oui, modifie l'offset
JR ATTSTP	Retour à la lecture entrée.
ISPLET: CP OFEH	Teste si code préfixe LETTRE
JR NZ,AFICAR	Non, donc c'est un caractère 'affichable'.
LD H,D	Modifie l'offset pour passage LETTRES
JR ATTSTP	Retour à la lecture entrée.
AFICAR: CALL OUTVDU	Affiche le caractère.
CALL MEMORY	Met le caractère en mémoire texte.
JR ATTSTP	Retour à lecture entrée.
;	
;	
;	
;	
;	
DECOKB: CP 'a'	Ignore l'étatde CAPS LOCK
JR C,ISITC?	
SUB 20H	
ISITC?: CP 'C'	C pour forcer en CHIFFRES ?

```

JR NZ,ISITL?
LD H,20H      Oui, modifie l'offset
RET
ISITL?: CP 'L'      L pour forcer en LETTRES ?
JR NZ,ISITS?
LD H,0       Oui, modifie l'offset
RET
ISITS?: CP 'S'      S pour effacer l'écran ?
JR NZ,ISITM?
LD A,DCH     Oui, exécute un PRINT CHR*(12) (CLS)
CALL OUTVDU
RET
ISITM?: CP 'M'      M pour lire la mémoire ?
JR NZ,ISITE?
LD A,2       Oui, renseigne le BASIC...
JP SORTIE    et prépare le retour.
ISITE?: CP 'E'      E pour émission
JR NZ,ISITT?
LD A,0       Oui, renseigne le BASIC
JP SORTIE    et prépare le retour.
ISITT?: CP 'T'      T pour modification des paramètres ?
RET NZ
LD A,1       Oui, renseigne le BASIC
SORTIE: LD (TXPARA),A  Retour vers le BASIC ou on lira
POP HL      l'adresse TXPARA pour savoir quoi faire...
RET
;
;
;
;--- LECTURE ENTREE SIGNAL --- (ceci peut etre modifié selon vos besoins)
;
LECINP: LD BC,PORTIN  Lit l'entrée
IN A,(C)    et dépose dans l'accu.
RLA         2 rotations successives
RLA         pour lire la ligne 'BUSY'
RET         de l'imprimante.
;
MEMORY: PUSH HL      Sauvegarde le contexte
LD HL,(MEMTXT)      HL pointe sur le prochain emplacement libre
LD (HL),A           On y range le caractère décodé.
INC HL             Incrémente le pointeur mémoire
LD A,H             Teste si la 'butée' est atteinte
CP 90H            ( &8FFF )
JR NZ,ACTMEM
LD HL,8000H       Oui, replace le pointeur en &8000.

```

ACTMEM: LD (MEMTXT),HL
POP HL
RET

Range le pointeur pour la prochaine fois.
Récupère le contexte.

END

CREATION DES MESSAGES MEMORISES POUR L'EMISSION RTTY

Les messages mémorisés utilisés par l'émission RTTY pourront être initialisés à partir du programme suivant. Il permet l'édition et la modification des messages, ainsi que leur sauvegarde sur disquette (ou sur cassette). Toutes les explications utiles sont données dans le programme, c'est pourquoi il peut sembler long... Rien n'empêche le lecteur un peu paresseux d'imaginer un programme réduit au strict minimum !

Une modification du programme peut aussi consister à remplacer le nom fixe 'MESSAGE', donné à la sauvegarde des messages mémorisés, par un nom introduit à partir d'une ligne 'INPUT'. Ceci permettrait de disposer, sur la même disquette, de messages différents, appropriés au trafic, par exemple: messages pour les VHF, en français, pour le décimétrique, en anglais etc.


```

10 ' CREATION DE MESSAGES
20 '
30 ' POUR L'EMISSION RTTY
40 '
50 INK 1,0:INK 0,13:BORDER 13:MODE 2
60 LOCATE 10,2:PRINT"CREATION DES MESSAG
ES MEMORISES POUR LE PROGRAMME RTTY"
70 LOCATE 10,3:PRINT"-----
-----"
80 PRINT:PRINT"Ce programme va vous perm
etre de creer les messages memorises qu
i seront      utilises dans la partie e
mission du RTTY."
90 PRINT:PRINT"Ces messages doivent teni
r sur 3 lignes, soit 240 caracteres, bie
n qu'un leger depassement puisse etre t
olere."
100 PRINT:PRINT"Lorsque vous voulez voir
 apparaitre un saut de ligne, lors de l'
utilisation      en RTTY, mettez dans vot
re message le caractere ^."
110 PRINT:PRINT"Ce caractere, provoquera
 lors de son emission, un retour chariot
 et un saut      de ligne."
120 PRINT:PRINT"Apres le dernier caracte
re de votre message, appuyez sur <ENTER>
."
130 PRINT:PRINT"Vous pourrez examiner le
s messages un par un et les modifier s'i
ls ne vous      conviennent pas. Ils app
araissent alors a l'ecran comme ils appa
raîtront dans      le programme RTTY."
140 PRINT:PRINT"Il ne vous restera plus
 alors qu'a les sauvegarder sur disque (o
u cassette)."
```

```

150 LOCATE 25,25:PRINT CHR$(24);"APPUYEZ
SUR UNE TOUCHE";CHR$(24)
160 WHILE INKEY#="" :WEND:MODE 2
```

```

170 PRINT:PRINT"Voulez-vous recuperer un
  fichier de messages memorises pour les
  modifier ?":GOSUB 430:IF r$="N" THEN 220
180 PRINT:PRINT"Introduisez la bonne dis
  quette (cassette) et appuyez sur un touc
  he"
190 WHILE INKEY$="":WEND:PRINT
200 OPENIN "MESSAGE":FOR n=1 TO 10:INPUT
  #9,me$(n):NEXT:CLOSEIN:PRINT CHR$(7)
210 MODE 2:PRINT:GOTO 280
220 MODE 2:t=1:nu=1
230 PRINT:PRINT "Message No";nu;"/10":GO
  SUB 460
240 nu=nu+1: IF nu=11 THEN 270
250 PRINT:PRINT"Voulez-vous entrer un au
  tre message (O/N) ?"
260 GOSUB 430:IF r$="N" THEN 280 ELSE 23
  0
270 PRINT:PRINT CHR$(24);"          C'es
  t le 10eme et dernier message, vous ne p
  ouvez en ajouter.";CHR$(24);
280 PRINT:PRINT"Voulez-vous revoir un me
  ssage ?":GOSUB 430
290 IF r$="N" THEN 360
300 CLS:PRINT:INPUT"Entrez le numero (1
  a 10) du message a visualiser";nu
310 IF nu<1 OR nu>10 THEN 300
320 PRINT:GOSUB 510
330 PRINT:PRINT"Voulez-vous modifier ce
  message ?":GOSUB 430
340 IF r$="N" THEN 280
350 PRINT:PRINT "Modification du message
  No";nu:GOSUB 460:GOTO 280
360 CLS:PRINT CHR$(24);"ATTENTION";CHR$(
  24):PRINT:PRINT"Si vous voulez conserver
  ces messages, vous devez les resauvegar
  der sur disque"
370 PRINT:PRINT:PRINT "Preparez la disqu
  ette et..."
380 PRINT:PRINT:PRINT:PRINT"...APPUYEZ S
  UR UNE TOUCHE"
390 IF INKEY$="" THEN 390
400 ' Sauvegarde des messages
410 OPENOUT"MESSAGE":FOR N=1 TO 10:WRITE
  #9,ME$(N):NEXT:CLOSEOUT

```

```

420 PRINT CHR$(7):END
430 R$=INKEY$:R$=UPPER$(R$):IF R$="" OR
(R$<>"O" AND R$<>"N") THEN 430
440 RETURN
450 ' Saisie du message
460 PRINT:PRINT"ENTREZ LE MESSAGE DE 240
CARACTERES MAXIMUM (ENVIRON 3 LIGNES)"
470 PRINT"          Le caractere ^ symbo
lise le CR/LF"
480 PRINT:INPUT ME$(NU):ME$(NU)=UPPER$(M
E$(NU)):ME$(NU)=ME$(NU)+CHR$(255)
490 PRINT:PRINT"La longueur du message e
st de";LEN(me$(nu))-1;"caracteres"
500 RETURN
510 FOR n=1 TO LEN(me$(nu))-1:c$=MID$(me
$(nu),n,1)
520 IF c$<>"^" THEN PRINT c$;ELSE PRINT
530 NEXT:PRINT:RETURN

```

REGLAGE DU DEMODULATEUR RTTY

Le réglage du démodulateur à PLL, utilisé pour le décodage RTTY, sera grandement facilité par l'utilisation d'un oscilloscope. Néanmoins, il est possible d'atteindre rapidement le but en lisant ce qui suit.

1- Utiliser des composants de bonne qualité (résistances et condensateurs) dans l'environnement de la PLL.

2- Pour la mise au point, utiliser le petit programme 'GENETONS' qui permettra, au besoin, d'ajuster la valeur des résistances (il est plus difficile et plus onéreux de jouer sur les condensateurs) R1, R2, R3 du circuit de commutation de shift. La résistance ajustable servant à régler l'accord fin du VCO, sera avantageusement placée sur la face avant du boîtier du démodulateur.

3- Le programme 'GENETONS' génère, comme son nom l'indique, des couples de tonalités correspondants aux valeurs de shift généralement rencontrées sur les ondes. Une fréquence fixe, à 1275 Hz est émise en alternance avec les fréquences décalées de 170, 425 et 850 Hz. Il faut souligner que, à cause de la programmation du générateur sonore de l'AMSTRAD, les fréquences produites ne sont pas exactement ce qu'elles devraient être. La tolérance est néanmoins acceptable.

PROCEDURE

Relier la sortie BF (jack stéréo) de l'AMSTRAD à l'entrée du démodulateur RTTY (par un cordon blindé).

Lancer le programme 'GENETONS'. Il génère le couple de fréquences correspondant au shift 170 Hz, soit 1275 et 1453 (au lieu de 1445). Vous pouvez les entendre en ajustant le volume sonore de votre AMSTRAD. Ajuster s'il le faut la valeur de R1, pour obtenir un clignotement régulier des diodes MARK et SPACE, la diode 'PORTEUSE' étant éteinte. Le potentiomètre 'Accord Fin', permet de se centrer dans la boucle de capture.

Un appui sur une touche permettra de générer le couple de fréquences suivant. La procédure est identique. Si vos composants sont de bonnes tolérances, vous n'aurez pas besoin de retoucher au réglage du potentiomètre de l'accord fin. Une légère différence entre les 3 shifts est néanmoins acceptable.

Votre démodulateur est désormais réglé.

```
10 ' GENERATEUR DE TONALITES
20 '
30 ' Diviseur = 62500 / Frequence
40 '
50 ' SOUND Canal,Diviseur,Duree,Volume

60 '
70 MODE 2:LOCATE 35,4:PRINT"GENETONS":LO
CATE 35,5:PRINT"-----"
80 LOCATE 15,7:PRINT"Sortir du programme
par BREAK (2 appuis sur ESC)"
90 LOCATE 11,9:PRINT"Pour changer le cou
ple de frequences appuyer sur <ESPACE>"
100 FOR n=1 TO 3:READ fv(n),ft(n),dv(n),
sh(n):NEXT
110 FOR n=1 TO 3:LOCATE 33,13
120 PRINT "Shift";sh(n);"Hz"
130 LOCATE 15,17:PRINT "Frequences 1275
Hz &";fv(n);"Hz (au lieu de";ft(n);"Hz)"

140 SOUND 1,49,5,15:SOUND 1,dv(n),5,15
150 WHILE INKEY$="":GOTO 140:WEND:NEXT:G
OTO 110
160 DATA 1453,1445,43,170
170 DATA 1689,1700,37,425
180 DATA 2155,2125,29,850
```

UTILISATION D'UNE CARTE D'ENTREE

Nous avons souligné précédemment cette possibilité.

1- Cas d'une carte E/S quelconque.

Votre carte est implantée à l'adresse &Fxxx. Vous disposez de 8 bits en entrée (0 à 7), et vous décidez d'utiliser le bit 7, sur lequel vous cablerez la sortie du démodulateur RTTY. Il faudra donc modifier le programme assembleur de la partie réception comme suit:

```
LD BC, 0FxxxH    (exemple LD BC, 0FBFAH)
IN A, (C)
RLA
NOP
RET
```

Une rotation à gauche amène le bit 7 dans l'indicateur de retenue qui est testé dans la suite du programme. Dans l'exemple donné, votre carte est à l'adresse &FBFAH. Si vous aviez choisi le bit 0, il aurait fallu faire une rotation à droite, RLA, au lieu de RRA. Si vous aviez choisi le bit 1, c'est 2 rotations à droite que vous auriez eu à faire. C'est la raison pour laquelle nous avons prévu un NOP, instruction qui ne sert à rien, et qui permet de conserver au programme son aspect initial, pour faciliter la modification directe des lignes de DATAS, pour les lecteurs ne possédant pas d'assembleur.

La modification dans le BASIC porte sur les lignes 1720 et 1730. La séquence d'origine est 01,32,F5,ED,78,17,17,C9.

32,F5 donnent l'adresse de la carte (ici F532)
17,17 indiquent ici 2 RLA

Voici, dans le cas de l'exemple ci-dessus, adresse FBFA et bit 0, ce qu'il aurait fallu écrire en remplacement:

```
01,FA,FB,ED,78,1F,00,C9
FA,FB remplacent l'adresse F532
1F est le code du RRA qui remplace le premier RLA (17)
00 est le code du NOP qui remplace le second RLA (17)
```

Les autres octets restent inchangés.

2- Cas de l'utilisation de la carte SSTV/FAX.

Cette carte laisse 4 entrées de libres. Ces entrées sont sur les 4 bits de poids fort. Prenons l'exemple du bit (D7, patte 11 du 74LS244). Si vous décidez de cabler la sortie du démodulateur sur ce point, le programme deviendra le suivant:

LD BC, 0FBFFH	Adresse de l'entrée
IN A, (C)	Met dans l'accu
RLA	Passe bit 7 dans retenue
NOP	
RET	et revient

En BASIC, cela donne pour la modification des lignes 1720 et 1730

```
...01,FF,FB,ED,78,17,00,C9...
```

Les autres octets restent inchangés.

3- Utilisation de la carte SSTV/FAX SANS DEMODULATEUR

Cela est possible, en VHF ou en ABSENCE TOTALE DE BROUILLAGE, si la réception est excellente. On peut se passer du démodulateur RTTY et n'utiliser que la carte SSTV/FAX couplée à la carte 'entrée'. Il faut savoir se caler correctement de manière à ce que, seul le BIT 0 de l'entrée clignote au rythme du RTTY. Vous pouvez faire des essais au moyen du programme 'REGLAGES', proposé dans ce livre pour ajuster les comparateurs de la carte SSTV/FAX.

Le programme sera alors modifié comme suit:

```
...01,FF,FB,ED,78,1F,00,C9...
```

Les autres octets sont inchangés.

Le sous-programme assembleur (LECINP) devient:

```
LD BC, 0FBFFH
IN A, (C)
RRA
NOP
RET
```

Un dernier mot, pour ceux qui ne connaissent pas l'assembleur: dans notre listing réception, PORTIN définit l'adresse de l'entrée...

RECEPTION FAC-SIMILE

L'AMSTRAD se prete assez bien à ce genre de réception car il a, en mode 2, une résolution graphique très honorable. Bien sur, le lecteur doit être averti de ce qu'il ne doit pas s'attendre à des résultats comparables à ceux obtenus au moyen de machines mécaniques. Afin de ne pas alourdir le texte, nous n'allons pas décrire en détail les caractéristiques de ce genre de transmission, invitant le lecteur à sa reporter à l'annexe de ce livre ou à un ouvrage spécialisé.

La réception FAX s'effectuera à 120, 180, ou 240 tours/mn, vitesses les plus utilisées dans les transmissions en bandes décamétriques (météo). L'interface utilisée est la même que pour la SSTV, ce qui simplifie largement les choses.

Le programme propose en fait plusieurs options.

RECEPTION AVEC MEMORISATION

L'image recue défile à l'écran. Elle est en même temps mémorisée dans l'ordinateur. Comme la mémoire du CPC, bien que vaste, n'est pas infinie, la réception s'arrêtera dès que celle-ci sera pleine. La capacité de stockage est de 32 KO (de &2000 à &9FFF).

La réception avec mémorisation ne dure en fait que l'équivalent de 800 lignes de balayage, soit environ 6 minutes à 120 tours/mn, 4 mn à 180 t/mn et environ 3 mn à 240 t/mn.

RECEPTION SANS MEMORISATION

L'image défile à l'écran mais n'est pas mémorisée. Ceci permet de faire une réception permanente, ce qui est utile car la transmission de certaines cartes météo demande jusqu'à 15 minutes.

Dans les deux cas de réception, seulement une ligne sur deux est décodée.

Pendant la réception, une image qui penche à droite pourra être redressée en appuyant sur la touche "C", une image qui penche à gauche avec la touche "CLR". Ces inclinaisons ne sont pas dues à un défaut du logiciel mais à la dispersion (léger écart de fréquence) entre les horloges des différents AMSTRAD. Nous verrons un peu plus loin comment minimiser ce défaut en adaptant le logiciel à votre ordinateur. Cette précaution évitera d'ajuster le réglage à chaque utilisation.

Si vous entreprenez la réception d'une image déjà commencée, il est plus que probable qu'elle ne sera pas bien cadrée. Dans ce cas, quelques appuis BREFS sur la touche "CTRL" rétabliront le cadrage correct.

L'arrêt de la réception s'obtient en appuyant sur "SHIFT".

En mode réception continue, 3 possibilités s'offrent alors à vous:

- vous appuyez sur ENTER et revenez au menu principal.
- vous appuyez sur COPY et obtenez une copie de l'image sur imprimante.
- vous appuyez sur une touche numérique et votre image sera sauvegardée sur disquette (ou cassette) sous le nom FAXIMA suivi d'un chiffre de 0 à 9, selon la touche que vous aurez frappée. Exemple: FAXIMA4.

ATTENTION, sur le 464, la cassette doit être positionnée et les touches PLAY et RECORD enfoncées. Sur les AMSTRAD équipés de disques, veillez à ce qu'une disquette soit présente, avec suffisamment de place disponible pour accueillir les 17 KO de chaque image...

En mode réception mémorisée, après l'appui sur "SHIFT" permettant de quitter le décodage, l'appui sur "ESPACE" renvoie au menu. Il est alors possible de sauvegarder l'image ou de l'imprimer.

Pour obtenir une image bien cadrée dès le début, il suffit de démarrer la réception (appui sur ENTER suivant l'option du menu) pendant l'émission des pulses de synchronisation, signal caractéristique que vous reconnaîtrez avec un peu d'habitude... L'image sera alors parfaitement cadrée.

L'option LECTURE permet de récupérer une image sauvegardée par la fonction SAUVEGARDE du menu. ATTENTION, ceci ne s'applique qu'au cas d'une image sauvegardée à partir du menu. Pour une image sauvegardée en sortie de réception "continue", il faudra utiliser l'option 11 du menu et appuyer sur "COPY" pour imprimer l'image ou sur "ESPACE" pour revenir au menu.

L'option IMPRESSION est prévue pour fonctionner avec une imprimante de type EPSON ou compatible. Ce choix a été fait parce que ces imprimantes sont les plus répandues. Voici quelques modèles compatibles:

- AMSTRAD DMP 2000
- EPSON LX et FX
- BROTHER M1009
- CENTRONICS GLP 3101
- SMITH CORONA D100 ou D200
- STAR GEMINI SG 10

Il est à noter que la recopie fonctionne aussi sur la MANNESMAN-TALLY MT808 mais avec une image un peu plus étroite en largeur. L'utilisateur pourra toujours adapter la routine de recopie, implantée à partir de l'adresse &A280, à son type de matériel...

Pour tous ceux qui ne possèdent pas d'imprimante, il est possible de visualiser l'image en mémoire par l'option 10 du menu. On fait alors défiler l'image grace aux touches de montée et descente du curseur.

CONNEXIONS: il faut utiliser la carte interface décrite dans ce livre pour pouvoir decoder. ATTENTION, si vous lancez la réception sans que la carte soit connectée, le programme se bloque...

ADAPTATION DU PROGRAMME A CHAQUE ORDINATEUR

Comme nous l'avons écrit, les écarts entre horloges des différentes machines ont une incidence sur l'image recue. Pour adapter au mieux le programme à votre AMSTRAD, voici le procédure:

- Calez le récepteur correctement sur l'émission et redressez au mieux l'image avec les touches "C" et "CLR".
- Arrêtez alors le programme par "SHIFT" puis Break (2 appuis successifs sur la touche "ESC").
- Tapez au clavier ? PEEK(&A003) et ? PEEK(&A004).
- Reportez les deux valeurs obtenues dans le programme BASIC à la ligne 150 si vous réglez le 120 t/mn, 160 pour le 180, 170 si vous étiez en 240 t/mn. C'est au prix de ce travail un peu fastidieux que vous obtiendrez un programme pleinement opérationnel.
- Sauvegardez alors cette version "personnalisée" du FAX.

Malheureusement, on constate dans certains cas des écarts dans le temps, surtout si votre AMSTRAD est dans un local soumis à d'importantes variations de température. Dans ce cas, il n'y a pas d'autre remède que d'utiliser les touches de "redressement".

Comme tous les programmes comprenant du langage machine, la plus grande rigueur est le maximum d'attention sont exigés...

- 1- Taper le programme BASIC principal et le sauvegarder (SAVE "FAX").
- 2- Effacer par NEW le contenu de la mémoire.
- 3- Taper le chargeur hexadécimal écrit en BASIC.
- 4- Faire RUN et... se reporter à la procédure décrite dans le chapitre SSTV pour introduire le code machine. Vous aurez 3 tables à introduire. La première commence en &A000, la seconde en &A200, la dernière en &A3E7. Ne pas oublier de taper l'arobas (A commercial) après chaque table.
- 5- Quand l'ordinateur affiche SAUVEGARDE, mettre en position la cassette ou la disquette, et sauvegardez cette partie par:

```
SAVE "FAXR.BIN",B,&A000,&577
```

6- Pour recharger le programme, faire RUN"FAX". Le programme BASIC se charge en premier et appelle automatiquement le programme en langage machine.

RECEPTION FAC-SIMILE (FAX)

CE LISTING NE DOIT PAS ETRE SAISI

ORG D0005H	Début d'implantation mémoire
LOAD D0005H	
MASQ: EQU D0001H	
BIT: EQU D0002H	
TEMPO: EQU D0003H	
LD A, B	
LD (BIT), A	
LD A, 80H	
LD (MASQ), A	Prépare le masque de point/octet
DI	Inhibition des interruptions
LD B, 20	
SYNC: CALL LECT	Lecture de la valeur recue à l'entrée
CP 0FFH	
JR NZ, SYNC	Attente de synchronisation (Pulses de synchro)
STSYNC: CALL LECT	
CP 0	
JR NZ, STSYNC	Attend la fin du pulse de synchro
DJNZ SYNC	Compte 20 pulses
LD IY, 2000H	IY pointe adresse début mémoire de stockage
DEBUT: LD HL, (0A575H)	HL pointe adresse début ligne du bas de l'écran
LD B, 80	B contient le nombre d'octets par ligne

REPL:LD A,0	
LD (HL),A	
INC HL	
DJNZ REPL	Efface le ligne du bas de l'écran
LD HL,(0A575H)	HL contient l'adresse de la ligne du bas
LD BC,640	BC contient le nombre de points par ligne
ECR:PUSH BC	
CALL LECT	Lit la valeur recue
PUSH BC	
PUSH HL	
LD HL,MASQ	Positionne le bit dans l'octet considéré
AND (HL)	
LD B,A	
POP HL	
LD A,(HL)	Affiche le nouveau point recu
OR B	
POP BC	
LD (HL),A	
PUSH HL	
LD HL,MASQ	
RRC (HL)	Prépare le masque pour le point suivant
LD (IY+0),A	Mémorise
LD A,(HL)	
AND 80H	
RLCA	
PUSH DE	
LD D,0	Change d'octet si les 8 bits ont été
LD E,A	utilisés dans la mémoire.
ADD IY,DE	
PUSH IY	
POP HL	
LD A,H	
POP DE	
AND A	
CP 0A0H	Teste si fin de mémoire (&A000)
JP Z,FIN	Arret si plus de mémoire
PUSH DE	
LD DE,MASQ	
LD A,(DE)	
AND 80H	
POP DE	
POP HL	Change d'octet si les 8 bits ont été
RLCA	utilisés dans l'écran
ADD A,L	
LD L,A	

XOR A	
ADC A,H	
LD H,A	
PUSH BC	
CALL KEY	Test du clavier
POP BC	
CP 223	Arret si appui sur SHIFT
JP Z,STOP	
CP 127	Recadre si appui sur CTRL
CALL Z,CADRE	
POP BC	
DEC BC	
LD A,B	
OR C	
JR NZ,ECR	Boucle si ce n'est pas le dernier point ligne
LD B,200	
LD IX,0A3E7H	Scrolling écran
SUITE:LD L,(IX+0)	
LD H,(IX+1)	
LD E,(IX+2)	
LD D,(IX+3)	
LD C,80	
LIGN:LD A,(DE)	
LD (HL),A	
INC DE	
INC HL	
DEC C	
JR NZ,LIGN	
PUSH BC	
LD BC,2	
ADD IX,BC	
POP BC	
DJNZ SUITE	
CALL KEY	Test clavier
AND 3	
PUSH HL	
PUSH DE	
LD HL,TOUCH	HL contient l'adresse table des touches
LD D,0	
LD E,A	
ADD HL,DE	
LD E,(HL)	Si appui sur ° diminue la tempo
INC HL	
INC HL	Si appui sur CLR augmente la tempo
INC HL	

INC HL	
LD D, (HL)	
LD HL, (TEMPO)	
ADD HL, DE	
LD (TEMPO), HL	
POP DE	
POP HL	
LD BC, (TEMPO)	Tempo de fin de ligne
T2:DEC BC	
LD A, B	
OR C	
NOP	
NOP	
NOP	
JR NZ, T2	Fin de tempo de fin de ligne
JP DEBUT	Continue
KEY:LD BC, 0F40EH	Lecture clavier
OUT (C), C	
LD B, 0F6H	
IN A, (C)	
AND 30H	
LD C, A	
OR 0C0H	
OUT (C), A	
OUT (C), C	
INC B	
LD A, 92H	
OUT (C), A	
PUSH BC	
LD C, 042H	
LD B, 0F6H	
OUT (C), C	
LD B, 0F4H	
IN A, (C)	
POP BC	
PUSH AF	
LD A, B2H	
OUT (C), A	
DEC B	
OUT (C), C	
POP AF	
RET	
FIN:POP HL	Arret de la réception
STOP:EI	
POP HL	

```

RET
LECT:PUSH BC           Lecture du port de l'interface
PUSH HL
LD BC,0FBFFH
IN A,(C)
AND 0FH
LD HL, TABLE         Va chercher valeur à afficher dans la table
ADD A,L
LD L,A
XOR A
ADC A,H
LD H,A
LD A,(HL)
POP HL
LD B,0FFH            Tempo inter-points
T1:DJNZ T1
POP BC
RET
CADRE:PUSH BC         Tempo de recadrage
LD B,150
T5:DJNZ T5
POP BC
RET
TABLE:DB 0,0,0,0,0,0,255,255,255
DB 255,255,255,255,255,255,255,255
TOUCH:DB 0,255,1,0
DB 0,255,0,0

```

```

ORG 0A200H           Affichage de l'image mémorisée
LOAD 0A200H         par transfert de la zone mémoire vers l'écran
LD IX,2000H        IX contient l'adresse de début de mémoire
ECRAN:CALL EXEC    Transfère mémoire vers l'écran
JR TESKEY          Test du clavier
EXEC:LD HL,0A3E7H
PUSH IX
LD B,200
AFFI:LD C,50H
LD E,(HL)
INC HL              Transfère le contenu de la mémoire dont
LD D,(HL)          le début est donné par IX, dans l'écran
INC HL
OCT:LD A,(IX+0)
LD (DE),A
INC DE
INC IX

```



```

DEC C
JR NZ,OCT
DJNZ AFFI
POP IX
RET
TESKEY:PUSH HL           Test du clavier
PUSH DE
PUSH IX
CALL OBB1BH
POP IX
POP DE
POP HL
JR NC, TESKEY
CP 2DH                   Si appui sur ESPACE, arret
RET Z
CP OFDH                  Si appui sur curseur haut, écran monte
JR Z, MONTE
CP OF1H                  Si appui sur curseur bas, écran descend
JR Z, DESCEND
JR TESKEY
MONTE:PUSH HL           Monte l'image d'une ligne
PUSH IX
POP HL
LD A,H
CP 2DH
JR NZ,OK1
LD A,L
CP 0
JR Z,RETOUR
OK1:PUSH DE
LD DE,50H
XOR A
SBC HL,DE
POP DE
PUSH HL
POP IX
RETOUR:POP HL
JR ECRAN
DESCEND:PUSH HL        Descend l'image d'une ligne
PUSH IX
POP HL
INC HL
LD A,H
CP 60H
JR Z,RETOUR
PUSH DE

```

LD DE,50H
DEC HL
ADD HL,DE
POP DE
PUSH HL
POP IX
JR RETOUR
LD IX,0
CALL EXEC
RET

ORG 0A280H
LOAD 0A280H

Recopie écran sur imprimante
compatible EPSON

LD C,1BH

CALL IMP

Initialise l'imprimante

LD C,40H

CALL IMP

LD C,1BH

CALL IMP

Interligne de 7/72 pouces

LD C,31H

CALL IMP

LD DE,0

LD HL,0

CALL 0B8C9H

Met l'origine graphique en 0,0

LD HL,018FH

HL contient la position Y du point en haut à gauche

LIGNE:LD DE,0

DE contient position X

PUSH HL

PUSH DE

LD C,1BH

CALL IMP

LD C,4CH

CALL IMP

Nombre de points par ligne

LD C,7FH

CALL IMP

LD C,02

CALL IMP

POP DE

POP HL

POINT:CALL 0BB1BH

Arrêt si appui sur une touche

RET C

PUSH HL

PUSH DE

LD C,0

C, registre de transit

LD B,7

B, nombre de points verticaux utiles par passe

BITS:PUSH BC

PUSH DE	
PUSH HL	
CALL 0BBFOH	Lit la couleur du point sélectionné
POP HL	
POP DE	
POP BC	
AND A	
JR NZ,ETEIND	
SCF	
ETEIND:RL C	Mémoire
DEC HL	
DEC HL	
LD A,H	
CP 0FFH	Test de fin d'écran
JR NZ,OUI	
XOR A	
RL C	
RL C	
RL C	
JR OK2	
OUI:DJNZ BITS	Prend bit suivant
OK2:XOR A	
CALL IMP	
POP DE	
POP HL	
INC DE	
LD A,D	
CP 02	
JR NZ,POINT	
LD A,E	
CP 7FH	
JR NZ,POINT	Groupe de bits suivants
LD DE,0EH	
XOR A	
SBC HL,DE	
LD A,H	
CP 0FFH	
JR Z,TERM	Fin
LD C,DAH	
CALL IMP	Retour à la ligne
JR LIGNE	
TERM:LD C,1BH	
CALL IMP	
LD C,41H	
CALL IMP	Interligne à 4/72 pouces

```

LD C,4
CALL IMP
LD C,DAH
CALL IMP
RET
IMP:CALL OBD2EH          S/P d'envoi d'un caractère à l'imprimante
JR C,IMP
LD A,C
CALL OBD2BH
RET
END

```

```

10 MEMORY &1FFF
20 LOAD"!FAXR.BIN"
30 MODE 2
40 LOCATE 27,1:PRINT"RECEPTION FAC-SIMIL
E":LOCATE 27,2:PRINT"-----
--"
50 LOCATE 27,3:PRINT"@ F&GKQ ET FC1EZ
H"
60 PRINT:PRINT" 1-RECEPTION 120 TR/MN A
VEC MEMORISATION DE L'IMAGE":PRINT" 2-R
ECEPTION 180 TR/MN AVEC MEMORISATION DE
L'IMAGE":PRINT" 3-RECEPTION 240 TR/MN A
VEC MEMORISATION DE L'IMAGE"
70 PRINT" 4-RECEPTION 120 TR/MN EN CONT
INU (SANS MEMORISATION)":PRINT" 5-RECEP
TION 180 TR/MN EN CONTINU":PRINT" 6-REC
EPTION 240 TR/MN EN CONTINU":PRINT" 7-S
AUVEGARDE DE L'IMAGE MEMORISEE"
80 PRINT" 8-LECTURE D'UNE IMAGE SUR K7
OU DISQUE (RECEPTION MEMORISEE)"
90 PRINT" 9-IMPRESSIION DE L'IMAGE MEMOR
ISEE (SUR IMP. TYPE EPSON)
100 PRINT" 10-AFFICHAGE DE L'IMAGE MEMOR
ISEE"
110 PRINT" 11-LECTURE / IMPRESSIION D'UNE
IMAGE SUR K7 OU DISQUE (RECEPTION CONTI
NUE)"

```

```

120 PRINT:PRINT"PENDANT LA RECEPTION,'I'
  REDRESSE UNE IMAGE QUI PENCHE A DROITE,
  'CLR' UNE IMAGE QUI PENCHE A GAUCHE.'SHI
FT' SUSPEND LA RECEPTION JUSQU'A L'APPUI
  SUR UNE AUTRE TOUCHE,QUI FAIT REVENIR
  AU MENU.PAR CONTRE,APRES 'SHIFT',L'APPUI
  SUR 'COPY',"
130 PRINT"IMPRIME LE CONTENU DE L'ECRAN.
"
140 REM
150 PRINT:INPUT "VOTRE CHOIX:";V
160 IF V<1 OR V>11 THEN 30
170 IF V=1 THEN POKE &A003,109:POKE &A0
04,119:POKE &A11E,130:GOTO 280
180 IF V=2 THEN POKE &A003,51:POKE &A00
4,54:POKE &A11E,65:GOTO 280
190 IF V=3 THEN POKE &A003,33:POKE &A004
,21:POKE &A11E,33:GOTO 280
200 IF V=4 THEN POKE &A052,0:V=1:GOTO 17
0
210 IF V=5 THEN POKE &A052,0:V=2:GOTO 18
0
220 IF V=6 THEN POKE &A052,0:V=3:GOTO 19
0
230 IF V=7 THEN GOTO 350
240 IF V=8 THEN GOTO 370
250 IF V=9 THEN GOTO 390
260 IF V=10 THEN CALL &A200:GOTO 30
270 IF V=11 THEN GOTO 430
280 CALL &A005:POKE &A052,&80
290 k$=INKEY$:IF k$="" THEN 290
300 IF ASC(k$)=13 THEN 30
310 IF ASC(k$)=224 THEN CALL &A280:PRINT
#8,CHR$(64):PRINT#8:GOTO 30
320 IF ASC(k$)<&30 OR ASC(k$)>&39 THEN 2
90
330 n$="FAXIMA"+k$:SAVE n$,b,&C000,&4000
340 GOTO 30
350 MODE 2:INPUT"NOM DE L'IMAGE (8CAR. M
AX)";N$
360 SAVE N$,B,&2000,&8000:GOTO 30
370 MODE 2:INPUT"NOM DE L'IMAGE (8CAR. M
AX)";N$
380 LOAD N$:GOTO 30
390 POKE &A273,0:POKE &A274,&20:CALL &A2
71

```

```

400 CALL &A280
410 POKE &A273,&80:POKE &A274,&5E:CALL &
A271
420 CALL &A280:PRINT#8,CHR$(64):PRINT#8:
GOTO 30
430 MODE 2:PRINT:PRINT"L'IMAGE VA SE CHA
RGER, POUR REVENIR AU MENU <ESPACE> POUR
L'IMPRIMER <COPY>"
440 PRINT:PRINT"POSITIONNER LA CASSETTE
OU LA DISQUETTE"
450 PRINT:INPUT"NOM DE L'IMAGE (BCAR, MA
X)";N$:MODE 2:LOAD N$
460 K$=INKEY$:IF K$="" THEN 460
470 IF ASC(K$)=224 THEN 420 ELSE 30

```

```

10 REM*****CHARGEUR LANG. MACHINE*****
20 MEMORY &9FFF
30 L=&A000:GOSUB 100
40 L=&A200:GOSUB 100
50 L=&A3E7:GOSUB 100
60 CLS:PRINT"SAUVEGARDE":PRINT:PRINT"APP
UYEZ SUR UNE TOUCHE"
70 IF INKEY$="" THEN 70
80 SAVE"FAXR.BIN",B,&A000,&577
90 END
100 CLS
110 J=L
120 PRINT HEX$(L);". ";CHR$(8);
130 FOR T=1 TO 8
140 N=1:A$=""
150 K$=INKEY$:IF K$="" THEN 150
160 K$=UPPER$(K$)
170 IF ASC(K$)=127 THEN PRINT CHR$(17):P
RINT CHR$(11);:L=J:GOTO 120
180 IF K$<"0" OR K$>"F" THEN 150
190 IF K$>"9" AND K$<"@" THEN 150
200 IF K$="@" THEN RETURN
210 PRINT K$;". ";CHR$(8);
220 A$=A$+K$
230 N=N+1:IF N<>3 THEN 150
240 A=VAL("&"+A$):POKE L,A:L=L+1
250 PRINT". ";CHR$(8);:NEXT T
260 PRINT":GOTO 110

```

A000	3E	40	08	72	3C	3E	08	32
A008	02	A0	3E	80	32	01	A0	F3
A010	06	14	CD	0A	A1	FE	FF	20
A018	F9	CD	0A	A1	FE	00	20	F9
A020	10	F0	FD	21	00	20	2A	75
A028	A5	06	50	3E	00	77	23	10
A030	FA	2A	75	A5	01	80	02	C5
A038	CD	0A	A1	C5	E5	21	01	A0
A040	A6	47	E1	7E	B0	C1	77	E5
A048	21	01	A0	CB	0E	FD	77	00
A050	7E	E6	80	07	D5	16	00	5F
A058	FD	19	FD	E5	E1	7C	D1	A7
A060	FE	A0	CA	06	A1	D5	11	01
A068	A0	1A	E6	80	D1	E1	07	85
A070	6F	AF	8C	67	C5	CD	D9	A0
A078	C1	FE	DF	CA	07	A1	FE	7F
A080	CC	23	A1	C1	0B	78	B1	20
A088	AE	06	C8	DD	21	E7	A3	DD
A090	6E	00	DD	66	01	DD	5E	02
A098	DD	56	03	0E	50	1A	77	13
A0A0	23	00	20	F9	C5	01	02	00
A0A8	DD	09	C1	10	E2	CD	D9	A0
A0B0	E6	03	E5	D5	21	3A	A1	16
A0B8	00	5F	19	5E	23	23	23	23
A0C0	56	2A	03	A0	19	22	03	A0
A0C8	D1	E1	ED	4B	03	A0	0B	78
A0D0	B1	00	00	00	20	F8	C3	26
A0D8	A0	01	0E	F4	ED	49	06	F6
A0E0	ED	78	E6	30	4F	F6	C0	ED
A0E8	79	ED	49	04	3E	92	ED	79
A0F0	C5	0E	42	06	F6	ED	49	06
A0F8	F4	ED	78	C1	F5	3E	82	ED
A100	79	05	ED	49	F1	C9	E1	FB
A108	E1	C9	C5	E5	01	FF	FB	ED
A110	78	E6	0F	21	2A	A1	85	6F
A118	AF	8C	67	7E	E1	06	FF	10
A120	FE	C1	C9	C5	06	96	10	FE
A128	C1	C9	00	00	00	00	00	00
A130	FF	FF	FF	FF	FF	FF	FF	FF
A138	FF	FF	00	FF	01	00	00	FF

TABLE 1

A200	DD	21	00	20	CD	09	A2	18
A208	1C	21	E7	A3	DD	E5	06	C8
A210	0E	50	5E	23	56	23	DD	7E
A218	00	12	13	DD	23	0D	20	F6
A220	10	EE	DD	E1	C9	E5	D5	DD
A228	E5	CD	1B	BB	DD	E1	D1	E1
A230	30	F3	FE	20	C8	FE	F0	28
A238	06	FE	F1	28	1E	18	E6	E5
A240	DD	E5	E1	7C	FE	20	20	05
A248	7D	FE	00	28	0B	D5	11	50
A250	00	AF	ED	52	D1	E5	DD	E1
A258	E1	18	A9	E5	DD	E5	E1	23
A260	7C	FE	60	28	F3	D5	11	50
A268	00	2B	19	D1	E5	DD	E1	18
A270	E7	DD	21	00	00	CD	09	A2
A278	C9	00	00	00	00	00	00	00
A280	0E	1B	CD	1E	A3	0E	40	CD
A288	1E	A3	0E	1B	CD	1E	A3	0E
A290	31	CD	1E	A3	11	00	00	21
A298	00	00	CD	C9	BB	21	8F	01
A2A0	11	00	00	E5	D5	0E	1B	CD
A2A8	1E	A3	0E	4C	CD	1E	A3	0E
A2B0	7F	CD	1E	A3	0E	02	CD	1E
A2B8	A3	D1	E1	CD	1B	BB	D8	E5
A2C0	D5	0E	00	06	07	C5	D5	E5
A2C8	CD	F0	BB	E1	D1	C1	A7	20
A2D0	01	37	CB	11	2B	2B	7C	FE
A2D8	FF	20	09	AF	CB	11	CB	11
A2E0	CB	11	18	02	10	DF	AF	CD
A2E8	1E	A3	D1	E1	13	7A	FE	02
A2F0	20	C9	7B	FE	7F	20	C4	11
A2F8	0E	00	AF	ED	52	7C	FE	FF
A300	28	07	0E	0A	CD	1E	A3	18
A308	97	0E	1B	CD	1E	A3	0E	41
A310	CD	1E	A3	0E	04	CD	1E	A3
A318	0E	0A	CD	1E	A3	C9	CD	2E
A320	BD	38	FB	79	CD	2B	BD	C9

TABLE 2

A3E7	00	C0	00	C8	00	D0	00	D8
A3EF	00	E0	00	E8	00	F0	00	F8
A3F7	50	C0	50	C8	50	D0	50	D8
A3FF	50	E0	50	E8	50	F0	50	F8
A407	A0	C0	A0	C8	A0	D0	A0	D8
A40F	A0	E0	A0	E8	A0	F0	A0	F8
A417	F0	C0	F0	C8	F0	D0	F0	D8
A41F	F0	E0	F0	E8	F0	F0	F0	F8
A427	40	C1	40	C9	40	D1	40	D9
A42F	40	E1	40	E9	40	F1	40	F9
A437	90	C1	90	C9	90	D1	90	D9
A43F	90	E1	90	E9	90	F1	90	F9
A447	E0	C1	E0	C9	E0	D1	E0	D9
A44F	E0	E1	E0	E9	E0	F1	E0	F9
A457	30	C2	30	CA	30	D2	30	DA
A45F	30	E2	30	EA	30	F2	30	FA
A467	80	C2	80	CA	80	D2	80	DA
A46F	80	E2	80	EA	80	F2	80	FA
A477	00	C2	00	CA	00	D2	00	DA
A47F	00	E2	00	EA	00	F2	00	FA
A487	20	C3	20	CB	20	D3	20	DB
A48F	20	E3	20	EB	20	F3	20	FB
A497	70	C3	70	CB	70	D3	70	DB
A49F	70	E3	70	EB	70	F3	70	FB
A4A7	C0	C3	C0	CB	C0	D3	C0	DB
A4AF	C0	E3	C0	EB	C0	F3	C0	FB
A4B7	10	C4	10	CC	10	D4	10	DC
A4BF	10	E4	10	EC	10	F4	10	FC
A4C7	60	C4	60	CC	60	D4	60	DC
A4CF	60	E4	60	EC	60	F4	60	FC
A4D7	B0	C4	B0	CC	B0	D4	B0	DC
A4DF	B0	E4	B0	EC	B0	F4	B0	FC
A4E7	00	C5	00	CD	00	D5	00	DD
A4EF	00	E5	00	ED	00	F5	00	FD
A4F7	50	C5	50	CD	50	D5	50	DD
A4FF	50	E5	50	ED	50	F5	50	FD
A507	A0	C5	A0	CD	A0	D5	A0	DD
A50F	A0	E5	A0	ED	A0	F5	A0	FD
A517	F0	C5	F0	CD	F0	D5	F0	DD

A51F	F0	E5	F0	ED	F0	F5	F0	FD
A527	40	C6	40	CE	40	D6	40	DE
A52F	40	E6	40	EE	40	F6	40	FE
A537	90	C6	90	CE	90	D6	90	DE
A53F	90	E6	90	EE	90	F6	90	FE
A547	E0	C6	E0	CE	E0	D6	E0	DE
A54F	E0	E6	E0	EE	E0	F6	E0	FE
A557	30	C7	30	CF	30	D7	30	DF
A55F	30	E7	30	EF	30	F7	30	FF
A567	80	C7	80	CF	80	D7	80	DF
A56F	80	E7	80	EF	80	F7	80	FF
A577	00	18	3C	7E	FF	18	18	18

TABLE 3

Les recopies d'écran ont été effectuées sur une imprimante MANNESMAN-TALLY MT80S

Les figures 1 à 4 sont des recopies de réceptions 'non mémorisées' (voir texte).
Les documents ont été retournés de 90 degrés par rapport à la visu écran.

En règle générale, il ne sera pas aisé à l'amateur non averti, d'interpréter les émissions reçues.

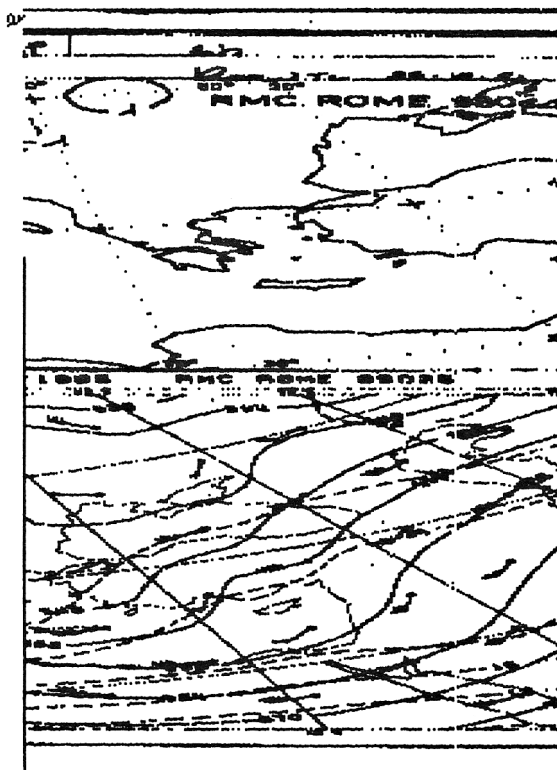


Figure 1: Sur ROME 8.146 khz.

A l'extreme gauche, on distingue le talon de la botte italienne. Au centre, l'ile de CRETE.

Noter la présence de 2 images: carte et fond de carte.

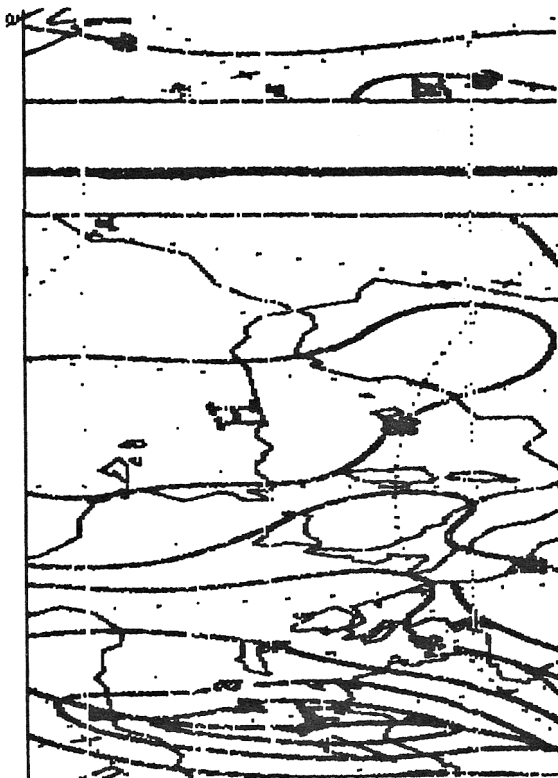


Figure 2: Sur ROME 8.146 khz.

L'Espagne, la Corse, la Sardaigne, la botte italienne et la Sicile, l'Afrique du Nord servent de fond à ces lignes 'iso'. Au 1/4 gauche de la carte apparait une ligne noire épaisse: ici, on n'avait pas démarré la réception sur les pulses de synchro...

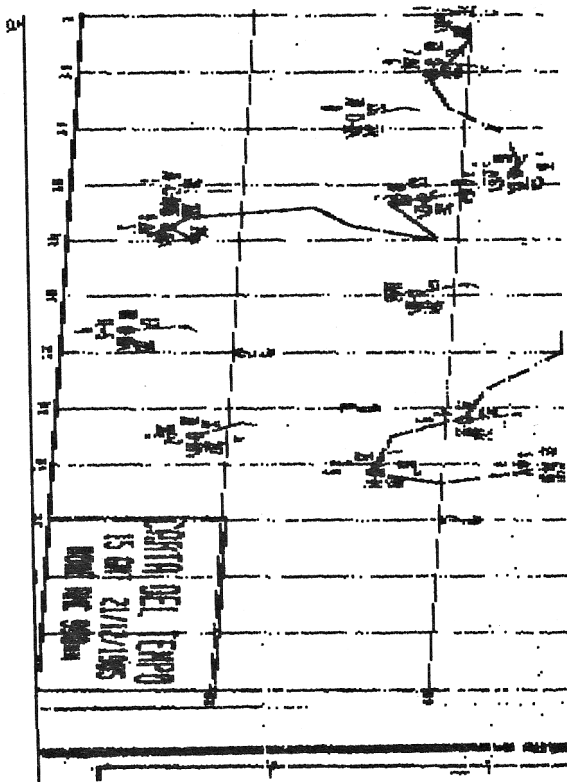


Figure 3: Sur ROME 8.146 khz.

Un autre type de graphique, plus abstrait...

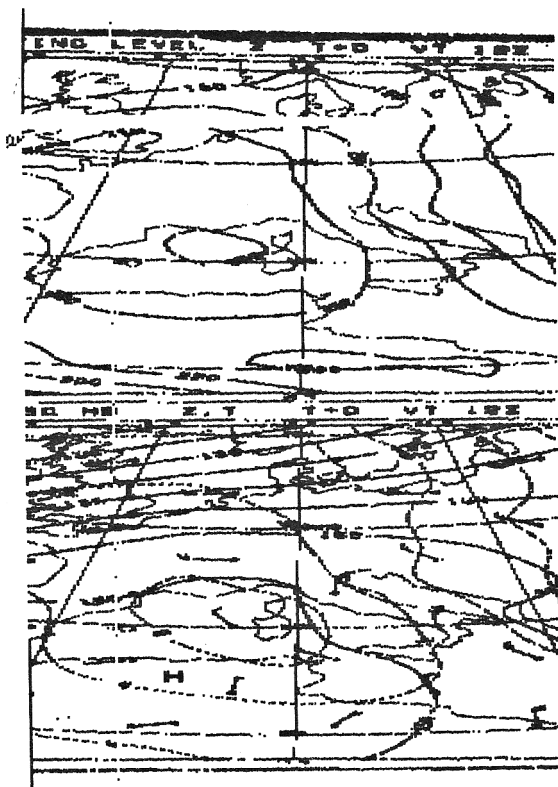


Figure 4: Sur ROME B.146 khz.

La Méditerranée. Repérez-vous grace à la Corse et la Sardaigne, au centre de la carte. Noter ici les 2 cartes...

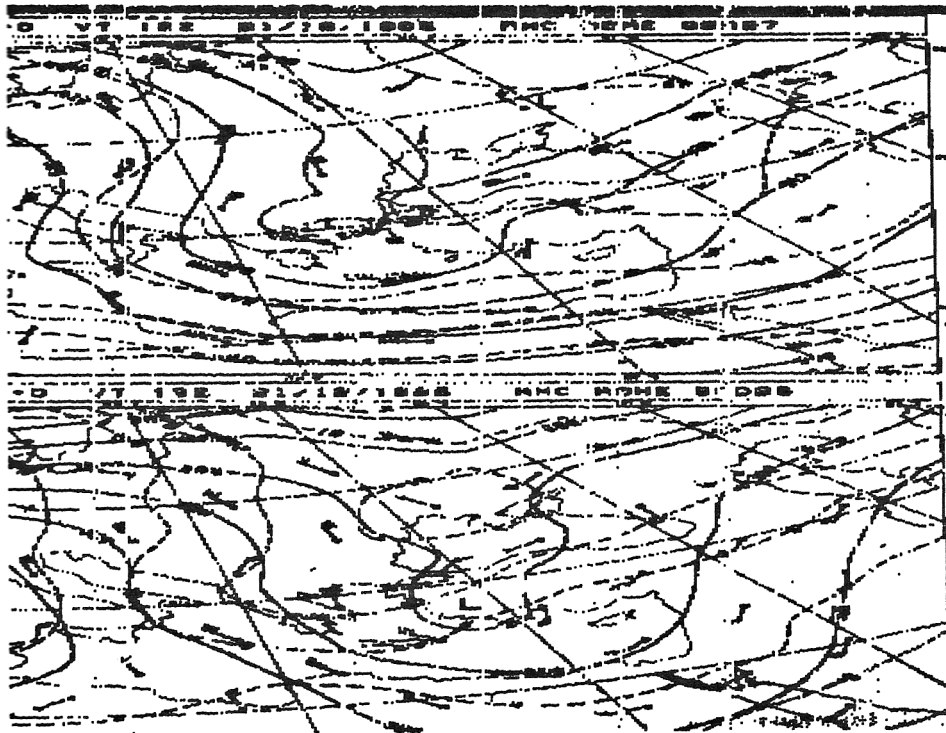


Figure 5: Sur ROME B.146 khz.

Cette carte est d'un format différent car elle émane de la copie du contenu de la mémoire (réception avec mémorisation). Malgré tout, il manque toute la moitié gauche, à l'ouest du méridien de Greenwich. On voit ici tout l'est du bassin méditerranéen. Le 1er 1/4 gauche montre le pied de la botte italienne et la Sicile. A droite, CHYPRE, les cotes de Syrie, Turquie etc...

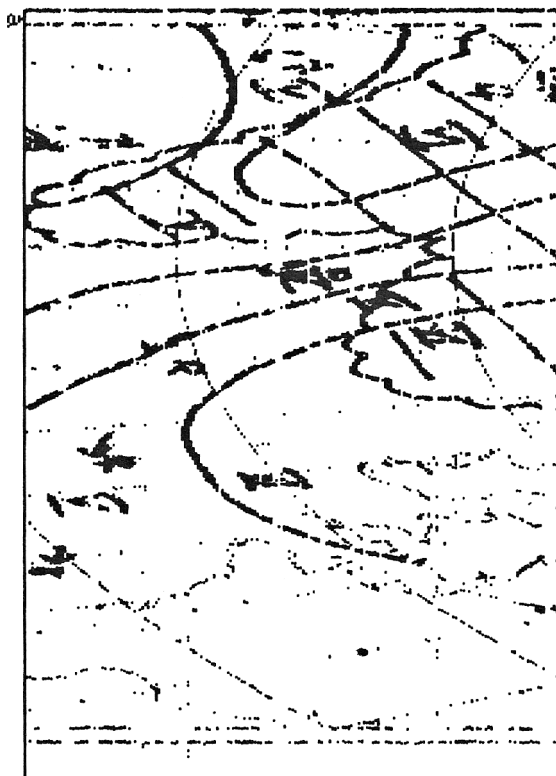


Figure 6: Sur LONDRES 4.248 khz.

Morceau de carte représentant la hauteur des vagues et la houle (Waves and Swell). Nous avons sélectionné l'extrait situé entre les 45ème et 55ème parallèles. On peut voir, sur la droite, l'Irlande, la Grande-Bretagne, la France et la naissance de l'Espagne.



Figure 7: Sur 138 khz

On peut aussi recevoir des images de ce genre, mais le logiciel de décodage n'est pas prévu pour traduire les divers niveaux de gris d'une photo. Bonjour Monsieur l'Inconnu !

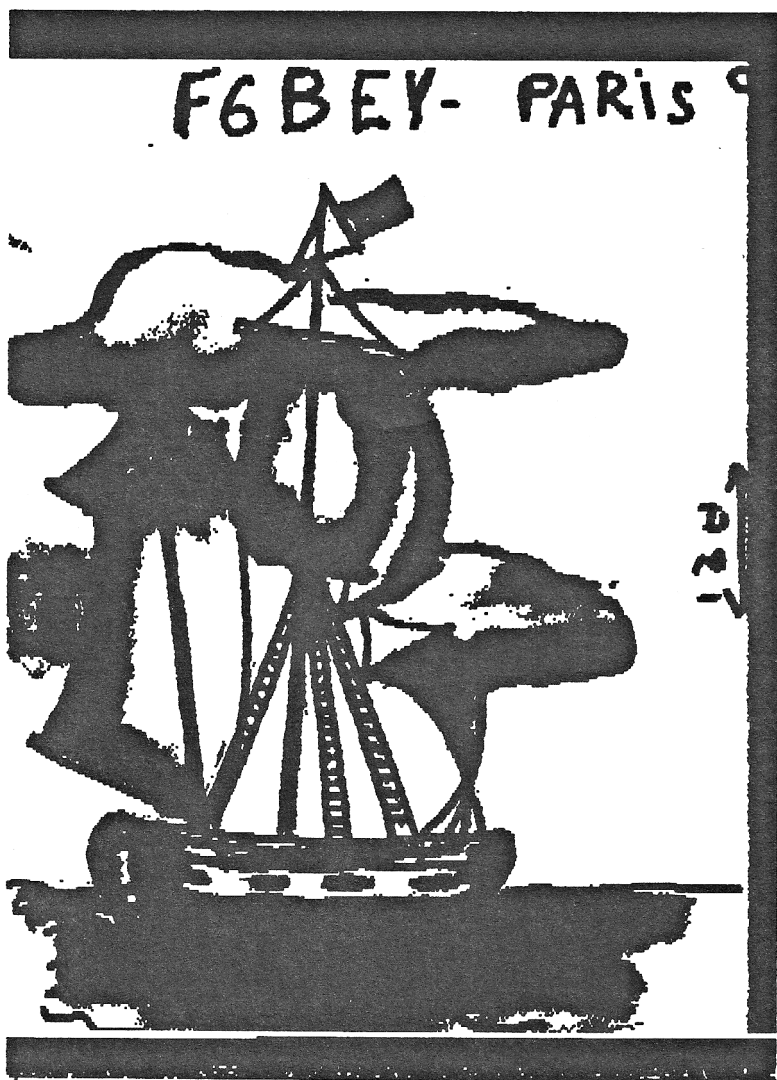


Figure 8: Sur 144 Mhz...

Le fac-similé commence à susciter de plus en plus d'intérêt chez les radioamateurs. Ici, F6BEY, de Paris, transmettait une image à F1EZH, pour l'aider à mettre au point le logiciel de décodage proposé dans ce livre...

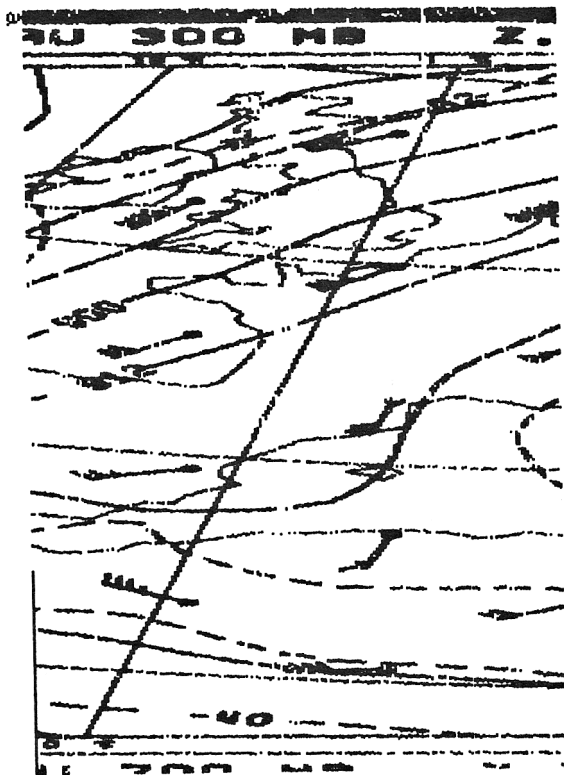
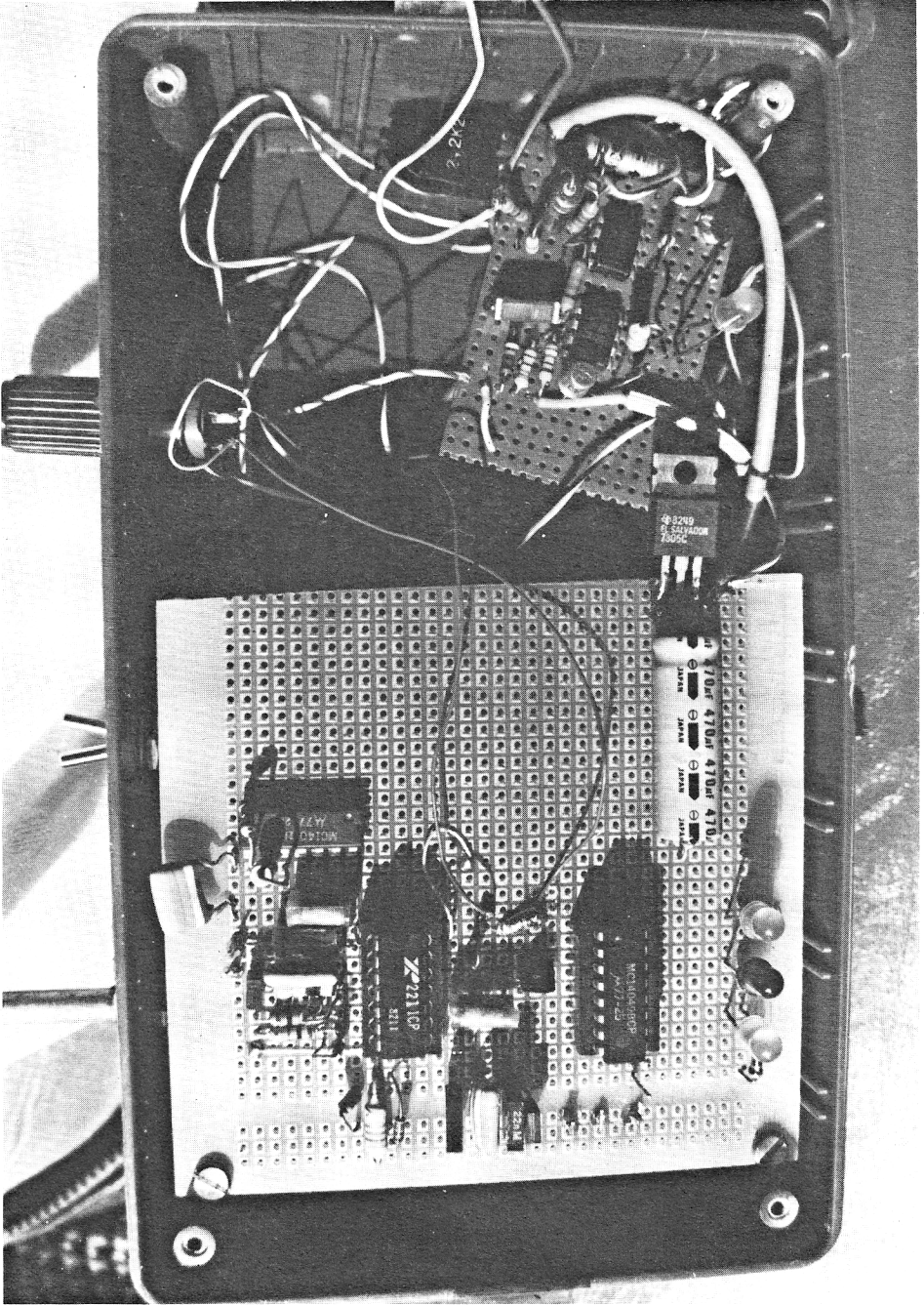


Figure 9: Sur ROMÉ...

Effet de loupe intéressant obtenu en décodant en émission
120 tours en position 240 tours/mn. Ça peut servir !



LES INTERFACES

Cet ouvrage serait incomplet si nous ne vous propositions les schémas des cartes interfaces entre récepteur et ordinateur.

Les 'décodeurs' ou 'démodulateurs', bien que ces termes soient impropres, sont destinés à transformer les signaux analogiques, issus de la sortie BF du récepteur, en signaux susceptibles d'être exploités par l'ordinateur, c'est à dire 'logiques'.

Tout montage capable de réaliser cette transformation pourrait, en théorie, être utilisé.

En CW, le montage fonctionne en décodeur de tonalité. Lorsque le signal reçu entre dans la bande de fréquences du NE 567, le signal en sortie bascule au rythme de la manipulation. Ce signal est disponible sur la sortie d'un transistor. Une diode électroluminescente s'allume sur les points et les traits.

La mise au point est pratiquement inexistante. A l'utilisation, régler la commande de fréquence du récepteur pour un clignotement franc de la diode, la 'bande passante' étant déterminée par P2.

En RTTY, c'est un peu la même chose, sauf que là, il faut reconnaître deux fréquences, celle du MARK et celle du SPACE. LE XR 2211 est un circuit intégrant une PLL (boucle à verrouillage de phase) capable d'accomplir ce travail. La largeur de la boucle de capture sera déterminée par des éléments commutables, afin de pouvoir recevoir des émissions à shift différents.

Cette solution du 'démodulateur à PLL' n'est certainement pas la meilleure au point de vue de la fiabilité en réception. Lorsque le brouillage est important, la PLL a tendance à se verrouiller sur n'importe quoi. Il est préférable d'utiliser, dans ce cas, un montage avec d'excellents filtres actifs. Néanmoins, dans le souci de fournir des montages simples à réaliser, nous préconisons le XR 2211, quitte à changer par la suite...

Un ensemble de diodes électroluminescentes (LED) indique les 2 fréquences (MARK & SPACE) et la perte de porteuse (Boucle hors verrouillage). Un CD 4049, circuit intégrant 6 inverseurs, est utilisé pour allumer les LED et pour l'inversion de shift.

La mise au point sera facilitée si les composants choisis au départ sont de bonne qualité (condensateurs et résistances du PLL). Bien que cela ne soit pas indispensable, l'aide d'un oscilloscope est souhaitable. Il est possible de régler, avec un peu d'habitude, l'interface sur la réception directe de signaux RTTY. L'alternative consiste à se servir de l'AMSTRAD comme générateur de fréquences MARK et SPACE et de coupler sa sortie BF à l'entrée du montage. La partie émission du programme, si vous en êtes sûr, peut aussi être utilisée. Les diodes D1 et D3 doivent clignoter franchement et D2 doit rester éteinte. Le centrage se fera avec le potentiomètre d'accord fin du VCO.

L'ENTREE SUR L'AMSTRAD se fera sur la prise imprimante. Reconnaissons tout de suite le défaut de ce choix: l'imprimante ne peut être utilisée facilement pendant le décodage... Il faut la débrancher, pour mettre en place le connecteur venant de l'interface ou encore, prévoir sur la ligne 'BUSY' du câble imprimante un inverseur à 2 positions: l'une pour le décodage, l'autre pour imprimer. Ce dispositif, peu technique, a le mérite d'être facile à mettre en œuvre, l'arrivée du signal en provenance du 'décodeur' se faisant sur les points 11, c'est-à-dire 'BUSY', pour le point chaud, et 14 pour la masse de la prise imprimante de l'AMSTRAD. Reportez-vous au manuel de votre AMSTRAD pour repérer ces points sur le connecteur imprimante.

Il y a mieux... mais c'est plus cher ! L'utilisation d'une carte d'entrée (ou d'entrée-sortie) donne un aspect plus technique à la liaison récepteur / ordinateur. Précisons tout de suite que les logiciels CW et RTTY n'ont pas été prévus pour cela, dans le but de proposer des interfacages aussi simples que possible, mais qu'ils peuvent être facilement modifiés.

Un minimum (tout petit...) de connaissances en assembleur permet de se tirer d'affaire et de lire, non plus l'entrée de la ligne 'BUSY' de l'imprimante, mais celle de votre carte d'entrée-sortie favorite. Dans ce cas, il n'est plus nécessaire de brancher et débrancher l'imprimante...

Le lecteur pourra également utiliser la carte '8 entrées' que nous proposons (et qu'il devra réaliser pour utiliser les logiciels SSTV et FAX). Cette carte est à l'adresse &FBFF. C'est cette adresse qui sera lue au lieu de la ligne 'BUSY'. En assembleur, cela donne:

```
LD BC, &FBFF
IN A, (C)
```

Après ces 2 instructions, les 8 bits de l'entrée sont dans l'accu.

Ensuite, il suffit de masquer par un AND ou d'effectuer une rotation pour récupérer le bit désiré...

La sortie logique du 'démodulateur' CW ou RTTY pourra être reliée à l'un des bits restés libres (il y en a 4) de la carte '8 entrées'.

Pour la SSTV et le FAX, on utilise donc une carte d'entrée sur laquelle est couplé un convertisseur 'analogique-digital'.

La carte 8 entrées est on ne peut plus simple. Il est à noter que l'on aurait pu faire mieux et proposer une carte 8 entrées-sorties organisée, par exemple, autour d'un PIO (Coupleur parallèle). Là encore, la simplicité et le faible coût de revient ont été recherchés.

Le décodage d'adresse de cette carte est effectué par un couple bien connu 74LS30 / 74LS32. Ceci permet d'avoir une carte qui ne répond qu'à une seule adresse mémoire, ici &FBFF. Le signal produit valide un 'latch' 74LS244 qui place nos entrées logiques sur le bus de données du micro. On dispose donc de 8 lignes d'entrée que l'on peut lire facilement, par BASIC (voir le programme de réglage de la carte 'SSTV') ou en assembleur (voir ci-dessus).

Ces 8 entrées recevront les signaux issus du convertisseur analogique digital. Il est facile d'imaginer la modification de programme permettant au lecteur possédant déjà une carte 8 entrées-sorties, de se passer de la réalisation de notre carte '8 entrées'. ATTENTION, cette modification sera plus difficile si votre carte est organisée autour d'un PIO puisqu'il faudra d'abord mettre ce dernier en configuration d'entrée.

La conversion analogique/digitale du signal SSTV (ou FAX) passe par 3 étapes:

1- Amplification et écretage du signal, permettant de s'affranchir des possibles variations de niveau dues au fading. C'est le rôle du LM 1458, monté en amplificateur, et des 2 diodes zéner écretant le signal à + ou - 3,6V.

2- Démodulation via une PLL dont la boucle de capture est ajustée finement au moyen d'un potentiomètre multitours, pour autoriser le passage des fréquences limites de synchro et du noir.

3- Amplification et alignement du signal par l'intermédiaire d'un transistor. Sur son collecteur, on dispose d'un beau signal 'vidéo'.

4- Un ensemble de comparateurs chargés d'effectuer la ségrégation noir, gris, blanc, synchro.

Si les signaux représentatifs des différentes teintes n'ont pas été filtrés, il n'en est pas de même pour la synchro...

Notons que nous avons limité à 4 teintes... On aurait pu monter 8 comparateurs et travailler en mode d'écran 0... Nous vous soufflons l'idée, si vous voulez l'exploiter !

LES REGLAGES

Les réglages de cette carte sont un peu délicats. Il faut bien ajuster le niveau d'entrée, pour qu'il reste compatible avec les tolérances du NE 565, et régler la boucle de capture de ce dernier afin que tout le signal (et rien que lui) puisse passer. L'oscilloscope serait bien utile mais, sans lui, une bonne dose de patience fera l'affaire. Il est préférable d'utiliser des potentiomètres 'multitours'...

PROCEDURE DE REGLAGE DES COMPARATEURS

Le réglage des comparateurs sera effectué avec l'assistance de l'AMSTRAD. Il suffit de saisir le petit programme de réglage (décrit en prime) et de brancher l'entrée de la carte à la sortie BF de l'ordinateur.

Après avoir lancé le programme, un octet apparaît, en binaire, à l'écran. Une double tonalité (1200 / 1500 Hz) est émise. Il faut alors régler P1 afin que le bit 0 (le plus à droite) oscille entre 0 et 1 au rythme de la tonalité BF. Dès que ce réglage est effectué, on peut passer à l'étape suivante. Un nouvel appui sur une touche provoque le changement des tonalités émises par l'AMSTRAD. Il faut alors régler P2 pour que ce soit le bit 1 qui 'clignote' entre 0 et 1. Répéter l'opération jusqu'au bit 3 (correspondant au réglage de P4).

Les 4 bits de poids fort (les plus à gauche), ne sont pas utilisés et restent à 1.

En résumé:

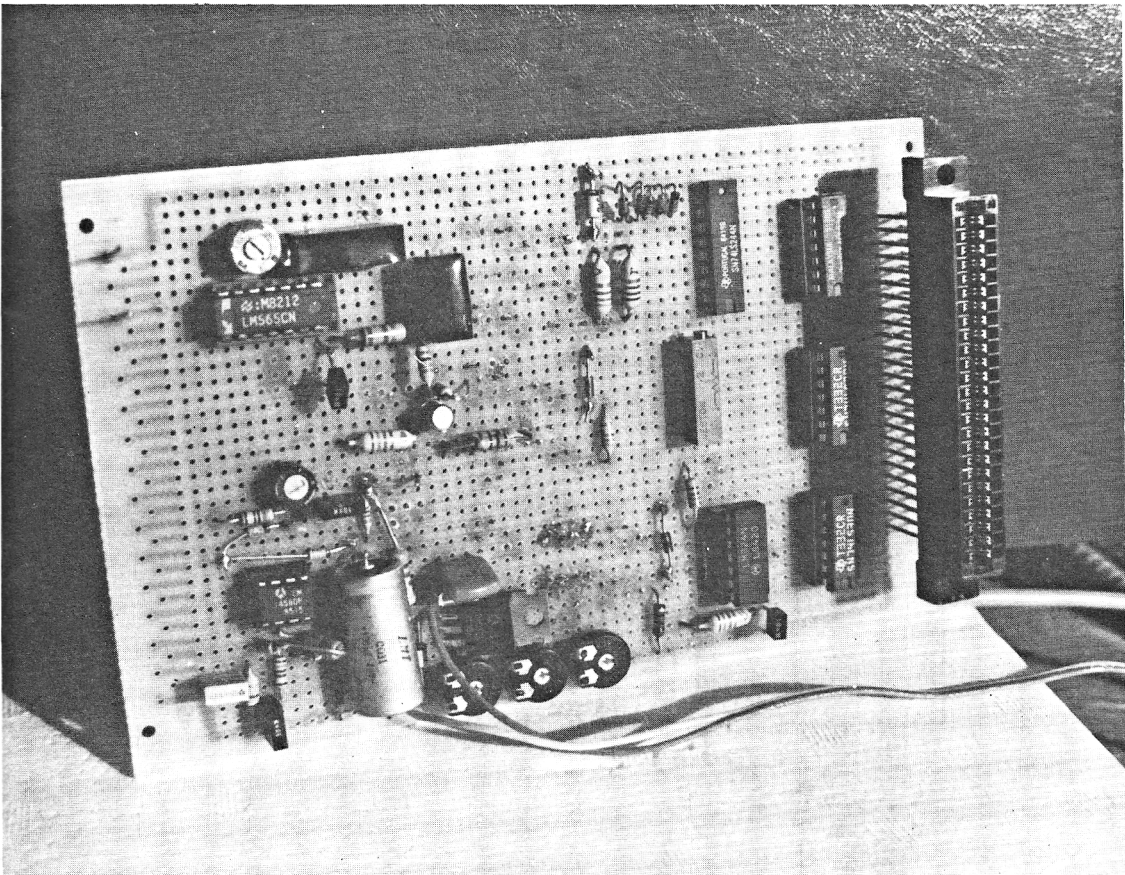
P1 règle la synchro	(SY)
P2 règle le gris foncé	(GF)
P3 règle le gris clair	(GC)
P4 règle le blanc	(BL)

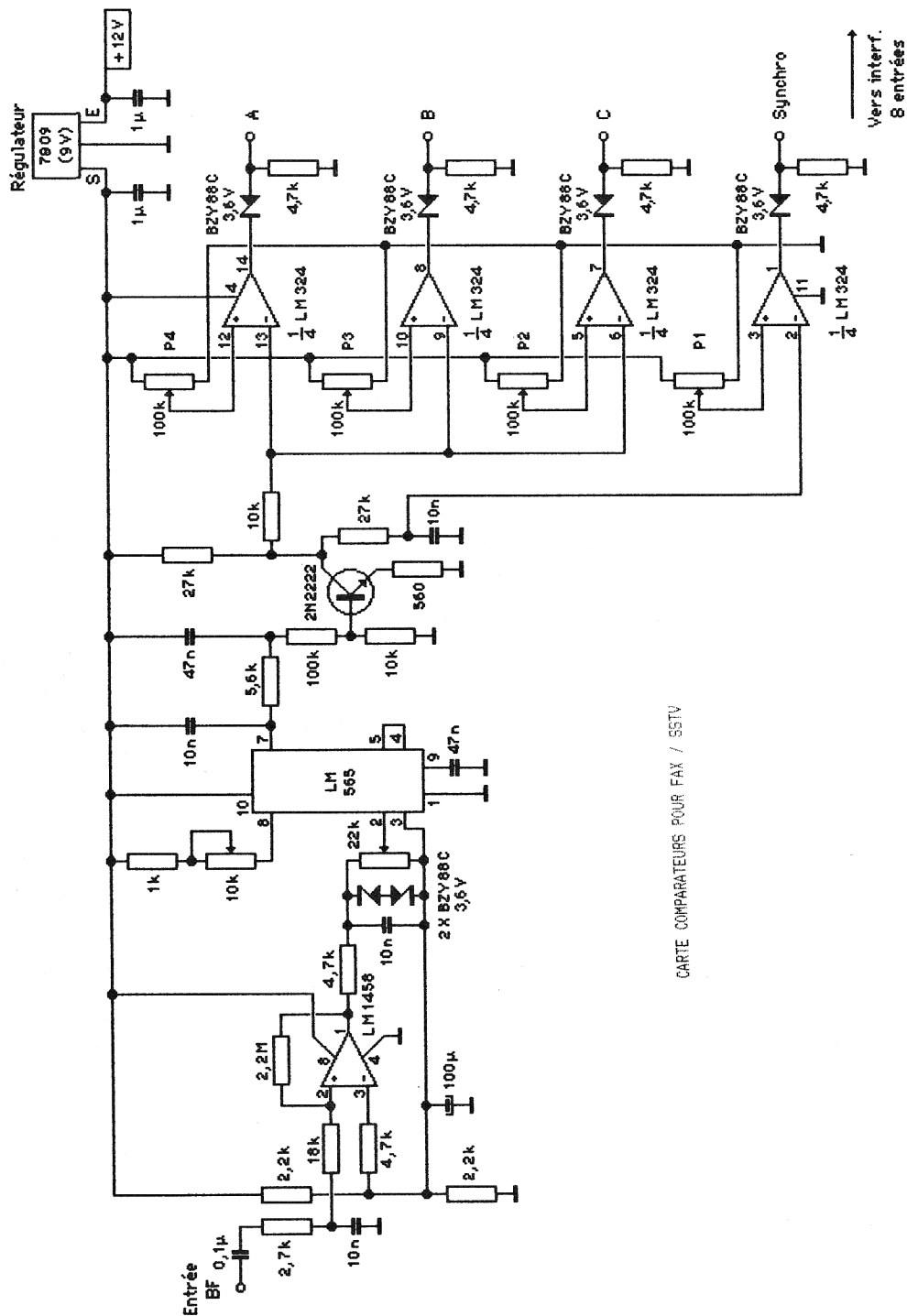
Octet: 1 1 1 1 BL GC GF SY


```

10 'REGLAGE CARTE COMPAREURS
20 '
30 '
40 DATA 41,51,37,39,34,36,32,34
50 CLS
60 FOR n=1 TO 4
70 READ a,b
80 c=a
90 SOUND 1,c,15,15
100 LOCATE 1,1
110 PRINT BIN$(INP(&FBFF));PRINT:PRINT:P
RINT"P";n
120 IF c=a THEN c=b ELSE c=a
130 IF INKEY$("<>") THEN 150
140 GOTO 90
150 NEXT n

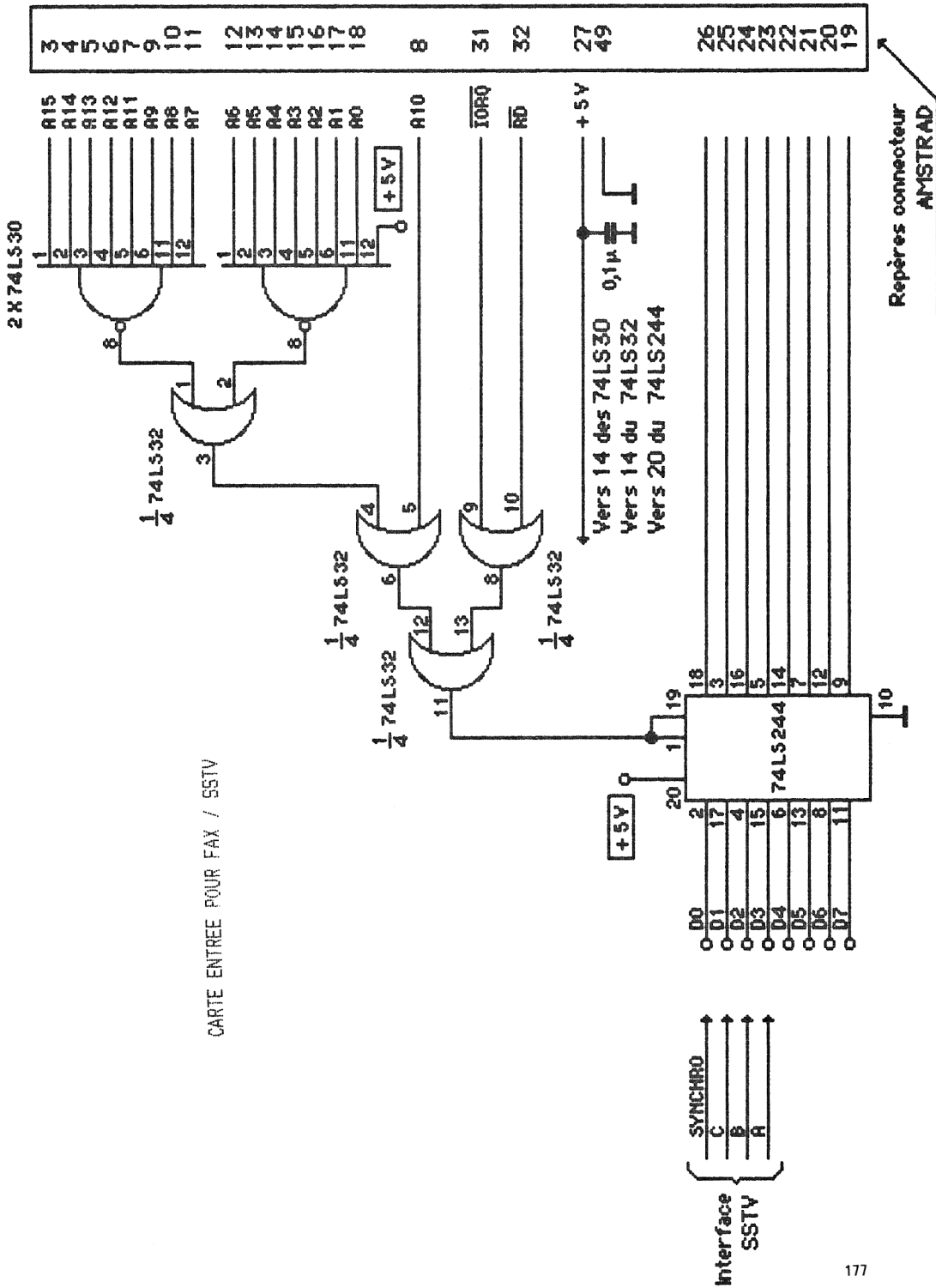
```





CARTE COMPAREURS POUR FAX / SSTV

CARTE ENTREE POUR FAX / SSTV



Valeurs des composants pour les différents 'SHIFT' :

170 Hz	C1	39n	R1	150k
425 Hz	C2	33n	R2	63k
850 Hz	C3	27n	R3	36k

Prendre des composants aux tolérances les plus strictes.

Les inverseurs 1 à 6 sont ceux d'un CD 4049.

L'alimentation du CD 4049 se fait ainsi : +5V en 1, 0V en 8

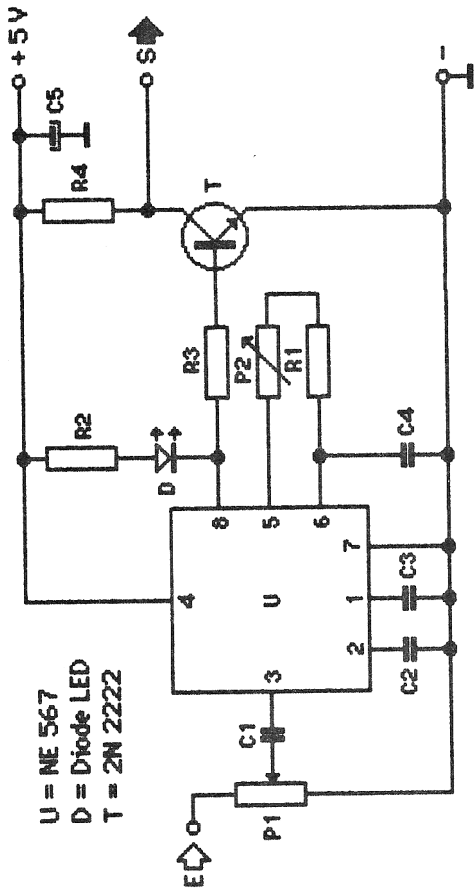
Le commutateur de shift et les composants C2, C3, R2, R3, peuvent être omis si l'on veut recevoir le 170 Hz seulement.

Le XR 2211 est sensible à une entrée de 2mV.

D1 et D3 clignotent au rythme du MARK et du SPACE.

D2 doit s'éteindre lorsqu'on est bien calé.

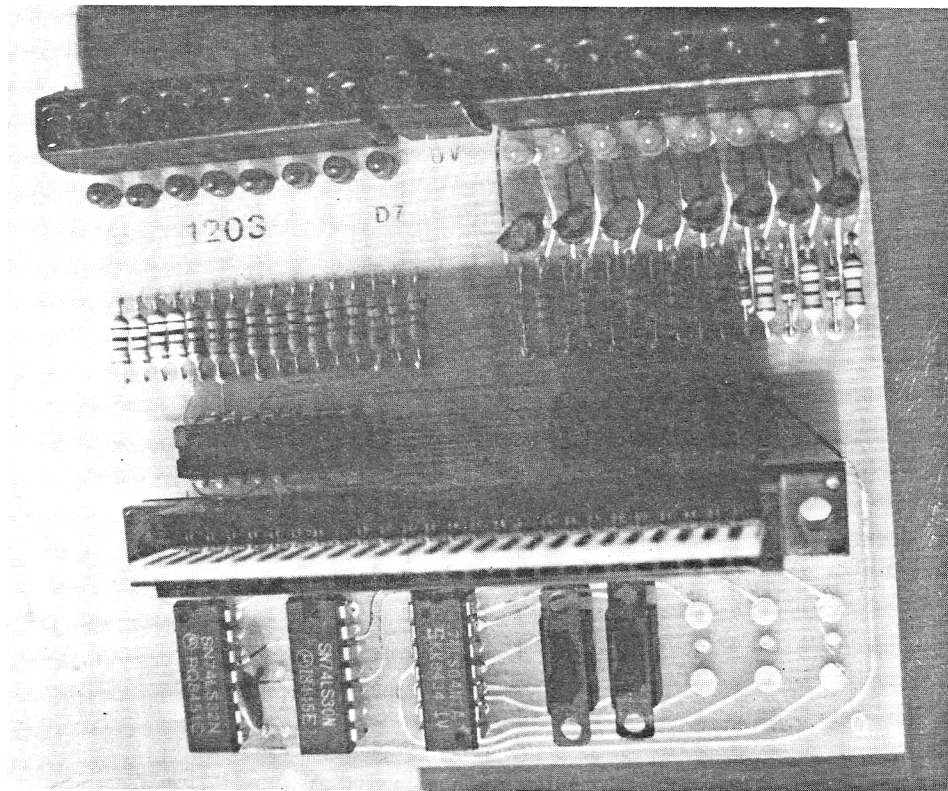
Le seul réglage consiste à ajuster la fréquence centrale du VCO.



U = NE 567
 D = Diode LED
 T = 2N 2222

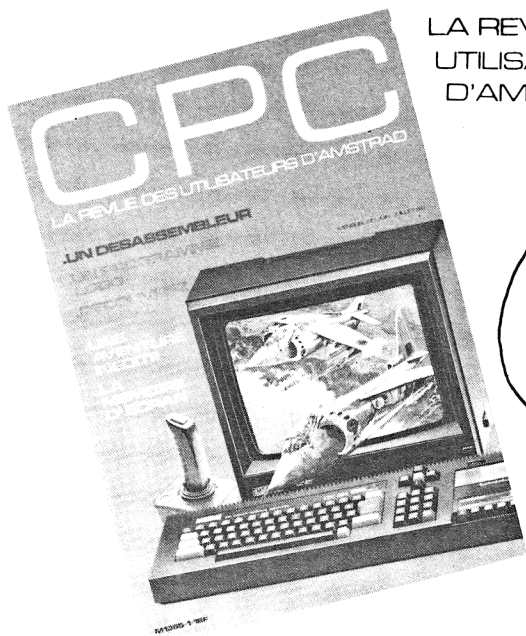
- R1 = 8,2 k
- R2 = 1 k
- R3 = 10 k
- R4 = 10 k
- P1 = Pot. 1 k
- P2 = Pot. 10 k multivoirs
- C1 = 0,22 μ
- C2 = 0,1 μ
- C3 = 1 μ
- C4 = 0,1 μ
- C5 = 47 μ 10 V

DEMULATEUR CW



LES VRAIS LIONS PROGRAMMENT A V E C C P C

LA REVUE DES
UTILISATEURS
D'AMSTRAD



**CHAQUE MOIS,
N'OUBLIEZ PAS!**



SORACOM
éditions

10, Av. du Général de Gaulle
BP 46
35170 BRUZ

INTERFACES MAGECO

Nous avons pensé à tous nos lecteurs qui ne construisent pas. Voulant que ce livre soit aussi complet que possible, nous avons décidé de décrire les possibilités offertes par deux cartes commercialisées. Il s'agit de l'interface 8 entrées-sorties et de la carte 8 entrées analogiques de MAGECO. Ces cartes sont disponibles en particulier chez 'ORDIVIDUEL' à Paris. Leur bon rapport qualité/prix nous à paru être un argument favorable, c'est pourquoi nous leur avons attribué une application...

LA CARTE 8 ENTREES-SORTIES

Elle est organisée autour d'un 74LS244 (8 buffers), pour la partie 'entrées', et d'un 74LS273 (8 bascules D), pour la partie 'sorties'. Par conséquent, elle sera facile à programmer. Le décodage d'adresse permet de sélectionner 4 emplacements différents, dans l'espace des entrées-sorties, ce qui garantit une certaine compatibilité avec d'autres cartes. Du très classique à ce niveau: 74LS30, 74LS32, 74LS04. La sélection de l'adresse se fait par un couple d'inverseurs agissant sur les lignes d'adresses A3 et A4.

Munie d'un connecteur, la carte vient se monter directement sur le bus de sortie de l'AMSTRAD. Ce connecteur est situé coté composants. De l'autre coté de la carte, on trouve un connecteur 'male' permettant la connexion d'une autre interface (ou du DRIVE pour le CPC 464). Une remarque, sur le 664 et le 6128, le montage de la carte sur le connecteur d'extension impose de surélever l'AMSTRAD. Ce petit défaut de conception disparaîtra si vous utilisez une courte nappe rallonge de bus.

Physiquement, les connexions de la carte vers le monde extérieur se font par l'intermédiaire d'un bornier à vis, qui se révèle bien pratique lors des expérimentations.

L'état des entrées est visualisé sur des diodes électroluminescentes rouges, celui des sorties par des diodes vertes. Une entrée en l'air correspond à l'état 1 et, la diode qui lui est associée étant alimentée à partir du + 5V, se trouve éteinte. La présence d'un 0 (masse) sur l'entrée allumera la diode. Pour chaque ligne, l'entrée vers le 74LS244 s'effectue à travers une résistance de 1 K.

Les diodes associées aux sorties s'allument pour un état 1 (haut) de la ligne correspondante. La sortie s'effectue sur le collecteur, ouvert, d'un transistor, l'espace émetteur-collecteur du transistor étant shunté par une diode. Au transistor, on pourra relier une charge qui sera commandée par la saturation de celui-ci. Cette charge devra assurer un courant maximal de 1A dans le transistor. Une alimentation externe, pouvant atteindre au maximum 20V, fournira la tension au circuit extérieur. Si la charge est selfique, cas d'un relais de commutation, il est indispensable de la shunter par une diode (genre 1N4148, par exemple).

Sa programmation est très simple: il suffit d'envoyer le profil binaire (ou traduit en hexa. ou décimal) de l'octet à présenter sur les lignes de sortie. Cette opération est réalisée par un OUT en BASIC ou par la séquence suivante, en assembleur:

LD BC, 0FBEOH	adresse carte
LD A,xx	xx octet à envoyer
OUT (C),A	opération de sortie
RET	retour

En BASIC, on écrira OUT &EO, xx (ou OUT &FBEO, xx)

Si on souhaite mettre les bits 2 et 7 à 1 (et seulement eux) on fera:

```
OUT &EO, 132 (= 10000100)
```

En entrée, pas de difficulté: un INP en BASIC ou la séquence assembleur suivante:

LD BC, 0FBEOH	adresse carte
IN A,(C)	lecture du port
RET	retour

Vous voulez recopier sur les sorties l'état des entrées ? Rien de plus simple...OUT &EO, INP(&EO) en BASIC et, comme vous avez compris la manoeuvre, vous saurez quoi écrire en ASSEMBLEUR. On ne va pas tout vous dire !

En cablant la sortie du démodulateur sur une des lignes d'entrée de la carte, on pourra libérer le port imprimante, utilisé à l'origine par le programme de réception. La partie réception du logiciel devra être modifiée.

Au chapitre 'Utilisation d'une carte d'entrée', nous expliquons comment modifier le sous-programme de réception (et les modifications correspondantes du BASIC). Nous les rappelons ici, dans le cas de l'utilisation du bit 7 de la carte MAGECO.

LB BC, OFREDH	Adresse carte MAGECO
IN A, (C)	Récupère l'octet en entrée.
NOP	
RLA	Met le bit 7 dans la Carry
RET	Revient.

Seule la ligne 1720 du BASIC sera modifiée:

```
... 01,EO,FB,ED,78,00
```

Notons qu'il est possible de cabler une des lignes du port de sortie sur le relais émission-réception, ce qui permet de télécommander par le programme, le fonctionnement de l'émetteur. Dans ce cas, le passage en émission et en réception sera commandé par une ligne de BASIC, placée avant l'appel des routines correspondantes.

Bien des applications sont envisageables avec cette carte 8 E/S. Citons en vrac: la commande du relais du rotor d'antennes, la commande de Scanning du récepteur (si celui-ci le permet), la mise en veille automatique de la station à une heure donnée etc.

LA CARTE 8 ENTREES ANALOGIQUES

Pourquoi une carte analogique ? Pour permettre à l'ordinateur de saisir des données telles que tension, température, niveau d'éclairement, vitesse du vent etc. Toutes ces données, à caractère analogique, seront converties en informations digitales par un convertisseur 'analogique-digital'.

L'âme de la carte est justement ce composant: le convertisseur analogique-digital ADC 809. C'est un circuit intégré à 28 broches effectuant la conversion par le procédé des 'approximations successives'. Avec ce procédé, le calculateur fait varier une tension de référence et compare la grandeur à

mesurer à cette tension. Lorsque le résultat de la comparaison est 'vrai' (bon), la grandeur, codée sur 8 bits, est disponible en parallèle sur les sorties de l'ADC 809.

Le temps de conversion détermine les performances et les limitations du circuit. Si l'on voulait convertir une image vidéo, il faudrait un composant extrêmement rapide (dont le prix se chiffre par centaines de francs). Dans notre cas, l'ADC 809 permet toutefois des mesures honorables.

La carte est réalisée en époxy double face, à trous métallisés. Les 8 entrées se font par un bornier. Le 0v et la tension de référence (ici 5V) sont également disponibles sur ce bornier. Un connecteur relais permet d'utiliser la carte en liaison avec une autre, par exemple la carte 8 E/S...

Côté électronique, on distingue, outre le convertisseur, les circuits du décodage d'adresse. Ce décodage place la carte à un emplacement programmable parmi 4, grâce au jeu de 2 interrupteurs.

La carte est programmable en BASIC. Dans ce cas, il sera impossible de bénéficier de toutes les performances de l'ADC 809, à cause de la lenteur du langage. Une programmation en langage machine résout le problème... Le temps de conversion entre deux mesures est évalué, dans ce cas, à 100 microsecondes. Il est donc possible d'échantillonner des fréquences BF (traitement de la voix, analyse de signaux etc.).

Programmer la carte est un jeu d'enfant. On dispose de 8 entrées: il suffit de désigner celle à laquelle s'adresse la mesure, par son numéro physique. La lecture du résultat est tout aussi simple... Voici un court exemple, par lequel on lit l'entrée numéro 3.

```
OUT (&FBFB),3: A=INPUT (&FBFB): PRINT A
```

Nous n'avons pas placé de boucle de temporisation entre la demande de mesure (OUT) et la lecture du résultat (INP), le BASIC étant suffisamment lent!

Le résultat obtenu est un nombre codé sur un octet, donc de 0 à 255. Pour établir la relation qui existe entre ce nombre et la tension réellement présente sur l'entrée considérée, on tient compte de la valeur de la tension de référence (ici 5V). Si la tension mesurée est à 5V, on obtient 0. Si elle est à 0V, on lira 255.

La relation est donc $Tension = Mesure * V_{Ref}/255$.

Ici on peut écrire $U=M*5/255$ ou U est la tension réelle et M la mesure.

ATTENTION, toutes les mesures doivent être ramenées dans la gamme 0 à 5V, sous peine de détruire le convertisseur. En alternatif, car il est possible de mesurer des tensions alternatives, la tension maximale autorisée sera de 5V crête à crête.

En langage machine, il faudra incorporer, entre la demande de mesure et la lecture du résultat, un délai d'environ 100 microsecondes... pour que la mesure ait le temps de s'effectuer. Cela donnera:

LD BC, 0FBFBH	Adresse de la carte
LD A, 3	Sélection entrée No 3
OUT (C), A	Demande de mesure
JSR TEMPO	Sous-programme d'attente
IN A, (C)	Lecture du résultat
RET	Retour

La précision du résultat sera de + ou - 20 mV. En effet, $5V/255 \approx 20mV$.

On pourra, lorsque la rapidité n'est pas un impératif, effectuer des mesures répétitives, dans le but de diminuer la dispersion et d'améliorer la précision. On prendra pour résultat, la moyenne établie sur les N mesures. Ceci pourra s'écrire, toujours pour l'entrée No 3:

10 M=0	Initialise la moyenne
20 FOR I = 1 TO 20	Nombre de mesures
30 OUT (&FBFB),3	Demande de mesure
40 A=INP (&FBFB)	Lecture du résultat
50 M=M+A	Totalise
60 NEXT	Mesure suivante
70 PRINT M=M/20	Imprime valeur moyenne

Rappelons que, pour gagner du temps en BASIC, il serait préférable de mettre toutes les instructions sur une même ligne...

Il serait difficile de citer toutes les applications possibles d'une carte analogique comme celle-ci: régulation de température (capteur thermique), surveillance d'un local (capteur optique), études diverses (courbes de décharge d'un condensateur chimique, d'une batterie d'accumulateur...).

Les deux cartes, utilisées ensemble, pourront commander la station radio et ce, en absence de l'opérateur. Nous suggérons un veilleur de bande automatique et un système d'orientation des antennes. Dans les deux cas, il est nécessaire de posséder un type de matériel particulier ou alors, de faire des adaptations sur le matériel existant, pour le rendre utilisable.

La veille automatique.

Si vous disposez d'un récepteur disposant d'une commande de scanning, il vous sera possible de surveiller une bande de fréquences en votre absence. L'un des auteurs de ce livre, possédant un MULTI 750E (E/R VHF de FDK), a utilisé l'ordinateur et ses deux cartes pour surveiller les variations de propagation sur 144 MHz, tout au long d'une journée, pendant ses heures de travail. ATTENTION, ames sensibles s'abstenir ! Il est parfois très rageant de constater, en rentrant à 19 heures, que l'on a raté une superbe ouverture !

A ce stade, il existe deux solutions: une simple, ne faisant pas appel à la carte B E/S: on 'écoute' simplement une seule fréquence (balise ou répéteur lointain). La sortie du 'S-METRE' est reliée à une entrée de la carte analogique.

L'autre est plus complexe et fait appel aux deux cartes. Le principe retenu est le suivant. La commande de scanning du récepteur est reliée à la carte B E/S. Cette liaison peut s'effectuer directement en parallèle sur les poussoirs 'UP' & 'DOWN'. La sortie du galvanomètre 'S-METRE' sera reliée à une entrée de la carte analogique. Dans certains cas, il faudra peut-être prévoir un amplificateur de tension, la carte n'étant sensible qu'à 20 mV.

Le récepteur sera placé sur une fréquence, marquant le début de la bande à écouter, le programme se chargera de faire varier cette fréquence. La suite, vous la devinez... On envoie une impulsion sur la commande 'UP', par l'intermédiaire de la carte B E/S: la fréquence est modifiée (de 100 Hz ou 5 KHz, tout dépend du 'pas' de balayage de votre appareil et du mode, BLU ou FM, que vous avez sélectionné). Il suffit alors de mesurer la tension en sortie du 'S-METRE' de manière répétitive, pendant une minute par exemple, de faire la moyenne des mesures, et de mémoriser le résultat dans un tableau. On attend alors quelques minutes, et on recommence l'opération sur une autre fréquence, après avoir fait avancer (par la commande de scanning et la carte B E/S) le récepteur d'un pas. Une fois arrivé au bout de la bande (ou du segment de bande à surveiller), on commande le scanning en sens inverse...

Pour dépouiller les résultats, il suffit d'un programme bien fait. On connaît l'heure de début d'observation. On a déterminé le nombre d'incréments en

fonction de la longueur de la période d'observation. On a programmé les variations d'incrément en fonction de la largeur de bande à surveiller. Il suffit d'avoir rangé tous les résultats dans un tableau. Au besoin, on peut sauvegarder, de temps en temps, ce tableau sur disquette ou cassette, et reprendre les observations... Il est même possible de sortir tous ces résultats sur imprimante, sous forme graphique... Quoi de plus simple que cette veille automatique ?

En extrapolant le principe précédent, il est permis de concevoir un programme capable de visualiser à l'écran, sous forme graphique, l'activité d'une bande. On obtient alors, à bon prix, un récepteur panoramique. En jetant un coup d'oeil sur l'écran de temps en temps, entre deux soudures, on aura vite un aperçu de l'activité qui règne sur la bande... Le seul inconvénient dans tout cela est que, il faut avoir déterminé auparavant, quelles sont les fréquences parasites générées par l'AMSTRAD, pour ne pas les confondre avec des émissions. Le logiciel sophistiqué, et bien conçu, n'affichera pas les raies correspondantes...

Orientation des antennes

Notre propos n'est pas de décrire une interface d'asservissement de la position des aériens, mais une astuce simple, tenant compte des possibilités offertes par certains rotors d'antennes, munis d'un galvanomètre indicateur de position. C'est le cas, par exemple, des KR 600, 800 etc.

La carte B E/S aura deux sorties reliées, à travers des relais s'il y a lieu (c'est plus prudent), aux commandes de sens de rotation du moteur. La carte analogique sera utilisée pour lire la tension du galvanomètre. Le reste est affaire de logiciel et... d'astuce (ce n'est pas très technique, répétons-le !).

Le programme devra intégrer une table de correspondance 'Tension-Azimut'. On peut envisager d'aller plus loin: les indicatifs des stations les plus souvent contactées (ou écoutées), seront également mémorisés. Il suffira alors d'entrer une direction au clavier (ou un indicatif), pour que l'AMSTRAD positionne, avec une précision relativement bonne, l'aérien dans la direction. On peut même envisager de coupler ce système au programme au programme LOCATOR... On donne une direction: elle est convertie par la table d'équivalence, en tension. Le moteur est commandé dans le bon sens jusqu'à ce que le galvanomètre indique au programme, par l'intermédiaire de la carte analogique, la tension voulue.

Toujours dans le même ordre d'idées, on peut coupler le récepteur... à l'antenne (c'est déjà fait !) mais par l'intermédiaire du S-METRE et du ROTOR. Bien sûr, dans la chaîne, on retrouvera les cartes analogique et B E/S. Il sera alors facile de suivre une station mobile en orientant toujours l'antenne dans le sens qui donnera le plus fort signal... sur le S-METRE.

Enfin, tous ceux qui craignent les effets néfastes du vent sur leurs aériens à grand gain (groupement d'antennes ou beam décimétrique) pourront associer les deux cartes, l'AMSTRAD, et une mini-station météo, capable de mesurer la force et la direction du vent. On mesure la vitesse. Si elle dépasse pendant un certain temps, choisi à l'avance (il ne faut pas transformer l'antenne en girouette !), pour s'affranchir des courtes rafales, la vitesse limite supportée par l'aérien, on commande le moteur pour placer les antennes dans le lit du vent. L'anémomètre sera couplé à la carte analogique, la girouette à la carte 8 E/S (ou à la carte analogique, selon sa conception), les commandes de moteur et lecture de position comme indiqué ci-dessus.

Par ces exemples, ou nous n'apportons pas de solution typique, en raison des différences trop importantes de matériel radio, nous avons voulu démontrer que, les cartes analogiques et d'entrée-sortie, peuvent occuper une place importante à la station, aux cotés de l'AMSTRAD. Le matériel MAGECO, que nous vous avons conseillé sans aucun souci publicitaire, a été choisi en raison de son excellent rapport qualité/prix qui devrait ravir tous ceux qui ne veulent pas se lancer dans des réalisations 'Hard'. Associées au(x) logiciel(s) approprié(s), elles transforment l'AMSTRAD en opérateur supplémentaire...

CONCLUSION

Nous espérons avoir montré au lecteur avide de communication, que l'AMSTRAD se prête très bien à ce genre d'applications. La liste de logiciels proposés ici n'était pas exhaustive; bon nombre d'entre-vous en découvriront d'autres. Le langage machine n'est pas un point de passage obligatoire grâce à la rapidité du BASIC de l'AMSTRAD, mais reste nécessaire pour accomplir certaines tâches.

D'autres domaines pourraient être explorés. L'utilisation de la synthèse vocale remplacera l'opérateur pour lancer appel. Ceci pourra se faire au moyen d'un synthétiseur vocal bâti autour du MEA 8000, plus performant pour nous français que le SPO 256 des anglos-saxons.

Le crayon optique, au moyen d'un logiciel approprié, sera utilisable en télévision d'amateur, développant ainsi une nouvelle forme de dialogue !

Ne pas oublier l'extrême puissance de calcul de l'ordinateur, qui deviendra un précieux auxiliaire pour tous ceux qui créent leur matériel. De même, un logiciel de poursuite de satellites, couplé à une carte d'entrée-sortie permettra aux passionnés de se consacrer au trafic avec un maximum d'efficacité.

L'avenir est aux transmissions numérisées, et il appartient aux amateurs de communication d'explorer au maximum ce domaine... au moyen de l'AMSTRAD, bien entendu !

Pour nos lecteurs qui découvriraient le monde fabuleux des ondes courtes, à la lecture de cet ouvrage, nous décrivons ci-après quelques uns des procédés utilisés en émission, 'amateur' ou 'générale'.

Nous nous permettons également de vous suggérer quelques critères de choix pour un récepteur.

LES PROCEDES DE TRANSMISSION

LA TELEGRAPHIE MORSE (CW)

C'est, très certainement, le mode de transmission des signaux le plus fiable qui soit, car il permet, avec des moyens relativement simples, de transmettre l'information avec un maximum d'efficacité. L'émetteur est très simplifié, par rapport aux émetteurs 'phonie' (surtout les BLU), transmettant la parole et, à la réception, les effets des différents brouillages ont peu d'incidence.

L'émission s'effectue, en général, par découpage de l'onde porteuse, ce qui garantit un maximum de puissance avec un minimum de moyens. A la réception, le matériel pourra être simplifié à l'extrême mais nécessitera la présence d'un opérateur spécialement entraîné... ou d'un AMSTRAD pourvu des logiciels de ce livre...

Le code inventé par Samuel MORSE est le suivant:

A	.-	B	-...
C	-.-	D	-..
E	.	F	..-
G	--.	H
I	..	J	.-.-
K	-.-	L	.-..
M	--	N	-.
O	---	P	.-.-
Q	-.--	R	.-.
S	...	T	-
U	..-	V	...-
W	.-.-	X	-.-.-
Y	-.--	Z	--..
0	-----	1	.-----
2	..----	3	...---
4-	5
6	-....	7	--...
8	---..	9	----,
=	-...-	?	..-...-
.	.-.-.-	,	-.-.-
/	-.-.-	ERREUR

En suivant des cours de lecture au son, ou en utilisant le logiciel DIDMORSE, décrit dans ce livre, le lecteur pourra apprendre à décoder les émissions en télégraphie MORSE.

LE RADIOTELETYPE (RTTY)

Ce mode de transmission permet également d'obtenir une grande efficacité tout en offrant, à l'utilisateur la possibilité de conserver, sur papier, une trace des messages recus.

Techniquement parlant, il est possible de dire que le RTTY est une transmission sérielle sur 7 bits respectant un code appelé code BAUDOT (du nom de son inventeur).

Dans ce code, chaque caractère est composé d'un bit de départ (START), de 5 bits de données et d'un bit de fin (STOP).

Avec 5 bits de données, il est possible de coder 31 caractères. La vitesse de transmission est variable, couramment de 45.45 Bauds (amateurs) à 110 bauds. La vitesse la plus répandue dans le milieu professionnel est le 50 Bauds ce qui donne une durée du bit égale à 20ms. Ces bits sont appelés 'moments'.

Les deux états, haut ou bas (0 ou 1), on dit MARK ou SPACE, sont codés par deux fréquences dont l'écart est appelé SHIFT.

Si le MARK est à 1275 Hertz, on aura comme valeur de SPACE:

1445 Hz pour un shift de 170 Hz

1700 Hz pour un shift de 425 Hz

2125 Hz pour un shift de 850 Hz

Le démodulateur transformera ces signaux modulés en états logiques 0/1.

Le bit de poids faible du caractère est émis le premier, de suite après le bit de START. Le bit de poids fort précède le bit de STOP. Ce bit de STOP est différent des autres en ce sens qu'il dure plus longtemps: 1,5 moment.

Un caractère spécial permet le passage du mode 'LETTRES' au mode 'CHIFFRES'.

Voici le code BAUDOT utilisé:

LETTRES	CHIFFRES	CODE BAUDOT	VALEUR DECIMALE
A	-	11000	3
B	?	10011	25
C	:	01110	14
D	*	10010	9
E	3	10000	1
F	E	10110	13
G	%	01011	26
H	H	00101	20
I	8	01100	6
J	sonnerie	11010	11
K	(11110	15
L)	01001	18
M	.	00111	28
N	,	00110	12
O	9	00011	24
P	0	01101	22
Q	1	11101	23
R	4	01010	10
S	'	10100	5
T	5	00001	16
U	7	11100	7
V	=	01111	30
W	2	11001	19
X	/	10111	29
Y	6	10101	21
Z	+	10001	17
RETOUR CHARIOT		00010	8
SAUT DE LIGNE		01000	2
MODE LETTRES		11111	31
MODE CHIFFRES		11011	27
ESPACE		00100	4

LE FAC-SIMILE (FAX)

Ce procédé permet d'émettre des documents 'papier', tels que cartes météo, photos etc. A sa source, des machines mécaniques, cylindres tournants sur lesquels sont fixés, d'un coté le document à émettre, de l'autre la feuille pour la réception.

L'analyse de l'image est faite par un élément photo-sensible se déplaçant le long du cylindre pendant la rotation de celui-ci.

Il n'y a pas de synchronisation entre l'émetteur et le récepteur, pendant toute la durée de la transmission. Au début seulement sont transmis des signaux de mise en phase. Une rigoureuse stabilité des deux appareils est donc indispensable.

Le paramètre appelé 'module de coopération' est fonction de la vitesse linéaire de déplacement de l'élément photo-sensible.

Plusieurs vitesses de rotation du cylindre sont en vigueur, la plus répandue étant le 120 tours/minute. On trouve aussi du 180, 90, 60 tours/mn.

La transmission radio de ces signaux s'effectue par modulation en fréquence d'un signal BF. Le noir est codé par une fréquence de 2300 Hz et le blanc à 1500 Hz.

La transmission d'un meme document dure plusieurs minutes: entre 6 et 10 minutes aux vitesses les plus courantes.

L'ordinateur ne permettra pas d'obtenir la meme résolution d'image que celle obtenue avec les machines mécaniques...

Il est préférable de démarrer la réception sur les signaux de mise en phase.

LA TELEVISION A BALAYAGE LENT (SSTV)

Contrairement aux procédés décrits ci-dessus, la SSTV n'est pratiquement utilisée que par les radioamateurs. Grace à elle, il est possible de transmettre des images, issues d'une caméra vidéo, à très longue distance. En effet, contrairement à la télévision conventionnelle, la SSTV n'occupe qu'une bande passante très faible, ce qui permet de transmettre les images par le canal d'un émetteur normal.

Le revers de la médaille est que, les images transmises, ne peuvent etre animées: une image demande 7 secondes.

Pour coder les signaux qui modulent l'émetteur, on utilise une bande de fréquences s'étendant de 1500 Hz, pour le noir, à 2300 Hz pour le blanc. Les signaux sont synchronisés par... une synchro ligne et une synchro trame.

La fréquence de codage des signaux de synchro est de 1200 Hz. La durée de la synchro ligne est de 5ms, celle de la synchro trame de 30ms.

Une ligne image dure environ 66ms.

L'avantage de ce procédé est que les images, simples signaux BF, peuvent être enregistrées sur cassette et revisualisées par la suite: pas besoin de magnétoscope !

Voici donc, résumés en quelques phrases, les différentes caractéristiques des procédés de transmission décrits dans cet ouvrage.

LE CHOIX DU MATERIEL

Les logiciels proposés dans cet ouvrage, ne pourront donner le meilleur de leurs possibilités que s'ils sont exploités en conjonction avec un bon ensemble de réception.

Par ensemble de réception, on entend le récepteur mais aussi l'antenne.

L'ANTENNE

L'antenne est un élément essentiel de la station de réception. Sans une bonne antenne, il est impossible d'obtenir une bonne réception. Celle-ci devra être située le plus haut possible et dégagée, autant que faire se peut, des sources de perturbations: lignes électriques, ventilations mécaniques d'immeubles etc.

Pour la réception seule, une antenne 'long fil' pourra suffire dans la plupart des cas. Il est également possible d'utiliser une antenne directive, qui a l'avantage de présenter un certain gain et d'éliminer les émissions perturbatrices, et l'inconvénient de son encombrement...

Une antenne verticale multibandes fera également l'affaire et donnera de bons résultats.

Par contre, et nous insistons sur ce point, les antennes actives seront à proscrire car leur amplificateurs à large bande récupèrent tous les parasites générés par le micro-ordinateur. Un simple 'bout de fil' à l'intérieur de la pièce présentera les mêmes inconvénients.

Le récepteur est l'élément principal de la station: son choix sera déterminant... Il va sans dire que ce choix sera guidé surtout par les possibilités financières de l'acquéreur.

Le prix des récepteurs modernes va de 3000 à plus de 20000 Francs. Leur conception et les options dont ils disposent justifient les différences.

Les récepteurs orientés 'radiodiffusion' sont à proscrire car leurs performances, suffisantes pour la réception de stations 'broadcast' lointaines, se dégradent rapidement pour la réception de 'trafic'.

La réception des signaux RTTY, FAX, SSTV et même CW, impose la présence de filtres d'excellente qualité (filtres à quartz de préférence) et une stabilité rigoureuse.

La préférence ira aux récepteurs synthétisés équipés d'un affichage de fréquence de précision (affichage digital).

Certains diront que les récepteurs actuels se 'gadgétisent', à cause des nombreuses commandes dont ils disposent. Pourtant, quelques mémoires en réception sont bien agréables à utiliser, surtout si on peut les balayer rapidement pour retrouver (ou surveiller) la fréquence d'une station qui n'émet pas en permanence.

De même, le PBT (Pass Band Tuning) permettra d'éliminer les perturbations, ce qui améliorera la réception dans les modes qui nous intéressent.

Le récepteur sera capable de démoduler AM/CW/BLU, par contre la FM n'est pas utilisée en décimétrie. C'est seulement si vous désirez utiliser le récepteur avec un convertisseur de fréquences (pour écouter les VHF/UHF) que cette option pourra vous servir.

Un atténuateur d'entrée, ou un gain HF réglable, n'est pas un gadget.

Sans soucis de publicité, nous citons ici quelques modèles de récepteurs répandus... parmi les moins de 8000 Francs.

KENWOOD R600 (ou mieux, R1000 hélas disparu)

YAESU FRG 7700

KENWOOD R2000

ICOM ICR 71

YAESU FRG 8800

Il est souhaitable que le récepteur soit suffisamment éloigné de l'ordinateur, surtout si l'antenne n'est pas performante. Avec l'AMSTRAD, les perturbations sont surtout virulentes sur les fréquences supérieures à 14 MHz. Une bonne prise de terre peut parfois arranger bien des choses.

ANNEXE I

QUELQUES FREQUENCES POUR COMMENCER L'ECOUTE

Pour nos lecteurs qui, à l'occasion de la lecture de cet ouvrage, découvriraient le monde passionnant des ondes courtes, nous listons ici quelques fréquences où ils pourront trouver des émissions intéressantes à écouter.

L'écoute des ondes courtes est passionnante mais ATTENTION, VOUS NE DEVEZ EN AUCUN CAS, ET SOUS QUELQUE FORME QUE CE SOIT, REVELER LE CONTENU DES EMISSIONS QUE VOUS AVEZ ENTENDUES.

LES BANDES 'RADIOAMATEUR'

En vertu de conventions internationales, les radioamateurs du monde entier disposent de bandes de fréquences réservées à leur usage.

En ondes courtes, voici la liste de ces fréquences (en MHz):

1.810 à 1.850	(avec des réserves)
3.500 à 3.800	CW jusqu'à 3.600 RTTY + ou - 20 KHz autour de 3.600
7.000 à 7.100	CW jusqu'à 7.040 RTTY + ou - 5 KHz autour de 7.040
10.100 à 10.150	CW essentiellement
14.000 à 14.350	CW jusqu'à 14.100 RTTY + ou - 10 KHz autour de 14.090
18.068 à 18.168	
21.000 à 21.450	CW jusqu'à 21.100 RTTY + ou - 20 KHz autour de 21.100
24.890 à 24.990	
28.000 à 29.700	CW jusqu'à 28.100 RTTY + ou - 50 KHz autour de 28.100

Il existe également d'autres bandes de trafic, en VHF et UHF, notamment les bandes de 144 à 146 MHz et 430 à 440 Mhz.

Sur 144.600 on trouve, en particulier dans les grandes villes, du trafic en RTTY. L'absence de perturbations et la proximité des stations pourront permettre à l'amateur de mettre au point son installation...

QUELQUES AUTRES FREQUENCES

Les ondes courtes comptent beaucoup d'usagers et certaines fréquences pourront aiguïser la curiosité, ou permettre de vérifier le bon fonctionnement des logiciels et interfaces proposés ici. Toutes les fréquences sont indiquées à un kilohertz près.

Emissions RTTY

METEO

Les émissions météo sont codées. Reportez-vous à l'ouvrage de Joerg KLINGENFUSS 'AIR AND METEO CODE MANUAL' pour en savoir plus sur ces séries de chiffres et de lettres que vous allez recevoir. C'est assez intéressant...

Les quelques fréquences citées sont actives fréquemment, et ont été recues, en particulier, aux heures (locales) indiquées.

Les vitesses et shift utilisés sont de 50 Bauds et 850 Hertz.

3.195	Prague	22.20
4.013	Paris	21.15
4.489	Bracknell (GB)	22.45
5.740	Djedda (Arabie Saoudite)	21.45
8.085	Moscou	22.25
8.163	Paris	17.00

AGENCES DE PRESSE

Les agences de presse assurent la transmission de leurs communiqués, dans le monde entier, en utilisant le Radiotélétype (RTTY) et le code Baudot.

Les vitesses et shift utilisés sont essentiellement le 50 Bauds et 425 Hertz.

Les textes sont transmis en Anglais, Français, Espagnol, mais aussi dans des langues qui peuvent rendre le décodage impossible (Arabe etc.). Ne soyez donc pas étonnés si votre ordinateur affiche des suites de caractères semblant incohérentes...

Transmettant sur plusieurs fréquences, selon la propagation et les heures du jour, les agences émettent pratiquement en permanence mais pas toujours sur les fréquences et aux heures indiquées ci-dessous.

5.460	Rabat (Maroc) en 75 Bauds...	23.00
5.840	AFP (Paris)	20.40
5.859	ADN (DDR)	22.25
6.960	Sofia (Bulgarie)	21.05
6.972	AGERPRESS (Bucarest)	18.00
6.984	Associated Press (Londres)	19.00
7.806	TANJUG (Belgrade)	18.00
8.022	AFP	20.00
8.060	TASS (URSS)	23.00
8.140	TASS	21.10
9.396	AFP	19.00
10.649	Associated Press	18.00
12.315	TASS	17.45
15.655	MAP (Maroc)	12.15
18.668	AFP	14.30
20.313	AFP	15.05

EMISSIONS EN TELEGRAPHIE MORSE (CW)

Les émissions en CW sont très nombreuses sur le spectre de fréquences radio, aussi nous ne vous indiquerons que peu d'entre elles, le choix étant très vaste. Signalons que beaucoup de stations (dans les bandes marines en particulier), transmettent des identifications en permanence, ce qui vous permettra d'ajuster votre matériel. La bande la plus propice est celle des 8 MHz qui a l'avantage d'être 'ouverte' toute la journée.

Leur origine n'a pas toujours été identifiée par les auteurs.

2.680	Station Allemande Météo	23.00
3.999	Station Météo	23.50
6.929	Météo en Espagnol	10.45
8.550	Saint-Lys (France)	18.50

EMISSIONS EN FAC-SIMILE (FAX)

Les émissions en FAX sont assez peu connues et présentent pourtant l'intérêt d'obtenir, sur l'écran (ou papier), des cartes météo et parfois, des photos. Toutes ne seront pas faciles à interpréter...

En France, Bracknell (GB) et Paris sont les deux stations les plus faciles, et les plus intéressantes, à recevoir car elles transmettent des cartes de la situation météo sur notre région (au sens large).

2.618	Bracknell	Soir
3.289	Bracknell	Jour/Soir
4.248	Bracknell	Jour/Soir
8.040	Bracknell	Jour/Soir
4.047	Paris	Soir
8.185	Paris	Jour

Signalons également la station Francaise émettant sur 134 KHz qui, hélas, ne sera pas recue dans toute la France, mais fort bien en région Parisienne. Nous avons recu sur cette fréquence, depuis Rennes, de belles reproductions de photos METEOSAT, malgré le signal relativement faible.

REFERENCES BIBLIOGRAPHIQUES

La lecture des ouvrages suivants nous a permis de progresser assez rapidement avec notre CPC. Nous nous permettons donc de suggérer à nos lecteurs d'en faire l'acquisition (en fonction, bien entendu, de leurs besoins...).

Précisons que les titres cités ci-dessous ne représentent en rien une liste exhaustive ou guidée par quelque mobile publicitaire...

CONNAISSANCE DU SYSTEME

- Le livre du FIRMWARE

Edité par AMSOFT sous la référence SOFT 158, il décrit avec force détails toutes les routines du système, donne des informations sur la programmation des interruptions, les extensions ROM etc. Faute de pouvoir citer ici son contenu, nous pourrions résumer en disant que c'est l'ouvrage de Référence pour le programmeur désirant découvrir les secrets de l'AMSTRAD.

- Le FIRMWARE du disque

Suite à l'ouvrage précédent, le SOFT 158a, offre la description des routines du DOS et donnera au lecteur l'occasion de mieux exploiter son Drive. Tout ce que vous voudriez connaître sur la structure de CP/M et d'AMSDOS. Vous y découvrirez ainsi que le DOS supporte la gestion d'une interface série, ce qui n'est mentionné nulle part dans le manuel utilisateur...

- La bible du programmeur (chez MICRO APPLICATION).

Traduit de l'allemand, comme le reste de la collection de cet éditeur, la bible détaille tous les secrets du CPC. La ROM est commentée, les variables système, les points d'entrée, le Hard, vous y trouverez la réponse aux questions, sans doute nombreuses, que vous vous posez. Ecrit pour le 464, les utilisateurs des autres CPC y trouveront néanmoins leur compte...

- Le livre du lecteur de disquette (MICRO APPLICATION).

Même philosophie que la 'Bible du programmeur', il décrit avec beaucoup de détails, le Hard et le Soft du disque. Vous y découvrirez comment lire ou écrire directement un secteur, les secrets du contrôleur FDC 765 et un listing abondamment commenté de la ROM du DOS.

- Le tour de l'AMSTRAD (CEDIC-NATHAN).

Pour les lecteurs qui cherchent un complément au manuel utilisateur de la machine, et qui sont perdus dans la jungle des titres des ouvrages parus à ce jour, ce livre leur offrira la possibilité d'approfondir leurs connaissances.

- Clefs pour AMSTRAD (PSI).

Compilation de toutes les documentations existantes, et de beaucoup d'ouvrages publiés sur la machine, ce livre a le mérite (tout en restant sommaire) de proposer au lecteur une foule de renseignements dans un minimum de place. Si votre budget 'documentation' est un peu serré...

- Le livre du CP/M 2.2 et CP/M 3.0 (Micro Application).

Pour le lecteur désireux de faire connaissance avec CP/M, ce système d'exploitation présent sur bien d'autres machines, nous conseillons cet ouvrage, qui en est un véritable mode d'emploi. C'est, à notre avis, un excellent complément à la documentation que doit posséder l'utilisateur de l'AMSTRAD voulant évoluer.

- Montages, extensions et périphériques (Micro Application).

Des schémas de cartes, avec les implantations des composants et les reproductions des circuits imprimés, permettant de réaliser des entrées-sorties, conversions analogiques-numériques, RS 232, programmeur d'EPROM etc. Voilà un livre qui va ravir tous ceux qui savent utiliser le fer à souder et qui veulent offrir à leur AMSTRAD un environnement.

Signalons, enfin, la revue 'CPC', mensuel des Editions SORACOM, entièrement consacrée aux AMSTRAD: du Hard, des programmes utilitaires et des jeux, du BASIC, du PASCAL, de l'ASSEMBLEUR, et toutes les nouveautés concernant la famille des CPC.

QUELQUES QUESTIONS RELATIVES A CE LIVRE

1- Comment avez-vous connu l'existence de livre ?

.....
.....

2- A-t-il répondu à votre attente ?

.....

3- Si non, pourquoi ?

.....
.....
.....

4- Quels sujets auriez-vous souhaité voir traiter ?

.....
.....
.....

5- Quels sont les sujets qui vous ont :

Le plus intéressé ?.....

.....

Le moins intéressé ?.....

.....

Dans le but d'améliorer nos prochains ouvrages, retournez cette fiche, accompagnée d'autres remarques éventuelles, à l'éditeur.

EDITIONS SORACOM
BP 46
35170 BRUZ

Composition FIDELTEX
Maquette Soracom
Impression JOUVE, MAYENNE
N° d'éditeur : 48
N° 50090. Dépôt légal : Février 1988

« La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause, est illicite » (alinéa premier de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 du Code Pénal. »



PRIX:115F. TTC

COMMISSIONERS OF THE GENERAL LAND OFFICE

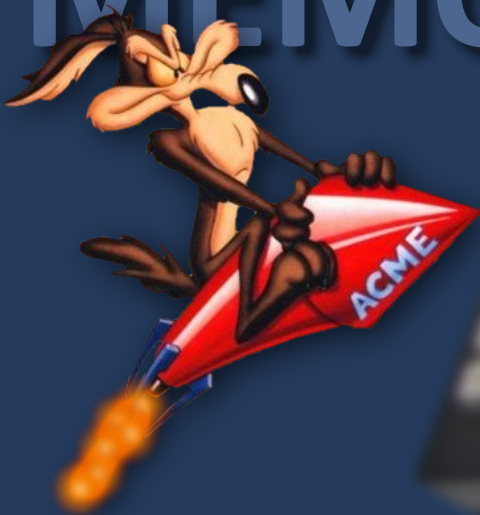


Document **numérisé**
avec amour par :

AMSTRAD

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>