

10/1.3.1

Des instructions évoluées pour votre MINITEL

Comme vous l'avez remarqué dans le programme du jeu PUISSANCE 4 présenté précédemment (voir Partie 10, chap. 1.3, p. 40 à 53), la gestion du MINITEL est relativement complexe. Par exemple : le positionnement à un endroit précis de l'écran exige l'envoi de trois codes : &31 (code US de positionnement), numéro de ligne (calculé en ajoutant la valeur &40 au numéro effectif), et numéro de colonne (calculé comme pour la ligne) ; l'inversion vidéo exige deux codes ; le passage en couleur, pour le fond comme pour le caractère, requière aussi deux codes, ce qui donne, pour huit couleurs disponibles : deux fois deux fois huit = 32 codes à retenir.

Sans parler de toutes les possibilités offertes par le MINITEL, on devine rapidement que la liste de codes hexadécimaux à retenir et à calculer va devenir rapidement fastidieuse.

Ne serait-il pas plus pratique de posséder des instructions parlantes, que l'on retrouve sur l'Amstrad-CPC, telles LOCATE pour le positionnement, INK pour la couleur de caractère, PAPER pour la couleur de fond. C'est ce que nous vous proposons ci-après.

Nous avons donc décidé de créer un jeu d'instructions, relativement facile à retenir, qui vous permettra de créer vos programmes beaucoup plus facilement.

Nous avons pour cela utilisé la possibilité des CPC de pouvoir ajouter des instructions Basic supplémentaires sous forme de RSXs. Ainsi chaque commande sera précédée du symbole ù ou 'selon le modèle de votre micro-ordinateur ; par exemples 'LOCATE, ligne, colonne, 'INK, couleur,...

I. Le nouveau jeu d'instructions

Nous allons étudier ici toutes les possibilités intéressantes d'instruction de gestion de l'écran du MINITEL, depuis l'envoi d'un caractère, d'une suite de caractères, de localisation, soulignement, couleur, effacement, masquage, etc.

Nous étudierons, selon l'intérêt et la complexité, les algorithmes et algorithmes des différents programmes, qui seront réalisés en Assembleur. (Que les programmeurs Basic n'abandonnent pas de suite, il leur sera fourni un chargeur de codes hexadécimaux pour utiliser les instructions).

!SEND (ENVOYER)

Syntaxe : !SEND, code
 !SEND, var%

L'instruction !SEND permettra d'émettre vers le MINITEL un code ASCII unique, soit explicite, soit contenu dans une variable entière.

Ainsi !SEND,&41 permettra l'affichage du caractère A sur l'écran du MINITEL, le code ASCII de A étant &41 I% = &42 : !SEND, I% affichera le caractère B.

Signalons qu'il est tout à fait possible d'envoyer par cette instruction des caractères de contrôle (!SEND, &0C effacera l'écran), et même des codes calculés (!SEND,&40 + &06).

Les programmeurs qui sont peu habitués aux codes hexadécimaux, mais qui préfèrent les codes décimaux, utiliseront l'instruction de la même façon (!SEND,65 affichera aussi le caractère A), et pourquoi pas en binaire.

Le programme de gestion est quasiment identique au programme d'envoi d'un code appelé par la commande CALL &A00,code, si ce n'est la définition de l'instruction dans un tableau.

!WRITE (ECRIRE)

Syntaxe : !WRITE, "Chaîne de caractères"
 !WRITE, variables

L'instruction WRITE permettra d'afficher sur le MINITEL une chaîne de caractères, soit explicitement écrite, soit par l'intermédiaire d'une variable chaîne de caractères.

Vous remarquez qu'une fois la chaîne de caractères émise par cette instruction, le curseur, même s'il n'est pas visualisé, se trouvera toujours positionné juste à droite du dernier caractère affiché, aucun passage à la ligne n'étant effectué.

Ainsi : 10 !WRITE,"Chaîne numero 1"
 20 !WRITE,"Chaîne numero 2"

affichera le texte : Chaîne numero 1 Chaîne numero 2 sur la même ligne et sans espace séparateur.

Ce qui pourrait sembler être ici un inconvénient, est parfois utile, par exemple pour afficher un texte suivi d'un nombre, comme dans l'instruction du Basic Locomotive

```
10 PRINT "Texte";variable.
```

il suffira d'écrire :

```
10 !WRITE,"Texte"  
20 !WRITE,STR$ (variable)
```

ou

```
10 !WRITE,"Texte" + STR$ (variable)
```

Le programme créé reprend le principe du programme assembleur appelé pa la commande CALL &A010, variable\$.

!SENDSEQ (ENVOI SÉQUENCE)

Syntaxe : !SENDSEQ,code1,code2,code3, ...
 !SENDSEQ,variable1%,variable2%,variable3%, ...

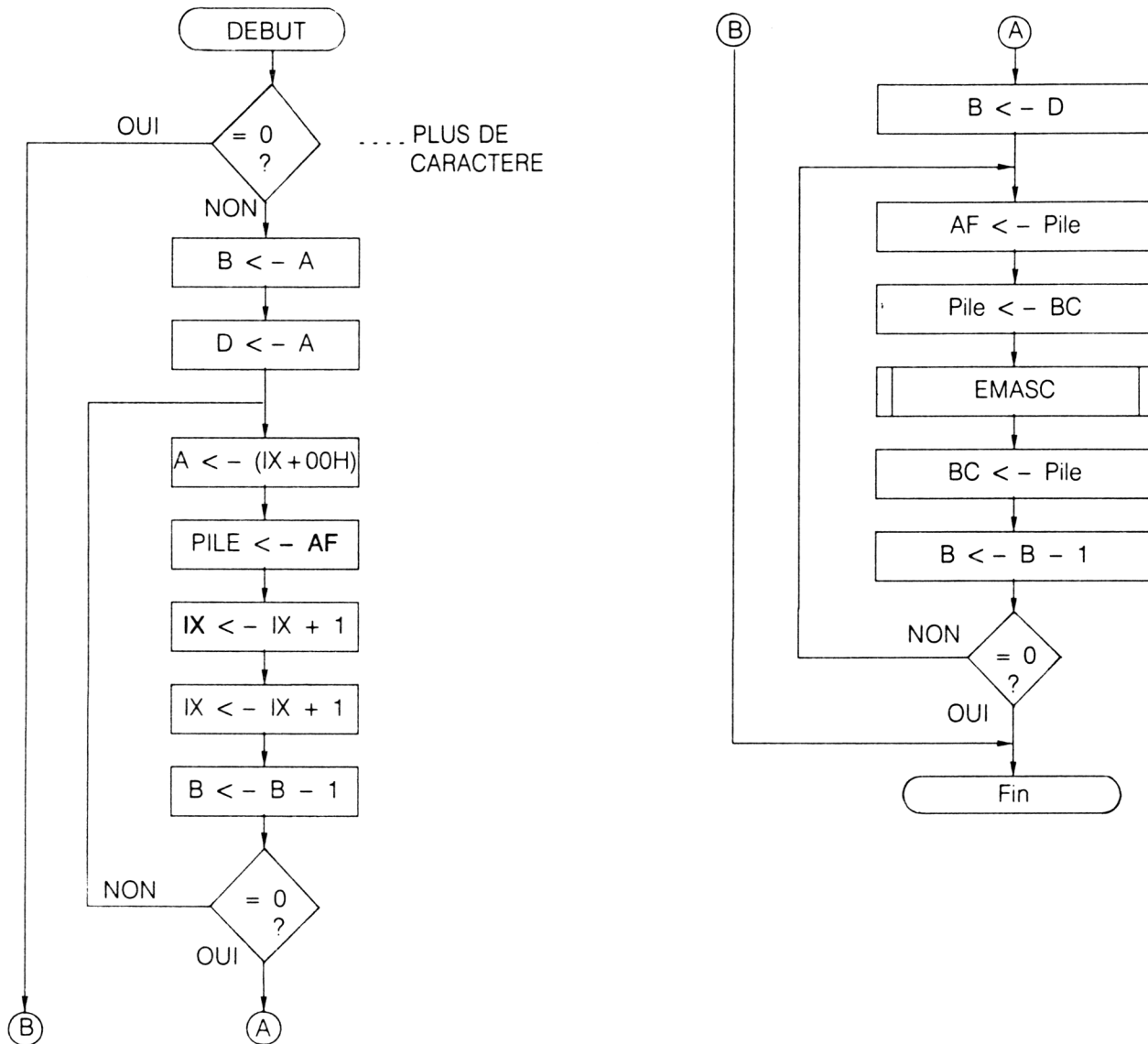
Cette instruction permet d'envoyer l'une après l'autre une suite de codes hexadécimaux (ou décimaux), soit explicites, soit contenus dans des variables entières, soit panachés (on pourrait très bien trouver : !SENDSEQ,code1,variable1%,code2, ...).

Il est possible d'émettre jusque 32 codes consécutifs par cette commande, limite maximale imposée par le passage de variables des instructions de type RSX.

Le programme de gestion fera appel, autant de fois qu'il y aura de caractères à envoyer (ce nombre se trouvant dans le registre A, dès l'entrée dans le programme Assembleur), à la procédure d'envoi d'un caractère appelée par !SEND ; ce qui nous donnera l'algorithme général suivant :

- DEBUT
 - SI il y a au moins un code transmis
 - POUR I DEPUIS 1 JUSQUE le dernier code
 - Récupérer un code dans la pile
 - PROCEDURE EMETTRE UN CODE ASCII
 - FINPOUR
 - FINSI
- FIN

L'ordinogramme issu de cet algorithme est donné ci-après :



!PRINT (AFFICHE LIGNE)

Syntaxe : !PRINT, "Chaîne de caractères"
!PRINT, variable\$

Cette instruction permet l'affichage d'une chaîne de caractères sur l'écran du MINITEL, mais cette fois-ci elle émet automatiquement en fin de ligne un retour chariot et un passage à la ligne inférieure.

Elle est équivalente à la suite d'instruction suivante :

```

10 !WRITE,"Chaîne de caractères"
20 !SEND,&0D : REM RETOUR CHARIOT
30 !SEND,&0A : REM LINE FEED
  
```

!RECEPT (RÉCEPTION)

Syntaxe : !RECEPT, @variable%

Cette instruction permet d'attendre la frappe d'un caractère frappé sur le clavier du MINITEL. Le code reçu est ainsi retourné sous forme numérique dans la variable entière **variable%**, qui doit auparavant avoir été affectée (par exemple par l'instruction **variable% = 0**), sous peine d'un message d'erreur.

Le programme fonctionne de façon identique au sous-programme Assembleur appelé par la commande.

CALL &A02B,@variable%

Signalons qu'il n'est possible ici que de recevoir un code unique, aussi il est conseillé d'utiliser des variables contenues dans un tableau indicé pour recevoir une suite de codes, ainsi le programme :

```
10 DIM VARIABLE% (10)
20 FOR I = 1 TO 10
30 VARIABLE%(I) = 0
40 NEXT I
50 FOR I = 1 TO 10
60 !RECEPT,@VARIABLE% (I)
70 NEXT I
```

permettra de recevoir les dix codes de dix touches frappées sur le MINITEL.

Afin de reconstituer les caractères frappés, il sera utile d'utiliser l'instruction **STR\$**, pour placer ceux-ci dans les variables alphanumériques :

```
80 FOR I = 1 TO 10
90 VARIABLE$(I) = STR$(VARIABLE%(I))
100 NEXT I
110 REM *** N'OUBLIER PAS DE DÉFINIR***
120 REM *** LE TABLEAU VARIABLE$(10) ***
```

Chacun des caractères n'est pas ré-émis vers le MINITEL, ce qui implique que lorsque le protocole de transfert entre modem et écran est bloqué, l'utilisateur du MINITEL travaille en aveugle, ce qui implique qu'un écho doit être effectué par l'intermédiaire de la prise péri-informatique, que nous étudierons ultérieurement.

RECAR (RÉCEPTION DE CARACTÈRES)

Syntaxe : !RECAR, @variable\$

Cette instruction, est une version plus évoluée de l'instruction précédente, car elle permet de recevoir une chaîne de caractères comprise entre 1 et 255 codes, avec possibilité de corrections ou d'annulations. Une validation finale est attendue, et un écho de chaque caractère est effectué.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

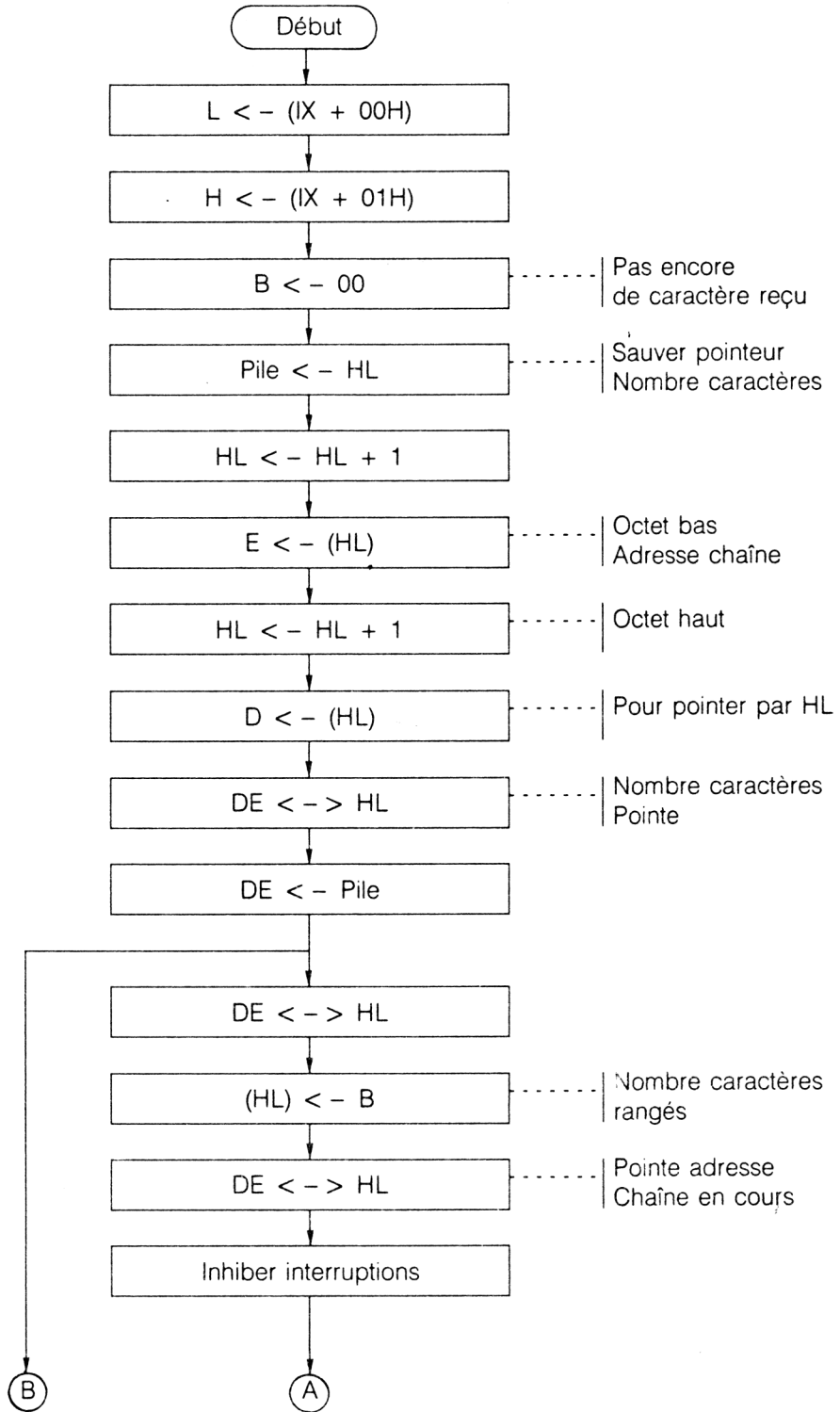
La variable **variable\$** doit auparavant avoir été définie, nous vous conseillons de prendre la taille maximale par **variable\$ = SPACE\$(255)**.

Lors de l'exécution, le clavier du CPC est inopérant ; l'utilisateur du MINITEL frappe la phrase qu'il désire, et a la possibilité de corriger le dernier caractère entré à l'aide de la touche **<CORRECTION>** (celui-ci est alors effacé, et le minitel à nouveau en position d'attente) ; il peut aussi décider d'annuler entièrement la phrase déjà entrée à l'aide de la touche **<ANNULATION>**, qui efface tous les caractères et se replace en position d'attente.

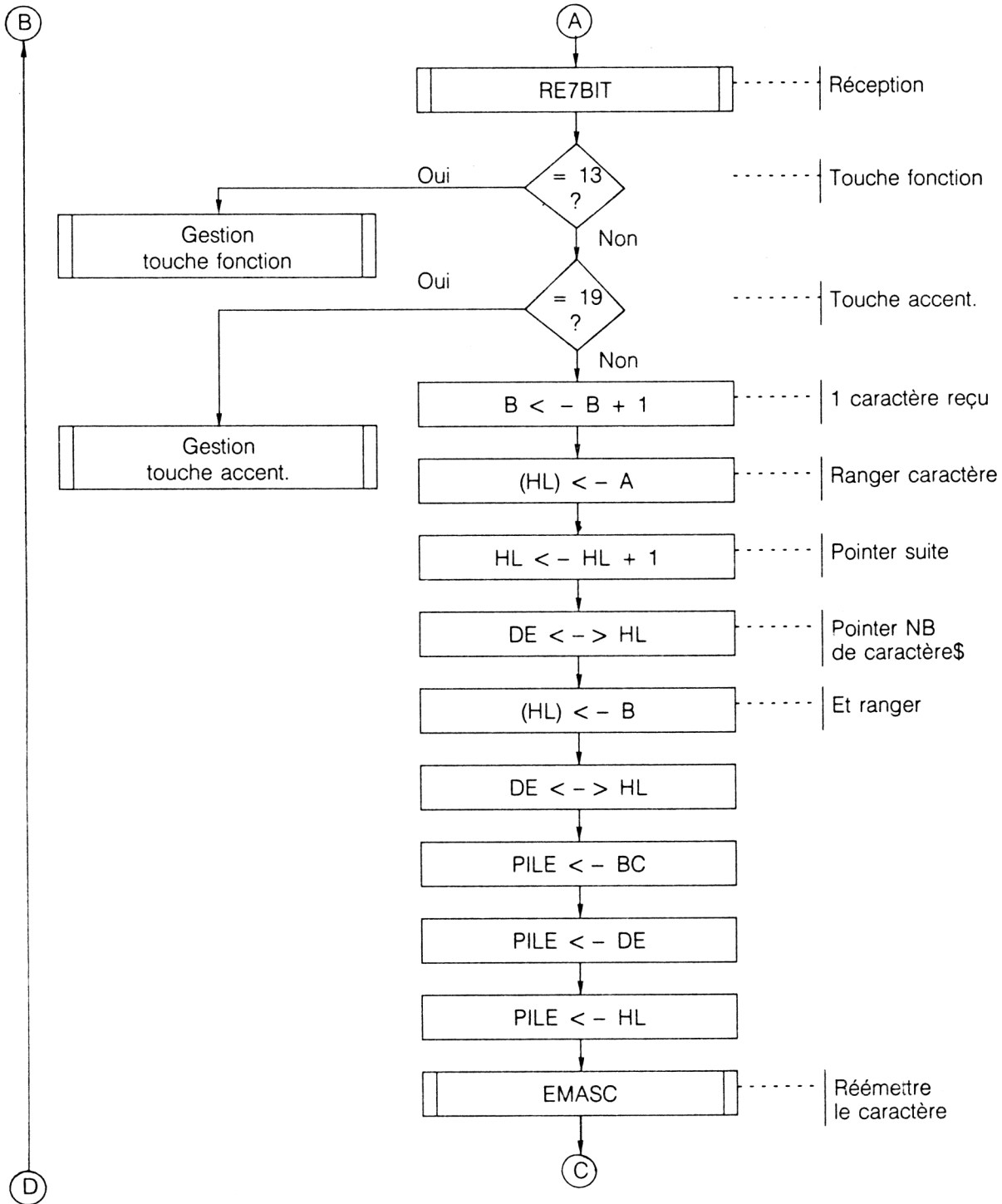
La main ne sera redonnée à l'Amstrad-CPC que par une validation de la phrase à l'aide de la touche **<ENVOI>** (aucune correction ou annulation ne sera alors possible), et les caractères entrés seront transmis dans **<variable\$>**.

Nous vous proposons d'étudier plus en détail la procédure réalisant cette nouvelle instruction.

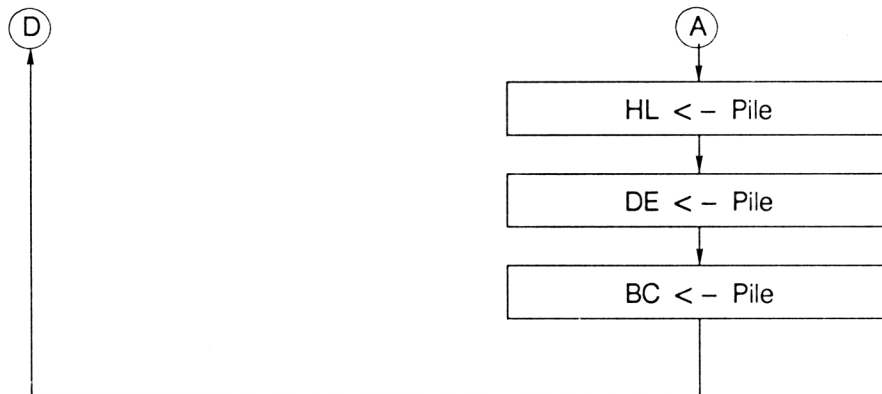
Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



On charge d'abord dans le registre double **HL** l'adresse du pointeur de la variable fournie dans la pile paramètres, et pointée par **IX + 00H** et **IX + 01H**.

Cette adresse contiendra le nombre de caractères de la chaîne, fixée à 00 par le registre **B**.

Une fois sauvée dans la pile du Z80, on récupère dans le registre double **DE** l'adresse de la chaîne pointée par **HL + 1** et **HL + 2** (double incrémentation de **HL**).

Après un savant échange pour utiliser **HL** comme pointeur des caractères de la variable, on récupère dans **DE** l'adresse du nombre de caractères.

Les pointeurs et initialisations ainsi fixés, on entre dans une boucle lisant les caractères.

Nouvel échange de **DE** et **HL**, sauvegarde du nombre de caractères, et encore un échange.

On inhibe les interruptions pour ne pas perdre de caractères lors de l'appel de la procédure de réception **RE7BIT**.

Le caractère reçu est testé. Si sa valeur est égale à **&13**, alors on est en présence d'une touche de fonction qui demandera un traitement spécial.

De même la valeur **&19** signalera l'arrivée d'un caractère accentué, nécessitant une gestion particulière.

Sinon, nous supposons que le caractère est valable, donc on incrémentera le registre **B** (un caractère de plus).

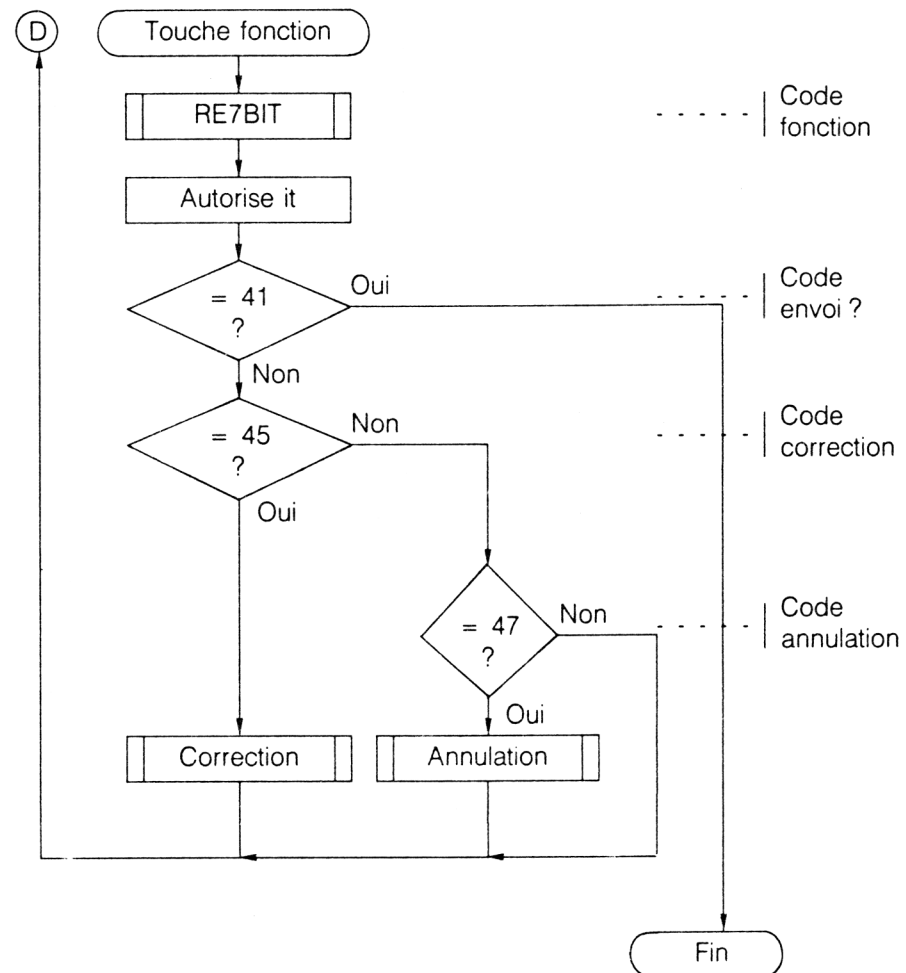
Ce caractère sera rangé dans la chaîne pointée par **HL**, que l'on incrémente aussitôt.

Nouvel échange savant entre **DE** et **HL** pour sauver le nombre de caractères à l'adresse du pointeur de variable, encore suivi d'un échange.

On empile ensuite les trois registres doubles utiles, pour réémettre en écho le caractère, par la procédure **EMASC** puis dépile. Le tour est bouclé !

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Intéressons-nous maintenant au traitement d'une touche fonction du MINITEL, décrit ci-après :



Il existe 11 touches fonctions dont une majorité envoie une séquence de codes sur la ligne péri-informatique, ou sur la ligne téléphonique lorsque le MINITEL est connecté sur un serveur utilisant le réseau commuté ou TRANSPAC.

CONNEXION/FIN	émet le code &13 - &49
REPETITION	émet le code &13 - &43
SOMMAIRE	émet le code &13 - &46
GUIDE	émet le code &13 - &44
ANNULATION	émet le code &13 - &45
CORRECTION	émet le code &13 - &47
ENVOI	émet le code &13 - &41
RETOUR	émet le code &13 - &42
SUITE	émet le code &13 - &48
LOUPE	n'émet aucun code en ligne, mais garde une action locale.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

La touche spéciale non marquée (touche **SHIFT**) ne provoque pas non plus d'émission en ligne, sauf lorsqu'on appuie simultanément sur une autre touche.

Ne nous intéressant pas pour l'instant, sachez que des procédures spéciales, par des séquences de codes différents peuvent être actionnées, en appuyant simultanément sur la touche spéciale et certaines touches de fonction : **REPETITION**, **CORRECTION** et **CONNEXION/FIN**.

Parmi ces touches, nous nous intéresserons plus particulièrement à **ENVOI**, **CORRECTION** et **ANNULATION**.

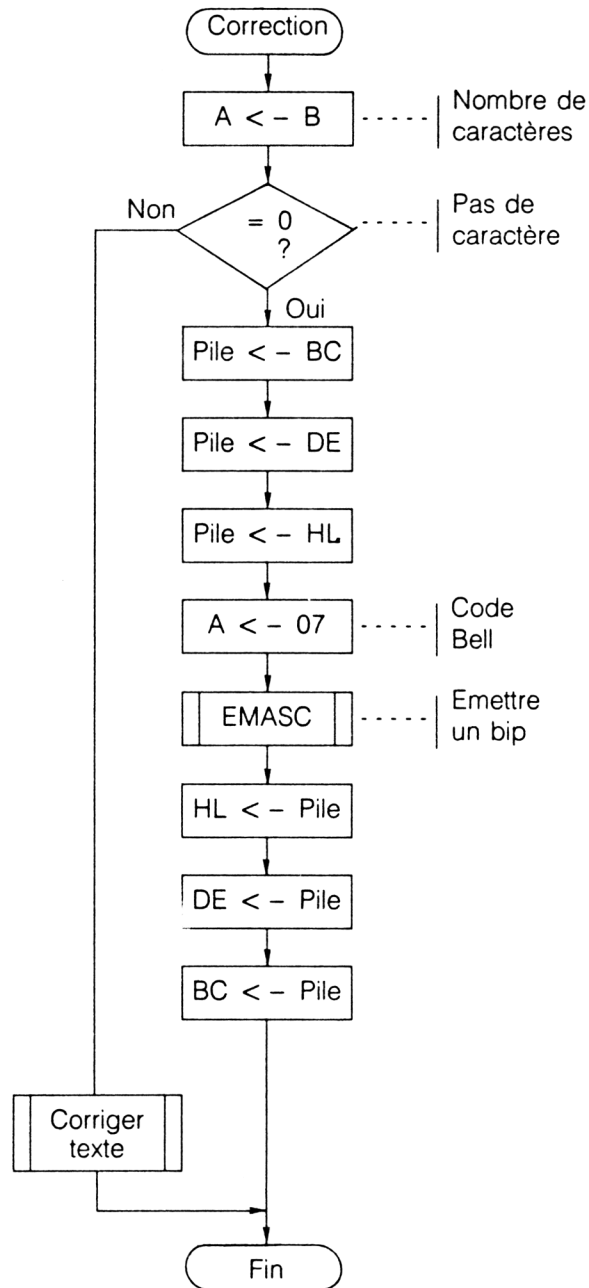
Notre traitement va donc lire un nouveau code à l'aide de la procédure **RE7BIT**.

Si le deuxième code reçu est **&41**, alors, l'utilisateur a appuyé sur **ENVOI**, pour valider son texte, ce qui produira un retour au Basic (FIN du traitement).

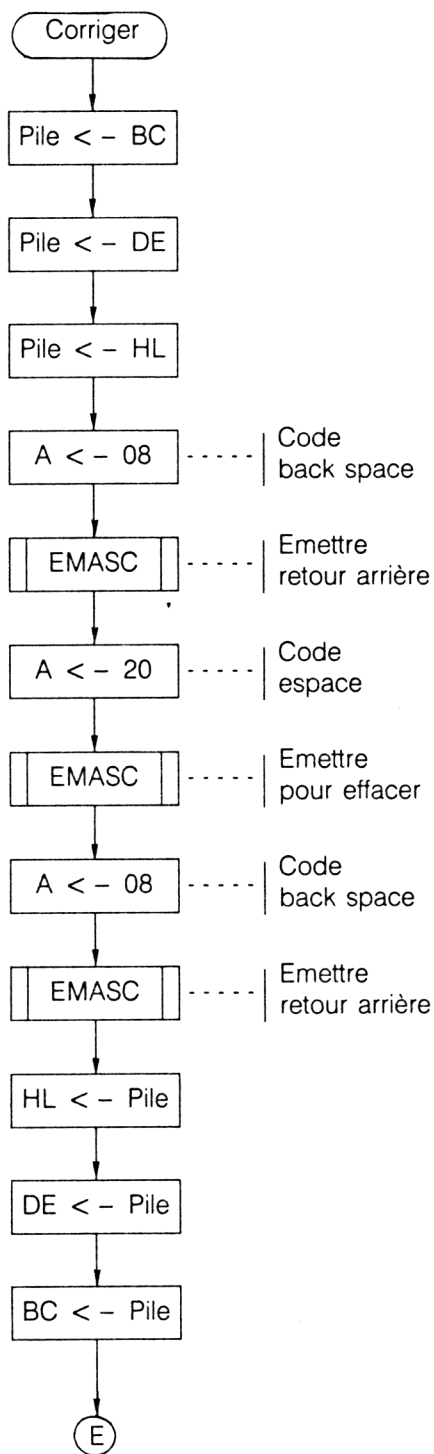
Par contre **&45** signalera une correction, et **&47** l'annulation du texte.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

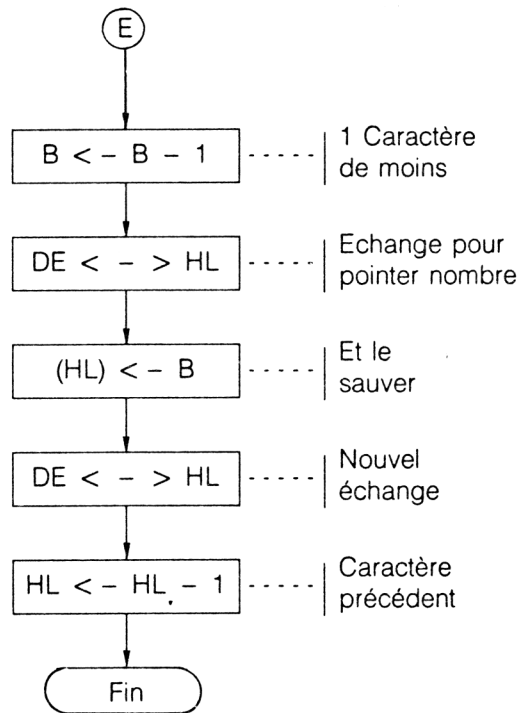
Les ordinogrammes suivants décrivent la correction :



Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



On charge le nombre de caractères dans le registre **A** pour tester s'il y a au moins un caractère.

Dans ce cas on corrige le texte (attention, le test est inversé, pour une facilité de programmation).

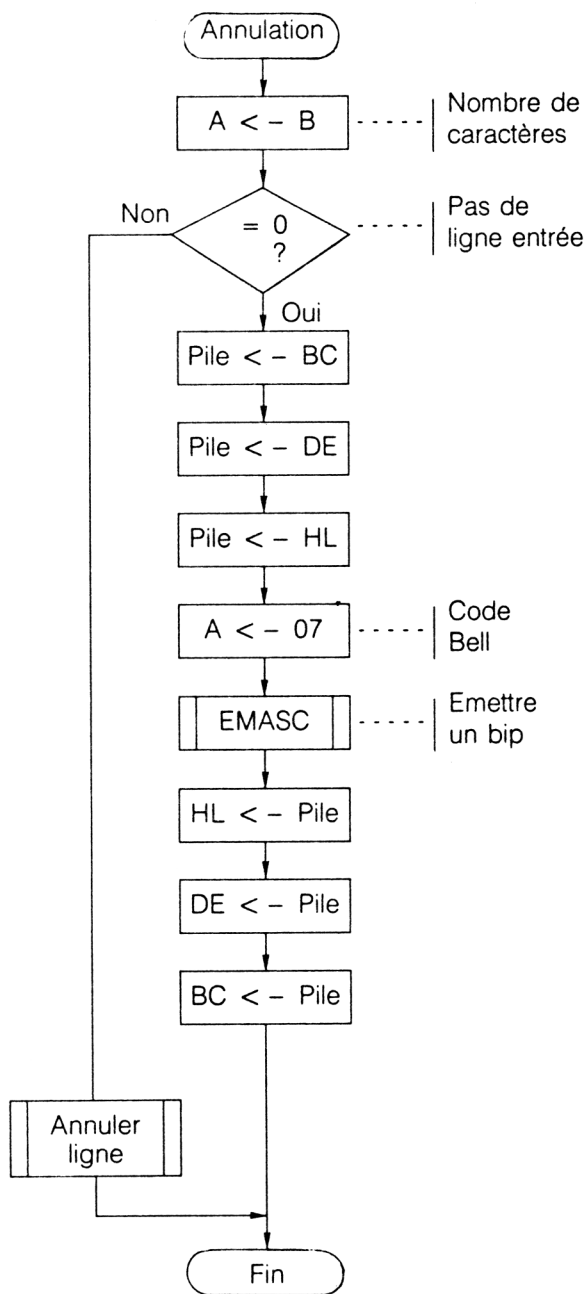
Sinon, il sera émis un BIP (code ASCII &07) par l'intermédiaire de la procédure **EMASC**, après empilage, et avant dépilage des registres doubles.

Pour corriger, après sauvegarde des registres, on émet un retour arrière (**BACK SPACE**, de code &08), un espace pour l'effacer (code &20), et un nouveau retour arrière pour se repositionner sur le caractère effacé. On récupère ensuite les registres.

Il faut maintenant décrémenter le nombre de caractères dans **B**, et par un nouvel échange (toujours aussi savant !), ranger dans le pointeur de la variable. On positionne aussi sur la variable précédente par décrémentation du registre **HL**.

La **FIN** renvoie à la boucle principale de réception de caractères.

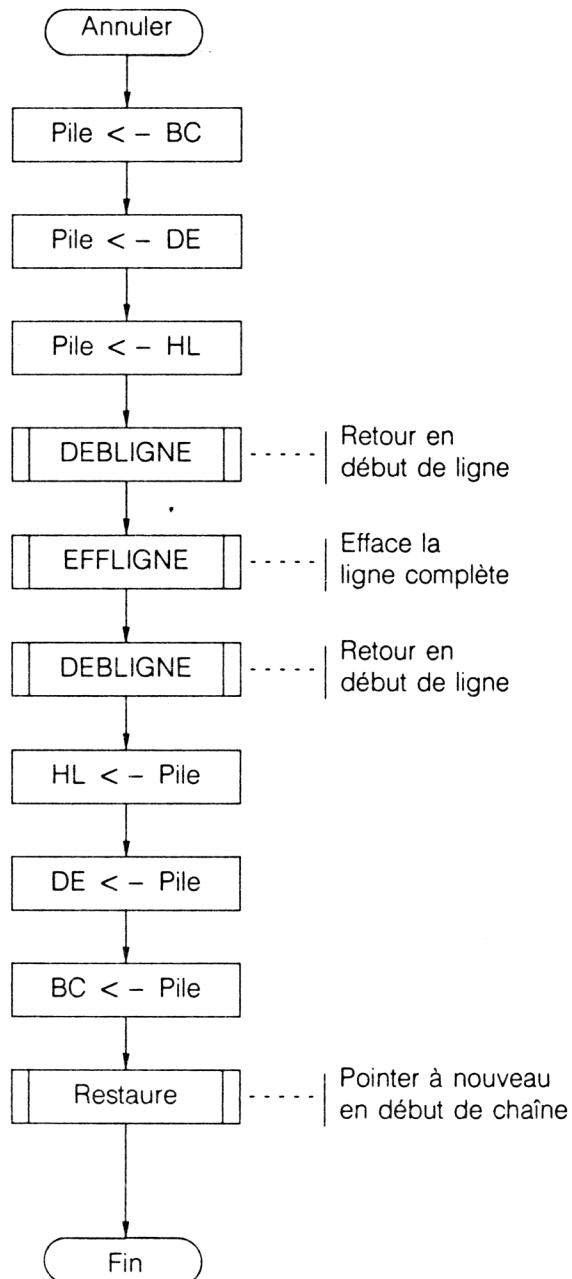
Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



L'ordinogramme ci-dessus reprend le principe de celui de la page 12 pour gérer le test de la possibilité d'annulation d'une chaîne de caractères, s'il en a été frappé au moins un, sinon, émission du caractère **BELL** provoquant un BIP.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Pour annuler une chaîne, nous allons utiliser l'ordinogramme décrit ci-après :

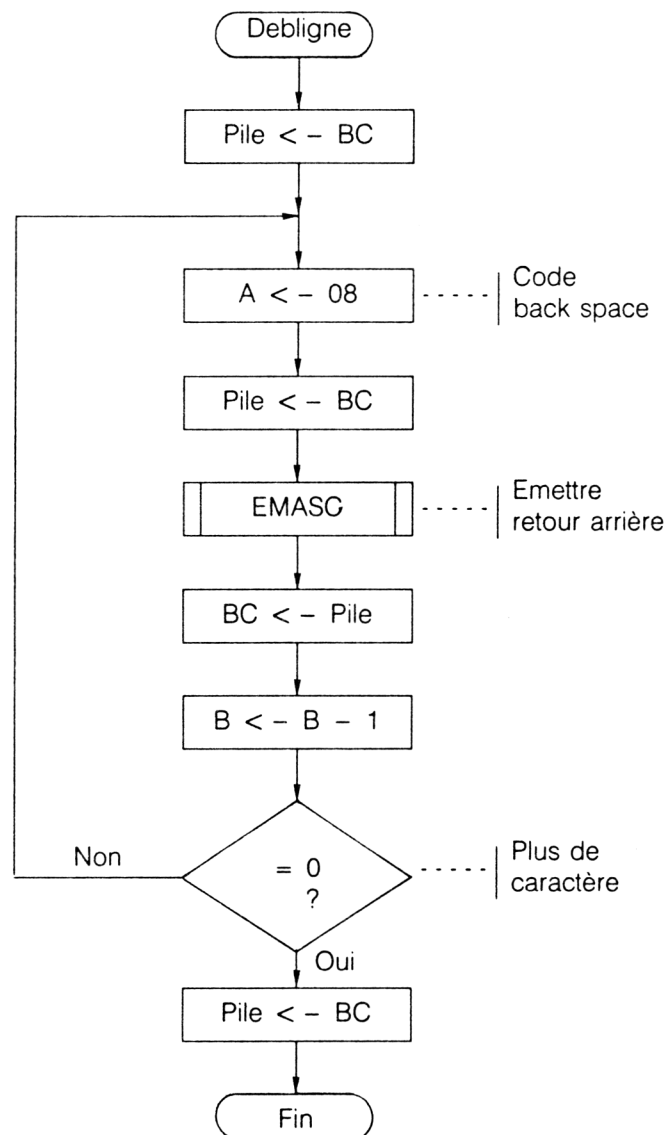


Une fois les registres sauvés, (on les récupère plus tard), on retourne en début de ligne, pour l'effacer, et le curseur revient sur le premier caractère à entrer. En fin de traitement, il faut restaurer le pointeur de variable au nombre de caractères initial, et pointer le début de la chaîne.

Le retour en début de ligne s'effectuera selon l'algorithme suivant :

- DEBUT
 - POUR I DE 1 JUSQU'AU nombre de caractères
frappés
 - Retourner d'un caractère en arrière
 - CO
à l'aide de BACK SPACE
 - FINCO
 - FINPOUR
- FIN

Ce qui donnera l'ordinogramme suivant qui inclut en plus, une sauvegarde des registres :

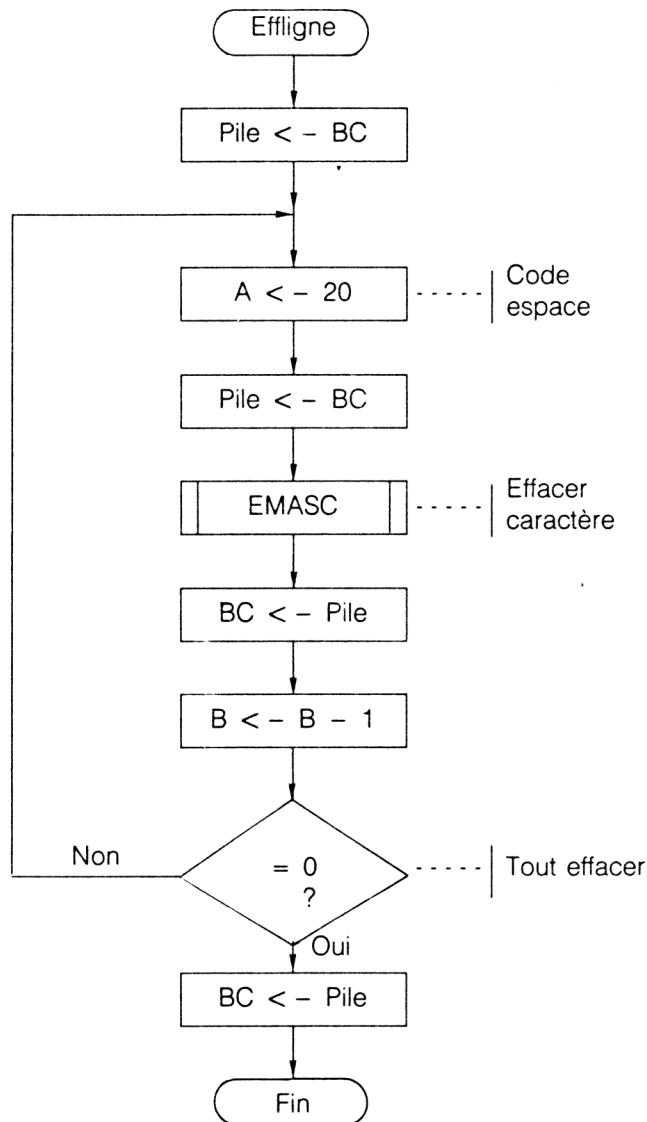


Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

La procédure d'effacement se déroulera selon l'algorithme suivant :

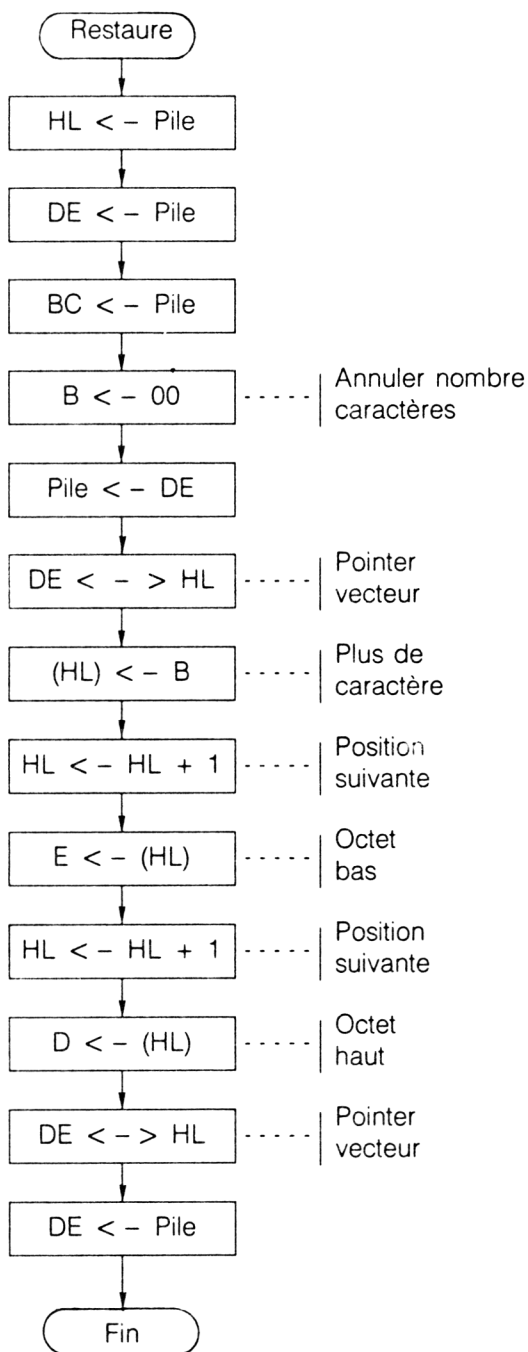
- DEBUT
 - POUR | DE 1 | JUSQU'AU nombre de caractères frappés
 - Emettre le caractère ESPACE
 - FINPOUR
- FIN

Ce qui permet de tracer l'ordinogramme suivant :



Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

L'ordinogramme ci-après décrit la séquence de restauration des différents pointeurs : le nombre de caractères doit être remis à zéro (on vient d'annuler, si nos souvenirs sont bons), et il faut pointer en début de la chaîne à entrer au MINITEL, le tout agrémenté de plus en plus de savants échanges entre les registres doubles **DE** et **HL**.



LOCATE (POSITIONNEMENT)

Syntaxe : :LOCATE,X,Y
 :LOCATE,varX%,varY%

Cette instruction positionne le curseur en ligne X, colonne Y, tout comme l'effectue l'instruction de même nom du Basic Locomotive.

Le numéro de ligne peut être compris entre 0 (ligne supérieure et 24 (dernière ligne en bas), le numéro de colonne se situera entre 1 (à gauche) et 40 (à droite).

Pourquoi la ligne zéro ? En fait la ligne supérieure est une ligne spéciale, initialement réservée aux messages, et dont l'accès est limité, car elle fonctionne différemment par rapport aux autres lignes (pas de possibilité de scrolling, pas de passage automatique à la ligne suivante après la colonne 40). La particularité de la ligne 0 est qu'elle ne peut être quittée que par l'exécution d'une autre instruction :LOCATE, ou par un LINE FEED (LF), qui renvoie à la position précédent l'accès à la ligne 0.

En effet l'écran du MINITEL est divisé en deux (voir figure 1).

Pour la réalisation de notre programme, il a fallu exécuter une conversion entre les numéros de ligne/colonne de la fonction :LOCATE, et les numéros effectifs sur le MINITEL.

Il fallut pour cela ajouter la valeur hexadécimale &40 aux nombres fournis, sachant que l'on peut aller de &40 à &40 + &18 = &50 (&18 = 24 lignes), ce qui nous donne 25 lignes en comptant la ligne 0, et de &41 à &69, ce qui nous donne 40 colonnes.

La localisation est ainsi effectuée par l'envoi de la séquence de caractères équivalente aux instructions suivantes :

```
10 :SEND,&1F : REM CODE US DE LOCALISATION
```

```
20 :SEND,X + &40 : REM COORDONNEE LIGNE
```

```
30 :SEND,Y + &40 : REM COORDONNEE COLONNE
```

Les erreurs éventuelles de débordement sont signalées sur l'écran de l'AMSTRAD-PC par le message "ERREUR DE COORDONNEES" et la non exécution de l'instruction, ceci sans interruption du programme en cours.

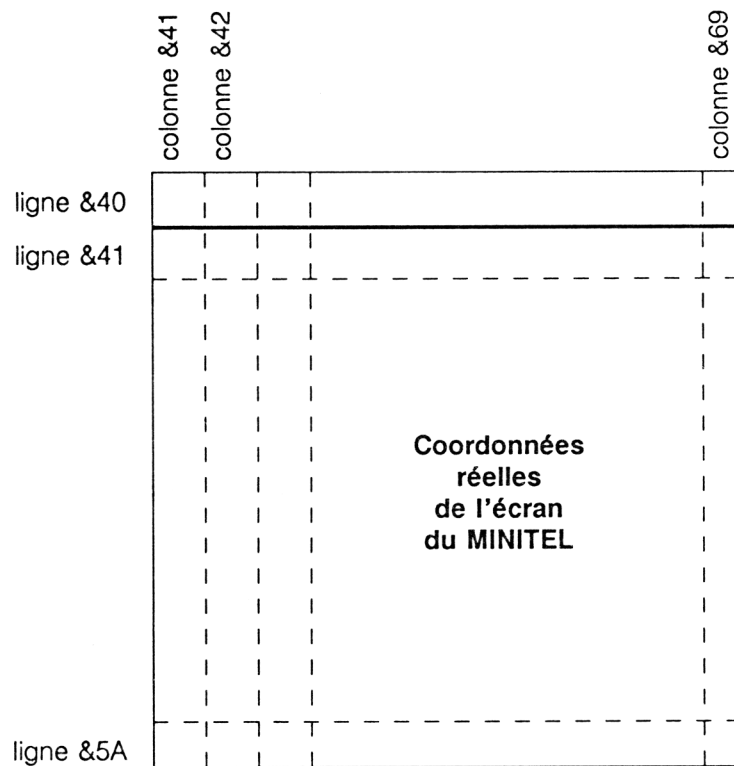
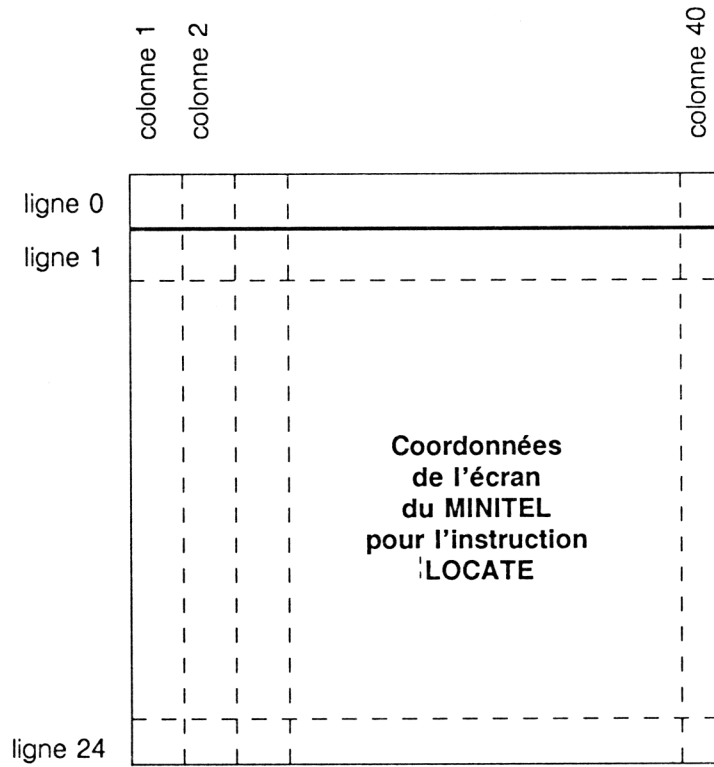
:CURSON (CURSEUR ALLUMÉ)

Syntaxe : :CURSON

Cette instruction permet de visualiser le curseur clignotant, qui apparaît comme l'alternance du caractère se trouvant sous le curseur, et la couleur de fond inversée.

L'allumage ou l'inversion du curseur dure environ une demi seconde, et n'est pas réglable.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD



L'équivalent de cette instruction est l'envoi du code &11, nommé DC1 sur le MINITEL :

```
10 !SEND,&11
```

!CURSOFF (CURSEUR ÉTEINT)

Syntaxe : !CURSOFF

Contrairement à !CURSON, cette instruction inhibe la visualisation du curseur actif, et est équivalente à l'envoi du code de DC4 (&14) :

```
10 !SEND,&14
```

!LOWVIDEO (INVERSION VIDÉO)

Syntaxe : !LOWVIDEO

Cette instruction permet d'afficher un texte en inversion vidéo sur l'écran du MINITEL, tout comme le réalise l'instruction PRINT CHR\$(24) sur le CPC.

Les caractères apparaissent donc inscrits avec la couleur du fond, le fond prenant la couleur du caractère.

Remarque :

Lors de l'utilisation de l'instruction !LOCATE l'effet de !LOWVIDEO est perdu.

Deux ou plusieurs instructions !LOWVIDEO consécutives sont équivalentes à une seule, il n'y a pas inhibition (voir !NORMVIDEO).

L'équivalent de cette instruction est :

```
10 !SEND,&1B : REM CODE ESCAPE
```

```
20 !SEND,&5D : REM INVERSION FOND
```

NORMVIDEO (VIDÉO NORMALE)

Syntaxe : !NORMVIDEO

Cette instruction inhibe le mode d'affichage en inversion vidéo, et permet un retour aux couleurs de fond et de caractères définis précédemment. Elle est semblable à une deuxième instruction PRINT CHR\$(24) du CPC.

L'équivalent de cette instruction est :

```
10 !SEND,&1B : REM CODE ESCAPE
```

```
20 !SEND,&5C : REM FOND NORMAL
```

!INK (COULEUR D'ENCRE DE CARACTÈRE)

Syntaxe : !INK,couleur
!INK,varcouleur%

Cette instruction permet de fixer la couleur de caractère.

Huit possibilités s'offrent à vous :

!INK,0 : caractère noir
!INK,1 : caractère rouge
!INK,2 : caractère vert
!INK,3 : caractère jaune
!INK,4 : caractère bleu
!INK,5 : caractère magenta
!INK,6 : caractère cyan
!INK,7 : caractère blanc

La majorité des utilisateurs de la télématique ne possède pas de MINITEL couleur, nous vous donnons, ci-dessous, les correspondances entre les couleurs et les niveaux de gris :

Noir : 0 %
Bleu : 40 %
Rouge : 50 %
Magenta : 60 %
Vert : 70 %
Cyan : 80 %
Jaune : 90 %
Blanc : 100 %

Dans le programme, l'accès aux couleurs de caractères s'effectue à l'aide du caractère d'échappement (ESCAPE = &1B), et il faut ajouter la valeur &40 aux numéros d'encres décrits ci-dessus.

Ainsi on accède aux différentes couleurs de caractères par l'équivalent du programme suivant :

```
10 !SENDSEQ,&1B,&40 :REM CARACTERE NOIR
20 !SENDSEQ,&1B,&41 :REM CARACTERE ROUGE
30 !SENDSEQ,&1B,&42 :REM CARACTERE VERT
40 !SENDSEQ,&1B,&43 :REM CARACTERE JAUNE
50 !SENDSEQ,&1B,&44 :REM CARACTERE BLEU
60 !SENDSEQ,&1B,&45 :REM CARACTERE MAGENTA
70 !SENDSEQ,&1B,&46 :REM CARACTERE CYAN
80 !SENDSEQ,&1B,&47 :REM CARACTERE BLANC
```

Dans l'instruction !INK, les erreurs sont signalées sur l'écran du CPC (numéro de couleur inexistante), par le message « ERREUR COULEUR DE CARACTERE », et l'instruction n'est pas exécutée, sans interrompre le programme.

Remarque :

Lors d'un positionnement par l'instruction !LOCATE, la couleur de caractère est perdue.

!PAPER (COULEUR D'ENCRE DU FOND)

Syntaxe : !PAPER,couleur
 !PAPER,,varcouleur%

Cette instruction permet d'affecter au fond de l'écran les huit couleurs possibles. La liste est identique à celle décrite par l'instruction **INK**.

Il faut savoir que la couleur de fond, une fois définie est latente, c'est-à-dire qu'elle n'est exécutée que lors de l'apparition d'un espace (code &20), ou d'un caractère graphique alphamosaïque.

On retrouvera les mêmes niveaux de gris que pour les couleurs de caractères, dans le cas des MINITELS noir et blanc.

L'accès aux couleurs de fond s'effectue par le code d'échappement, et on aurait le programme suivant pour les différentes couleurs :

```
10 !SENDSEQ,&1B,&50 :REM FOND NOIR
20 !SENDSEQ,&1B,&51 :REM FOND ROUGE
30 !SENDSEQ,&1B,&52 :REM FOND VERT
40 !SENDSEQ,&1B,&53 :REM FOND JAUNE
50 !SENDSEQ,&1B,&54 :REM FOND BLEU
60 !SENDSEQ,&1B,&55 :REM FOND MAGENTA
70 !SENDSEQ,&1B,&56 :REM FOND CYAN
80 !SENDSEQ,&1B,&57 :REM FOND BLANC
```

Dans l'instruction **!PAPER**, les erreurs sont signalées sur l'écran du CPC (numéro de couleur inexistante), par le message "ERREUR COULEUR DE FOND", et l'instruction n'est pas exécutée, sans interrompre le programme.

Remarque :

Lors d'un positionnement par l'instruction **!LOCATE**, la couleur de fond actuelle est perdue.

Lors d'un positionnement où une couleur de fond a été définie et validée, le nouveau texte possède cette couleur de fond, sans qu'il soit besoin de la définir à nouveau.

!CLS (EFFACEMENT)

Syntaxe : !CLS

Cette instruction efface la partie d'écran comprise entre les lignes 1 et 24, et entre les colonnes 1 et 40.

La couleur de fond prise par cet effacement, est la couleur à la mise sous tension, c'est-à-dire le noir.

Tous les attributs définis précédemment (couleur de fond, caractères, soulignement, masquage, grandeur, etc.) sont perdus.

Seule la ligne zéro n'est pas affectée.

L'équivalent de cette instruction est l'envoi du code &0C (Form Feed) :

```
10 SEND,&0C
```

!ALLCLS (EFFACEMENT NORMAL)

Syntaxe : !ALLCLS

Cette instruction efface totalement l'écran du MINITEL, y compris la ligne supérieure.

Elle est l'équivalent d'un accès à la ligne zéro, avec remplissage de cette ligne avec des espaces, suivi d'une instruction !CLS :

```
10 !LOCATE,0,1
20 FOR I = 1 TO 40
30     !SEND,&20
40 NEXT I
50 !LOCATE,1,1
60 !CLS
```

Comme pour !CLS, tous les attributs définis sont perdus.

!G0 (JEU DE CARACTÈRE G0)

Syntaxe : !G0

Cette instruction sélectionne le jeu de caractère G0, c'est-à-dire les caractères ASCII traditionnels.

L'équivalent de cette instruction est l'envoi du caractère SI (Shift In de code &0F).

```
10 !SEND,&0F
```

!G1 (JEU DE CARACTÈRE G1)

Syntaxe : !G1

Cette instruction sélectionne le jeu de caractère G1, c'est-à-dire le jeu de caractères graphiques alphamosaïques.

L'équivalent de cette instruction est l'envoi du caractère SO (Shift OUT de code &0E) :

```
10 !SEND,&0E
```

Rappelez-vous que les caractères graphiques peuvent servir de délimiteurs pour valider la couleur de fond.

!G2 (JEU DE CARACTÈRE G2)

Syntaxe : !G2

Cette instruction effectue un accès au jeu de caractères accentués. Il suffit ensuite d'envoyer le code de l'accent choisi, suivi par le caractère à accentuer.

Cet accès est unique, et doit être renouvelé pour chaque lettre à accentuer.

Il est l'équivalent de l'instruction :

```
10 !SEND,&19 : REM CODE SS2
```

!UPPER (LETTRES MAJUSCULES)

Syntaxe : !UPPER

Cette instruction fait passer le MINITEL en mode majuscule, qui impose l'affichage des caractères alphabétiques en majuscule d'imprimerie sur le clavier du MINITEL.

Les codes renvoyés sur la prise péri-informatique correspondent aussi aux codes ASCII de caractères majuscules.

Pour obtenir un caractère minuscule il faudra appuyer simultanément sur la touche spéciale, sans inscription (SHIFT).

L'équivalent de cette instruction est l'envoi de la séquence dénommée PRO2 - STOP - ENSEIGNEMENT, et correspondant au programme suivant :

```
10 !SEND,&1B  
20 !SEND,&3A  
30 !SEND,&6A  
40 !SEND,&45
```

!LOWER (LETTRES MINUSCULES)

Syntaxe : !LOWER

Cette instruction fait passer le MINITEL en mode minuscule, qui impose l'affichage des caractères alphabétiques en minuscule d'imprimerie sur le clavier du MINITEL.

Les codes renvoyés sur la prise péri-informatique correspondent aussi aux codes ASCII de caractères minuscules.

Pour obtenir un caractère majuscule il faudra appuyer simultanément sur la touche spéciale, sans inscription (SHIFT).

L'équivalent de cette instruction est l'envoi de la séquence dénommée PRO2 - START - ENSEIGNEMENT, et correspondant au programme suivant :

```
10 !SEND,&1B  
20 !SEND,&3A  
30 !SEND,&69 : REM START  
40 !SEND,&45 : REM ENSEIGNEMENT
```

!SOULON (SOULIGNEMENT ACTIF)

Syntaxe : !SOULON

Cette instruction active le mode caractère souligné, mais ne sera validée, comme pour la couleur de fond que par l'apparition d'un délimiteur tel l'espace.

Si vous vous trouvez dans le jeu de caractères graphiques alphamosaïques, les caractères seront disjoints.

L'équivalent de cette instruction est l'envoi d'un caractère d'échappement, suivi du code de lignage :

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&5A : REM DEBUT LIGNAGE
```

!SOULOFF (SOULIGNEMENT NON ACTIF)

Syntaxe : !SOULOFF

Cette instruction désactive le mode caractère souligné.

Si vous vous trouvez en mode caractères graphiques alphamosaïques, vous perdez le mode disjoint au cas où il était validé.

L'équivalent de cette instruction est l'envoi d'un caractère d'échappement suivi du code de fin de lignage :

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&59 : REM FIN DE LIGNAGE
```

!CLIGNOTE (CARACTÈRE CLIGNOTEMENT)

Syntaxe : !CLIGNOTE

Cette instruction active le clignotement des caractères qui la suivent. Elle est exécutée immédiatement.

Son équivalent est l'envoi d'ESCAPE suivi du code de clignotement.

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&48 : REM CLIGNOTEMENT
```

!FIXE (CARACTÈRE FIXE)

Syntaxe : !FIXE

Cette instruction annule le clignotement s'il était actif. Si aucun clignotement n'était validé, elle n'a aucun effet.

Son équivalent est l'envoi d'ESCAPE suivi du code de caractère fixe :

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&49 : REM FIXE
```

!DBL (DOUBLE LARGEUR)

Syntaxe : !DBL

Cette instruction permet d'afficher les caractères qui suivent en double largeur.

Dès que l'instruction !DBL est rencontrée, les emplacements colonnei et colonnei + 1 d'une même rangée sont respectivement affectés des parties gauche et droite des caractères.

Si un nouvel affichage d'un caractère en double taille est effectué en colonnei + 1 de la ligne, la partie droite du caractère sera affichée sur cette colonne, la partie gauche étant reléguée en colonnei + 2.

Remarque :

Cette instruction est utilisable en ligne zéro.

Les caractères semi-graphiques ne sont pas affectés par cette instruction.

Dans le programme, cette instruction est équivalente à la succession d'un code ESCAPE suivi du code de double largeur :

```
10 !SEND,&1B : REM ESCAPE
20 !SEND,&4E : REM DOUBLE LARGEUR
```

!DBH (DOUBLE HAUTEUR)

Syntaxe : !DBH

Cette instruction permet d'afficher les caractères qui suivent en double hauteur.

Dès que l'instruction !DBH est rencontrée, les emplacements lignei et lignei - 1 d'une même colonne sont respectivement affectés des parties basse et haute des caractères, jusque la fin de la ligne.

Si un nouvel affichage est effectué en double hauteur sur la lignei - 1, la partie haute sera affichée sur la lignei, et la partie basse sur la lignei - 1.

Remarque :

Il n'est pas possible d'utiliser cette instruction en lignes zéro et un, elle serait ignorée.

Les caractères semi-graphiques ne sont pas affectés par cette instruction.

Equivalent dans le programme :

```
10 !SEND,&1B : REM ESCAPE
20 !SEND,&4D : REM DOUBLE HAUTEUR
```

!DBG (DOUBLE GRANDEUR)

Syntaxe : !DBG

Cette instruction permet d'afficher les caractères qui suivent en double grandeur.

Dès que l'instruction !DBG est rencontrée, les emplacements lignei et lignei - 1 d'une même colonne sont respectivement affectées des parties basses et hautes des caractères, jusqu'à la fin de la ligne.

Si un nouvel affichage est effectué en double grandeur sur la lignei - 1, la partie haute sera affichée sur la lignei, et la partie basse sur la lignei - 1.

Remarque :

Il n'est pas possible d'obtenir un affichage en quinconce.

Il n'est pas possible d'utiliser cette instruction en lignes zéro et un, elle serait ignorée.

Les caractères semi-graphiques ne sont pas affectés par cette instruction.

Equivalent dans le programme :

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&4F : REM DOUBLE GRANDEUR
```

!NORMH (HAUTEUR NORMALE)

Syntaxe : !NORMH

Cette instruction permet de redéfinir les caractères en dimension normale, tels qu'ils le sont à l'initialisation.

Equivalent, dans le programme :

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&4C : REM DOUBLE GRANDEUR
```

!BS (BACK SPACE)

Syntaxe : !BS

Cette instruction effectue un déplacement du curseur d'un caractère en taille normale vers la gauche en émettant sur la ligne le code &08 (!SEND,&08).

Si le curseur est initialement en colonne 1, il se déplace en colonne 40 de la ligne précédente.

S'il se trouve en ligne 1 - colonne 1, il passe en ligne 24 - colonne 40, si le mode scrolling n'est pas actif, sinon il fait descendre l'écran d'un caractère vers le bas.

!HT (HORIZONTAL TABULATION)

Syntaxe : !HT

Cette instruction effectue un déplacement du curseur d'un caractère en taille normale vers la droite en émettant sur la ligne le code &09 (!SEND,&09).

Si le curseur est initialement en colonne 40, il se déplace en colonne 1 de la ligne suivante.

S'il se trouve en ligne 24 - colonne 40, il passe en ligne 1 - colonne 1, si le mode scrolling n'est pas actif, sinon il fait monter l'écran d'un caractère vers le haut.

!LF (LINE FEED)

Syntaxe : !LF

Cette instruction effectue un déplacement du curseur d'un caractère en taille normale vers le bas en émettant sur la ligne le code &0A (!SEND,&0A).

S'il se trouve en ligne 24, il passe en ligne 1, si le mode scrolling n'est pas actif, sinon il fait monter l'écran d'un caractère vers le haut.

L'utilisation de cette instruction en ligne zéro, restitue la position du curseur avant qu'il ne soit positionné sur la ligne supérieure.

!VT (VERTICAL TABULATION)

Syntaxe : !VT

Cette instruction effectue un déplacement du curseur d'un caractère en taille normale vers le haut en émettant sur la ligne le code &0B (!SEND,&0B).

S'il se trouve en ligne 1, il passe en ligne 24, si le mode scrolling n'est pas actif, sinon il fait descendre l'écran d'un caractère vers le bas.

!HOME (POSITION INITIALE)

Syntaxe : !HOME

Cette instruction positionne le curseur en ligne 1 colonne 1, sans effacer l'écran.

Son équivalent pourrait être :

10 !LOCATE,1,1

mais elle a l'avantage d'être plus rapide car elle utilise le code de RS: &1E (*Record Separator*), qu'elle envoie uniquement sur la ligne série.

!CANCEL (REPLISSAGE)

Syntaxe : !CANCEL

Cette instruction remplit les emplacements depuis la position du curseur jusque la fin de la ligne avec des espaces.

L'exécution est plus rapide que le remplissage par l'envoi d'espaces, et elle permet de ne pas connaître le nombre d'espaces à émettre.

Son équivalent est l'envoi du code CAN (&18) :

```
10 !SEND,&18
```

!BELL (CLOCHE)

Syntaxe : !BELL

Emet un BIP sur le MINITEL (!SEND,&07)

!ROULEAU (SCROLLING)

Syntaxe : !ROULEAU

L'utilisation de cette instruction provoque la mise en œuvre du mode scrolling de l'écran situé entre les lignes 1 et 24 du MINITEL (la ligne zéro n'étant pas affectée).

Tout caractère apparaissant après la ligne 24 - colonne 40 provoquera une remontée de l'écran (l'ex-ligne1 disparaissant), libérant ainsi une nouvelle ligne.

Au cas où il a été sélectionné le mode double hauteur, ou double grandeur, l'écran remonte de deux lignes.

Le programme gérant cette instruction utilise la séquence de codes définis par la commande PRO2 - START - &43., soit l'équivalent :

```
10 !SEND,&1B  
20 !SEND,&3A  
30 !SEND,&69 : REM START  
40 !SEND,&43 : REM ROULEAU
```

!PAGE

Syntaxe : !PAGE

Cette instruction positionne la partie d'écran située entre les lignes 1 et 24 en mode page, c'est-à-dire que le mode rouleau est inhibé.

Tout caractère apparaissant après la ligne 24 - colonne 40 s'affichera en ligne 1 - colonne 1.

Si les caractères possèdent l'attribut double hauteur ou double taille, le

caractère suivant la ligne 24 ou 23 - colonne 40 est positionné en ligne 2 - colonne 1.

La gestion est effectuée dans le programme par la commande PRO2 - STOP - &43.

```
10 !SEND,&1B
20 !SEND,&3A
30 !SEND,&6A : REM STOP
40 !SEND,&43 : REM ROULEAU
```

!REPEAT

Syntaxe : !**REPEAT**,nombreentier
 !**REPEAT**,variable%

Cette instruction permet de répéter un certain nombre de fois le dernier caractère émis (entre 1 et 63).

Exemple :

```
10 !WRITE,« ABCDE »
20 !REPEAT,10
```

affichera la chaîne de caractère suivante : ABCDEEEEEEEEEEEEE, le caractère E apparaissant $1 + 10 = 11$ fois.

Il est possible de cumuler les instructions !REPEAT afin de répéter plus de 63 fois un caractère.

Dans le programme, le nombre fourni est augmenté de &40, et est précédé du code de REP: &12. Son équivalent serait, pour le programme précédent :

```
10 !WRITE,« ABCDE »
20 !SEND,&12
30 !SEND,&40 + 10
```

!MASKON (MASQUAGE)

Syntaxe : !**MASKON**

Cette instruction permet de ne pas afficher une partie de texte, lorsqu'elle est validée par le délimiteur espace.

Equivalent, dans le programme Assembleur :

```
10 !SEND,&1B : REM ESCAPE
20 !SEND,&58 : REM MASQUAGE
```

!MASKOFF (DÉMASQUAGE)Syntaxe : **!MASKOFF**

Cette fonction dévalide le mode masquage, et fait réapparaître instantanément un texte éventuellement masqué, lorsqu'elle est utilisée sur le même délimiteur.

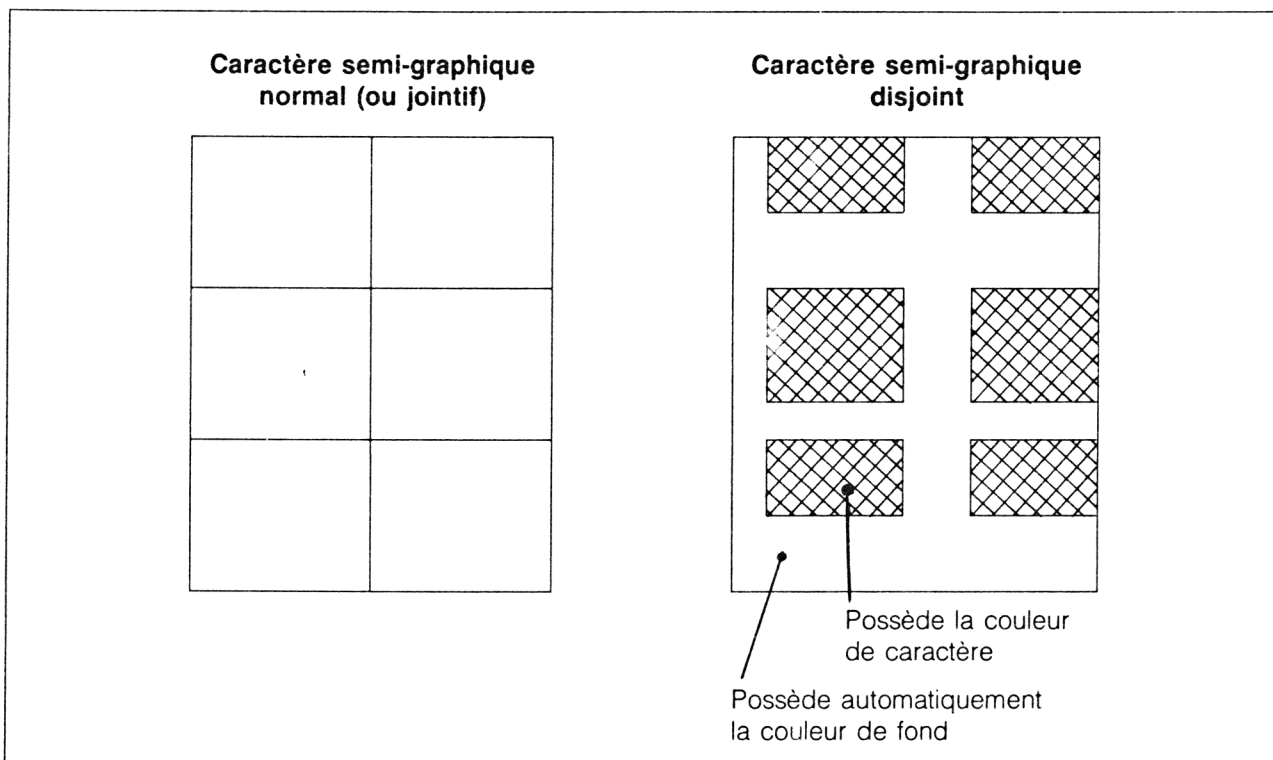
Équivalent, dans le programme Assembleur :

```
10 !SEND,&1B : REM ESCAPE  
20 !SEND,&5F : REM DEMASQUAGE
```

!DISJ (CARACTÈRES GRAPHIQUES DISJOINTS)Syntaxe : **!DISJ**

Cette instruction valide le mode de caractères disjoints lorsque l'on utilise le jeu G1.

Vous trouverez en figure 2 la différence entre les caractères semi-graphiques normaux, et les caractères disjoints.



L'équivalent de cette instruction est l'envoi d'un caractère d'échappement, suivi du code de lignage :

10 SEND,&1B : REM ESCAPE
20 SEND,&5A : REM DEBUT LIGNAGE

Afin de créer manuellement vos graphismes, nous vous proposons une grille préliminaire, avant toute programmation en figure 3.

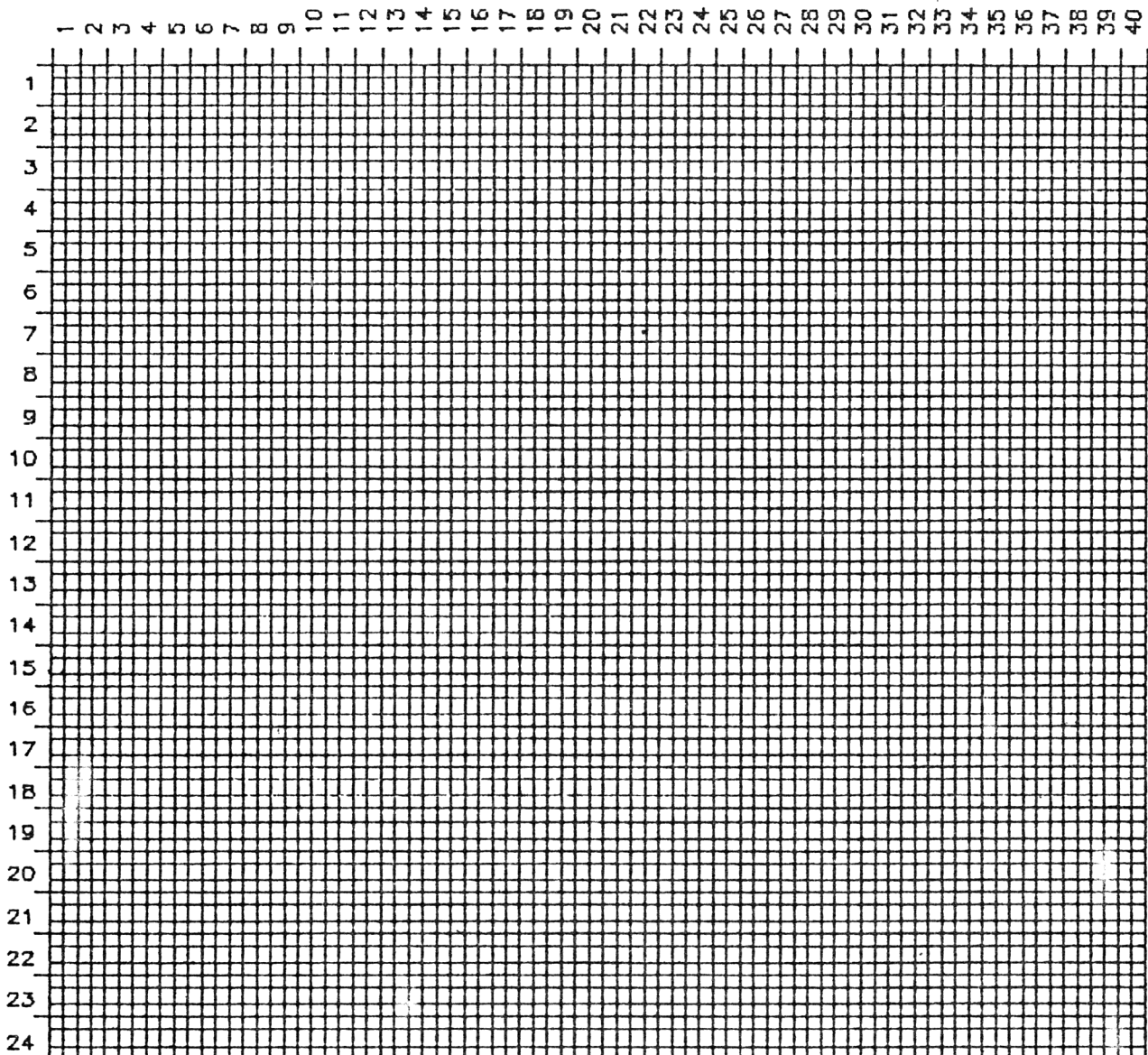


Fig. 3

II. Le programme Assembleur

Nous vous proposons dans les pages suivantes le programme Assembleur permettant la création des nouvelles fonctions décrites plus haut.

```

8000'
TITLE MINIT3.MAC
SUBTTL gestion minitel par RSXs
;*****
;* TRANSMISSION CPC VERS *
;* MINITEL PAR LIAISON *
;* CENTRONICS DU PORT *
;* IMPRIMANTE *
;* AVEC UN JEU D'INSTRUC- *
;* SOUS FORME DE RSXs *
;* POUR GERER FACILEMENT *
;* LE MINITEL *
;*****
;
;
;*** TABLE DES EQUIVALENCES ***
;
;PORT DU PIO POUR LIGNE BUSY
F500 FORBUS EQU 0F500H
;
;PORT IMPRIMANTE POUR LIGNE DONNER
EF00 PORIMP EQU 0EF00H
;
;POUR TEMPORISATION 833 ms
006D TEMP8 EQU 06DH
;
;POUR TEMPORISATION 833/2 ms
0036 TEMP4 EQU 036H
;
;VECTEUR KL-LOG-TEXT
BCD1 KLOGTX EQU 0BCD1H
;
;VECTEUR AFFICHAGE CPC
BB5A AFFICH EQU 0BB5AH
;
;*****
;
;
;*** ORIGINE D'ASSEMBLAGE ***
;
ORG 09900H
;
;*** ORIGINE DE CHARGEMENT ***
;
;
;
;*****
;* INSTALLATION DES *
;* DIFFERENTES RSXs *
;* POUR GERER LE *
;* MINITEL *
;*****
;
DEBUT:
9900' 21 9A55' LD HL,KERNEL ;4 OCTETS
;RESERVES AU SYSTEME
9903' 01 990F' LD BC, VECTEU ;VECTEUR

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

;POUR INSTAALTION DES INSTRUCTIONS
9906'   CD BCD1      CALL KLOGTX ;CREER INSTRUCTIONS
9909'   3E C9       LD A,0C9H ;RET POUR EVITER
990B'   32 9900'    LD (DEBUT),A ;TOUT NOUVEL APPEL
990E'   C9         RET ;FIN INSTALLATION
;
;
990F'   998F'      VECTEU: DEFW TABLE;VECTEUR
;
;*** TABLE DES SAUTS AUX ***
;*** NOUVELLES INSTRUCTIONS ***
;
9911'   C3 9C4B'    JP SEND ;ENVOI CODE
9914'   C3 9C5B'    JP SENCAR ;ENVOI CARACTERE
9917'   C3 9C76'    JP SENSEQ ;ENVOI SEQUENCE
991A'   C3 9C8F'    JP PRINT ;AFFICHAGE LIGNE
991D'   C3 9C9D'    JP RECEPT ;RECEPTION CODE
9920'   C3 9CB2'    JP RECAR ;RECEPTION LIGNE
9923'   C3 9AA2'    JP LOCATE ;POSITIONNEMENT
9926'   C3 9AE8'    JP CURSON ;CURSEUR VISUALISE
9929'   C3 9AKE'    JP CUROFF ;CURSEUR ETEINT
992C'   C3 9AF3'    JP LOWVID ;INVERSION VIDEO
992F'   C3 9AFE'    JP NORVID ;VIDEO NORMALE
9932'   C3 9BF7'    JP INK ;COULEUR CARACTERE
9935'   C3 9C21'    JP PAPER ;COULEUR FOND
9938'   C3 9A82'    JP CLS ;EFFACE ECRAN
993B'   C3 9A88'    JP ALLCLS ;EFFACE TOUT
993E'   C3 9B09'    JP G0 ;ACCES JEU G0
9941'   C3 9B0F'    JP G1 ;ACCES JEU G1
9944'   C3 9B15'    JP G2 ;ACCES JEU G2
9947'   C3 9B1B'    JP UPPER ;FORCAGE MAJUSCULE
994A'   C3 9B30'    JP LOWER ;FORCAGE MINUSCULE
994D'   C3 9B45'    JP SOULON ;SOULIGNEMENT
9950'   C3 9B50'    JP SOULOF ;FIN DE SOULIGNEMENT
9953'   C3 9B5B'    JP CLIGNO ;CLIGNOTEMENT
9956'   C3 9B66'    JP FIXE ;CARACTERE FIXE
9959'   C3 9B71'    JP DBL ;DOUBLE LARGEUR
995C'   C3 9B7C'    JP DBH ;DOUBLE HAUTEUR
995F'   C3 9B87'    JP DBG ;DOUBLE GRANDEUR
9962'   C3 9B92'    JP NORMH ;HAUTEUR NORMALE
9965'   C3 9B9D'    JP BS ;BACKSPACE
9968'   C3 9BA3'    JP HT ;HORIZONTAL TAB
996B'   C3 9BA9'    JP LF ;LINE FEED
996E'   C3 9BAF'    JP VT ;VERTICAL TAB
9971'   C3 9BB5'    JP RC ;CARRIAGE RETURN
9974'   C3 9BBB'    JP HOME ;POSITION HAUT GAUCHE
9977'   C3 9BC1'    JP CANCKL ;EFFACE DROITE LIGNE
997A'   C3 9BC7'    JP BELL ;CLOCHE
997D'   C3 9BCD'    JP ROULE ;MODE ENSEIGNEMENT
9980'   C3 9BE2'    JP PAGE ;MODE PAGE FIXE
9983'   C3 9A59'    JP REPEAT ;REPETITION
9986'   C3 9A6C'    JP MASKON ;MASQUAGE
9989'   C3 9A77'    JP MASKOF ;DEMASCAGE
998C'   C3 9B45'    JP DISJ ;JEU G1 DISJOINT
;
;
;*** TABLE DES INSTRUCTION ***
;
TABLE:

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

998F'	53 45 4E	DEFB "SEN"
9992'	C4	DEFB "D"+080H
9993'	57 52 49 54	DEFB "WRIT"
9997'	C5	DEFB "E"+080H
9998'	53 45 4E 44	DEFB "SENDSK"
999C'	53 45	
999E'	D1	DEFB "Q"+080H
999F'	50 52 49 4E	DEFB "PRIN"
99A3'	D4	DEFB "T"+080H
99A4'	52 45 43 45	DEFB "RECEP"
99A8'	50	
99A9'	D4	DEFB "T"+080H
99AA'	52 45 43 41	DEFB "RECA"
99AE'	D2	DEFB "R"+080H
99AF'	4C 4F 43 41	DEFB "LOCAT"
99B3'	54	
99B4'	C5	DEFB "E"+080H
99B5'	43 55 52 53	DEFB "CURSO"
99B9'	4F	
99BA'	CE	DEFB "N"+080H
99BB'	43 55 52 53	DEFB "CURSOF"
99BF'	4F 46	
99C1'	C6	DEFB "F"+080H
99C2'	4C 4F 57 56	DEFB "LOWVIDE"
99C6'	49 44 45	
99C9'	CF	DEFB "O"+080H
99CA'	4E 4F 52 4D	DEFB "NORMVIDE"
99CE'	56 49 44 45	
99D2'	CF	DEFB "O"+080H
99D3'	49 4E	DEFB "IN"
99D5'	CB	DEFB "K"+080H
99D6'	50 41 50 45	DEFB "PAPE"
99DA'	D2	DEFB "R"+080H
99DB'	43 4C	DEFB "CL"
99DD'	D3	DEFB "S"+080H
99DE'	41 4C 4C 43	DEFB "ALLCL"
99E2'	4C	
99E3'	D3	DEFB "S"+080H
99E4'	47	DEFB "G"
99E5'	B0	DEFB "0"+080H
99E6'	47	DEFB "G"
99E7'	B1	DEFB "1"+080H
99E8'	47	DEFB "G"
99E9'	B2	DEFB "2"+080H
99EA'	55 50 50 45	DEFB "UPPE"
99EE'	D2	DEFB "R"+080H
99EF'	4C 4F 57 45	DEFB "LOWE"
99F3'	D2	DEFB "R"+080H
99F4'	53 4F 55 4C	DEFB "SOULO"
99F8'	4F	
99F9'	CE	DEFB "N"+080H
99FA'	53 4F 55 4C	DEFB "SOULOF"
99FE'	4F 46	
9A00'	C6	DEFB "F"+080H
9A01'	43 4C 49 47	DEFB "CLIGNOT"
9A05'	4E 4F 54	
9A08'	C5	DEFB "E"+080H
9A09'	46 49 58	DEFB "FIX"
9A0C'	C5	DEFB "E"+080H

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9A0D' 44 42          DEFB "DB"
9A0F' CC            DEFB "L"+080H
9A10' 44 42          DEFB "DB"
9A12' C8            DEFB "H"+080H
9A13' 44 42          DEFB "DB"
9A15' C7            DEFB "G"+080H
9A16' 4E 4F 52 4D    DEFB "NORM"
9A1A' C8            DEFB "H"+080H
9A1B' 42            DEFB "B"
9A1C' D3            DEFB "S"+080H
9A1D' 48            DEFB "H"
9A1E' D4            DEFB "T"+080H
9A1F' 4C            DEFB "L"
9A20' C6            DEFB "F"+080H
9A21' 56            DEFB "V"
9A22' D4            DEFB "T"+080H
9A23' 43            DEFB "C"
9A24' D2            DEFB "R"+080H
9A25' 48 4F 4D       DEFB "HOM"
9A28' C5            DEFB "E"+080H
9A29' 43 41 4E 43    DEFB "CANCE"
9A2D' 45
9A2E' CC            DEFB "L"+080H
9A2F' 42 45 4C       DEFB "BKL"
9A32' CC            DEFB "L"+080H
9A33' 52 4F 55 4C    DEFB "ROULEA"
9A37' 45 41
9A39' D5            DEFB "U"+080H
9A3A' 50 41 47       DEFB "PAG"
9A3D' C5            DEFB "E"+080H
9A3E' 52 45 50 45    DEFB "REPEA"
9A42' 41
9A43' D4            DEFB "T"+080H
9A44' 4D 41 53 4B    DEFB "MASKO"
9A48' 4F
9A49' CE            DEFB "N"+080H
9A4A' 4D 41 53 4B    DEFB "MASKOF"
9A4E' 4F 46
9A50' C6            DEFB "F"+080H
9A51' 44 49 53       DEFB "DIS"
9A54' CA            DEFB "J"+080H
;
;
;*** KERNEL POUR SYSTEME ***
9A55' 30 30 30 30    KERNEL: DEFB "0000"
;
;
;*****
;* REPETITION D'UN CODE *
;*****
;
REPEAT:
9A59' FE 01          CP 01H ;UN CARACTERE
9A5B' C0             RET NZ ;NON ALORS FIN
9A5C' DD 7E 00       LD A,(IX+00H) ;NOMBRE DE
;REPETITIONS
9A5F' F5             PUSH AF
9A60' 3E 12          LD A,012H ;CODE REPETITION
9A62' CD 9C51'       CALL EMASC

```


Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9A65'   F1           POP AF
9A66'   C6 40       ADD A,040H ;POUR CORRECTION
                          ;SUR SIX BITS
9A68'   CD 9C51'   CALL EMASC.
9A6B'   C9         RET
                          ;
                          ;
                          ;*****
                          ;* MASQUAGE DE CARACTERES *
                          ;*****
                          ;
MASKON:
9A6C'   3E 1B       LD A,01BH ;GRILLE C1
9A6E'   CD 9C51'   CALL EMASC
9A71'   3E 58       LD A,058H ;MASQUAGE
9A73'   CD 9C51'   CALL EMASC
9A76'   C9         RET
                          ;
                          ;
                          ;*****
                          ;*   DEMASQUAGE   *
                          ;*****
                          ;
MASKOF:
9A77'   3E 1B       LD A,01BH ;GRILLE C1
9A79'   CD 9C51'   CALL EMASC
9A7C'   3E 5F       LD A,05FH ;DEMASQUAGE
9A7E'   CD 9C51'   CALL EMASC
9A81'   C9         RET
                          ;
                          ;
                          ;*****
                          ;* EFFACEMENT ECRAN *
                          ;*****
                          ;
CLS:
9A82'   3E 0C       LD A,0CH ;CODE CLS
9A84'   CD 9C51'   CALL EMASC
9A87'   C9         RET
                          ;
                          ;
                          ;*****
                          ;
                          ;*****
                          ;* EFFACE TOUT MINITEL *
                          ;*****
                          ;
ALLCLS:
9A88'   3E 1F       LD A,01FH ;LOCALISATION
9A8A'   CD 9C51'   CALL EMASC
9A8D'   3E 40       LD A,040H ;EN LIGNE 0
9A8F'   CD 9C51'   CALL EMASC
9A92'   3E 41       LD A,041H ;COLONNE 1
9A94'   CD 9C51'   CALL EMASC
9A97'   3E 18       LD A,018H ;ET CANCEL
9A99'   CD 9C51'   CALL EMASC
9A9C'   3E 0C       LD A,00CH ;CLS ECRAN
9A9E'   CD 9C51'   CALL EMASC
9AA1'   C9         RET

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

;
;*****
;
;
;*****
;* LOCALISATION SUR *
;* L'ECRAN DU MINITEL *
;*****
;
LOCATE:
9AA2' DD 6E 00 LD L,(IX+00H);POSITION
9AA5' DD 66 01 LD H,(IX+01H);COLONNE
9AA8' 7C LD A,H
9AA9' FE 00 CP 00;NOMBRE TROP GRAND
9AAB' 20 2E JR NZ,ERRLOC;ALORS ERREUR
9AAD' 7D LD A,L
9AAE' FE 00 CP 00H;COLONNE 0?
9AB0' 28 29 JR Z,ERRLOC;IMPOSSIBLE
9AB2' FE 28 CP 028H;DERNIERE COLONNE?
9AB4' 30 25 JR NC,ERRLOC;SI PLUS LOIN
;IMPOSSIBLE,ALORS ERREUR
9AB6' F5 PUSH AF;SAUVER COLONNE
9AB7' DD 6E 02 LD L,(IX+02H);RECUPERE
9ABA' DD 66 03 LD H,(IX+03H);LIGNE
9ABD' 7C LD A,H;LIGNE
9ABE' FE 00 CP 00;TROP GRANDE?
9AC0' 20 18 JR NZ,ERRLO2;ALORS ERREUR
9AC2' 7D LD A,L
9AC3' FE 19 CP 019H;PLUS GRAND QUE
9AC5' 30 13 JR NC,ERRLO2;DERNIERE LIGNE
;ALORS ERREUR
9AC7' F5 PUSH AF;SAUVER LIGNE
9AC8' 3E 1F LD A,01FH;CODE LOCALISATION
9ACA' CD 9C51' CALL EMASC
9ACD' F1 POP AF;LIGNE
9ACE' C6 40 ADD A,040H;CORRECTION
9AD0' CD 9C51' CALL EMASC
9AD3' F1 POP AF;COLONNE
9AD4' C6 40 ADD A,040H;CORRECTION
9AD6' CD 9C51' CALL EMASC
9AD9' C9 RET
;
;*** ERREUR DELOCALISATION ***
;
ERRLO2:
9ADA' F1 POP AF;PURGER FILE
;
ERRLOC:
9ADB' 21 9DF3' LD HL,MESLOC;ZONE MESSAGE
ERRLO3:
9ADE' 7E LD A,(HL)
9ADF' FE 00 CP 00;DERNIERE LETTRE?
9AE1' C8 RET Z
9AE2' CD BB5A CALL AFFICH
9AE5' 23 INC HL;POSITION SUIVANTE
9AE6' 18 F6 JR ERRLO3
;
;*****
;

```


Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9B09'   3E 0F           LD A,0FH ;CODE SI
9B0B'   CD 9C51'       CALL EMASC
9B0E'   C9             RET
;
;*****
;
;
;*****
;* ACCES AU JEU  G1      *
;*****
;
G1:
9B0F'   3E 0E           LD A,0EH ;CODE SO
9B11'   CD 9C51'       CALL EMASC
9B14'   C9             RET
;
;*****
;
;
;*****
;* ACCES AU JEU  G2      *
;*****
;
G2:
9B15'   3E 19           LD A,019H ;CODE SS2
9B17'   CD 9C51'       CALL EMASC
9B1A'   C9             RET
;
;*****
;
;
;*****
;* FORCAGE EN MAJUSCULES *
;*****
;
UPPER:
9B1B'   3E 1B           LD A,01BH ;ESC
9B1D'   CD 9C51'       CALL EMASC
9B20'   3E 3A           LD A,03AH ;PRO3
9B22'   CD 9C51'       CALL EMASC
9B25'   3E 6A           LD A,06AH ;DEACTIVE
9B27'   CD 9C51'       CALL EMASC
9B2A'   3E 45           LD A,045H ;MINUSCULE
9B2C'   CD 9C51'       CALL EMASC
9B2F'   C9             RET
;
;*****
;
;
;*****
;* FORCAGE EN MINUSCULES *
;*****
;
LOWER:
9B30'   3E 1B           LD A,01BH ;ESC
9B32'   CD 9C51'       CALL EMASC
9B35'   3E 3A           LD A,03AH ;PRO3
9B37'   CD 9C51'       CALL EMASC
9B3A'   3E 69           LD A,069H ;ACTIVE

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9B3C'   CD 9C51'   CALL EMASC
9B3F'   3E 45      LD A,045H ;MINUSCULES
9B41'   CD 9C51'   CALL EMASC
9B44'   C9         RET
;
;*****
;
;
;*****
;*      SOULIGNEMENT      *
;*****
;
DISJ:
SOULON:
9B45'   3E 1B      LD A,01BH ;JEU C1
9B47'   CD 9C51'   CALL EMASC
9B4A'   3E 5A      LD A,05AH ;SOULIGNE
9B4C'   CD 9C51'   CALL EMASC
9B4F'   C9         RET
;
;*****
;
;
;*****
;*  FIN DE SOULIGNEMENT  *
;*****
;
SOULOF:
9B50'   3E 1B      LD A,01BH ;JEU C1
9B52'   CD 9C51'   CALL EMASC
9B55'   3E 59      LD A,059H ;FIN SOULIGNE
9B57'   CD 9C51'   CALL EMASC
9B5A'   C9         RET
;
;*****
;
;
;*****
;*      CLIGNOTEMENT      *
;*****
;
CLIGNO:
9B5B'   3E 1B      LD A,01BH ;JEU C1
9B5D'   CD 9C51'   CALL EMASC
9B60'   3E 48      LD A,048H ;CLIGNOTE
9B62'   CD 9C51'   CALL EMASC
9B65'   C9         RET
;
;*****
;
;
;*****
;*  CARACTERES FIXES      *
;*****
;
FIXE:
9B66'   3E 1B      LD A,01BH ;JEU C1
9B68'   CD 9C51'   CALL EMASC
9B6B'   3E 49      LD A,049H ;FIXE

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9B6D'   CD 9C51'   CALL EMASC
9B70'   C9         RET
;
;*****
;
;
;*****
;*   DOUBLE   LARGEUR   *
;*****
;
DBL:
LD A,01BH ;JEU C1
CALL EMASC
LD A,04EH ;DBLE LARGEUR
CALL EMASC
RET
;
;*****
;
;
;*****
;*   DOUBLE HAUTEUR   *
;*****
;
DBH:
LD A,01BH ;JEU C1
CALL EMASC
LD A,04DH ;DBLE HAUTEUR
CALL EMASC
RET
;
;*****
;
;
;*****
;*   DOUBLE GRANDEUR   *
;*****
;
DBG:
LD A,01BH ;JEU C1
CALL EMASC
LD A,04FH ;DBLE TAILLE
CALL EMASC
RET
;
;*****
;
;
;*****
;*   HAUTEUR   NORMALE   *
;*****
;
NORMH:
LD A,01BH ;JEU C1
CALL EMASC
LD A,04CH ;NORMAL
CALL EMASC
RET
;

```


Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

;
;*****
;* RETOUR INITIALE *
;*****
;
HOME:
LD A,01EH
CALL EMASC
RET
;
;*****
;
;*****
;* CANCEL *
;*****
;
CANCEL:
LD A,018H
CALL EMASC
RET
;
;*****
;
;*****
;* UN SON DE CLOCHE *
;*****
;
BELL:
LD A,07H
CALL EMASC
RET
;
;*****
;
;*****
;* MODE ROULEAU *
;*****
;
ROULE:
LD A,01BH ;ESC
CALL EMASC
LD A,03AH ;PRO3
CALL EMASC
LD A,069H ;ACTIVE
CALL EMASC
LD A,043H ;ROULEAU
CALL EMASC
RET
;
;*****
;
;*****
;* MODE PAGE *
;*****
;

```

9BBB'	3E 1E	
9BBD'	CD 9C51'	
9BC0'	C9	
9BC1'	3E 18	
9BC3'	CD 9C51'	
9BC6'	C9	
9BC7'	3E 07	
9BC9'	CD 9C51'	
9BCC'	C9	
9BCD'	3E 1B	
9BCF'	CD 9C51'	
9BD2'	3E 3A	
9BD4'	CD 9C51'	
9BD7'	3E 69	
9BD9'	CD 9C51'	
9BDC'	3E 43	
9BDE'	CD 9C51'	
9BE1'	C9	


```

PAGE:
9BE2' 3E 1B LD A,01BH ;ESC
9BE4' CD 9C51' CALL EMASC
9BE7' 3E 3A LD A,03AH ;PRO3
9BE9' CD 9C51' CALL EMASC
9BEC' 3E 6A LD A,06AH ;DEACTIVE
9BEE' CD 9C51' CALL EMASC
9BF1' 3E 43 LD A,043H ;ROULEAU
9BF3' CD 9C51' CALL EMASC
9BF6' C9 RET
;
;
;*****
;* MODIFICATION DE LA *
;* COULEUR D'ENCRE DU *
;* TEXTE AFFICHABLE *
;*****
;
INK:
9BF7' DD 66 01 LD H,(IX+01H)
9BFA' DD 6E 00 LD L,(IX+00H)
9BFD' 7C LD A,H
9BFE' FE 00 CP 00H;TROP GRAND?
9C00' 20 12 JR NZ,ERRINK;ALORS ERREUR
9C02' 7D LD A,L
9C03' FE 08 CP 08;8 ENCRE MAXIMUM
9C05' 30 0D JR NC,ERRINK;SINON ERREUR
9C07' F5 PUSH AF;SAUVER PROVISoireMENT
9C08' 3E 1B LD A,01BH;ESCAPE
9C0A' CD 9C51' CALL EMASC
9C0D' F1 POP AF;RECUPERE ENCRE
9C0E' C6 40 ADD A,040H;CORRECTION ENCRE
9C10' CD 9C51' CALL EMASC
9C13' C9 RET
;
;*** ERREUR ENCRE ***
;
ERRINK:
9C14' 21 9E0D' LD HL,MESINK;MESSAGE ERREUR ENCRE
ERRIN1:
9C17' 7E LD A,(HL)
9C18' FE 00 CP 00H;FIN?
9C1A' C8 RET Z
9C1B' CD BB5A CALL AFFICH
9C1E' 23 INC HL
9C1F' 18 F6 JR ERRIN1
;
;*****
;
;*****
;* MODIFICATION DE LA COULEUR *
;* DU FOND (COULEUR PAPIER ) *
;*****
;
PAPER:
9C21' DD 66 01 LD H,(IX+01H)
9C24' DD 6E 00 LD L,(IX+00H)
9C27' 7C LD A,H

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9C28'   FR 00           CP 00H
9C2A'   20 12         JR NZ,ERRPAP
9C2C'   7D            LD A,L
9C2D'   FE 08         CP 08;8 ENCREs POSSIBLEs
9C2F'   30 0D         JR NC,ERRPAP;SINON ERREUR
9C31'   F5           PUSH AF;SAUVE ENCRE
9C32'   3E 1B         LD A,01BH;ESCAPE
9C34'   CD 9C51'     CALL EMASC
9C37'   F1           POP AF;RECUPERE ENCRE
9C38'   C6 50         ADD A,050H;CORRECTION
                          ;DIFFERENTE POUR LE FOND
9C3A'   CD 9C51'     CALL EMASC
9C3D'   C9           RET
                          ;
                          ;*** MESSAGE ERREUR FOND ***
                          ;
9C3E'   21 9E2D'     ERRPAP:
                          LD HL,MESPAP;ZONE MESSAGE
9C41'   7E           ERRPA1:
                          LD A,(HL)
9C42'   FE 00         CP 00
9C44'   C8           RET Z
9C45'   CD BB5A      CALL AFFICH
9C48'   23           INC HL
9C49'   18 F6        JR ERRPA1
                          ;
                          ;*****
                          ;
                          ;*****
                          ;* EMISSION D'UN CODE      *
                          ;* ASCII CONTENU DANS      *
                          ;* UNE VARIABLE NUMERIQUE *
                          ;* NUMERIQUE          *
                          ;*****
                          ;
9C4B'   FE 01         SEND:
9C4D'   C0           CP 01;UN CARACTERE TRANSMIS?
                          RET NZ;SINON RETOUR
                          ;
9C4E'   DD 7E 00     LD A,(IX+00H);RECUPERE CARACTERE
                          ;
                          EMASC:
                          ;
9C51'   CD 9DAF'     CALL PARITE;CALCUL PARITE
9C54'   CD 9DB7'     CALL TRAME;FORMATION TRAME
9C57'   CD 9DCD'     CALL EMETTRE;EMISSION TRAME
9C5A'   C9           RET;FIN
                          ;
                          ;*****
                          ;
                          ;*****
                          ;* EMISSION D'UNE          *
                          ;* CHAINE DE CARACTERES      *
                          ;* CONTENUE DANS UNE          *
                          ;* VARIABLE ALPHANUMERIQUE *
                          ;* OU EXPLICITE            *
                          ;*****

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

;
SENCAR:
9C5B' FE 01 CP 01H;UNE CHAINE?
9C5D' C0 RET NZ;SINON RETOUR
;
9C5E' DD 6E 00 LD L,(IX+00H);RECUPERE
9C61' DD 66 01 LD H,(IX+01H);ADRESSE POINTEUR
9C64' 46 LD B,(HL);RECUPERE LE NOMBRE
;DE CARACTERES DE LA CHAINE
9C65' 23 INC HL;ADRESSE SUIVANTE
9C66' 5E LD E,(HL);OCTET BAS
9C67' 23 INC HL;ADRESSE SUIVANTE
9C68' 56 LD D,(HL);OCTET HAUT
;DE L'ADRESSE DE LA CHAINE
9C69' EB EX DE,HL;DANS HL POUR
;SERVIR DE POINTEUR
;
EMCA1:
9C6A' 7E LD A,(HL);UN CARACTERE
9C6B' E5 PUSH HL;SAUVE HL
9C6C' C5 PUSH BC;ET BC
9C6D' CD 9C51' CALL EMASC;EMETTRE LE CARACTERE
9C70' C1 POP BC;RECUPERER
9C71' E1 POP HL;LES REGISTRES
9C72' 23 INC HL;ADRESSE SUIVANTE DE
;LA CHAINE DE CARACTERES
9C73' 10 F5 DJNZ EMCA1;ENCORE UN, ALORS
;RETOURNER POUR LE CARACTERE
;SUIVANT
9C75' C9 RET;SINON FIN
;
;*****
;
;*****
;* ENVOI D'UNE SEQUENCE *
;* VERS LE MINITEL *
;*****
;
SENSEQ:
9C76' FE 00 CP 00H;PAS DE CARACTERE?
9C78' C8 RET Z;ALORS RETOUR
9C79' 47 LD B,A;NOMBRE DE CODES
9C7A' 57 LD D,A;POUR RESERVIR
;
SENSE1:
9C7B' DD 7E 00 LD A,(IX+00H)
9C7E' F5 PUSH AF
9C7F' DD 23 INC IX;CODE
9C81' DD 23 INC IX;SUIVANT
9C83' 10 F6 DJNZ SENSE1;ENCORE UN CODE?
;
9C85' 42 LD B,D;NOMBRE DE CARACTERES
SENSE2:
9C86' F1 POP AF;RECUPERE UN CODE
9C87' C5 PUSH BC;SAUVE BC
9C88' CD 9C51' CALL EMASC;A EMETTRE
9C8B' C1 POP BC;RECUPERE NB DE CODES
9C8C' 10 F8 DJNZ SENSE2

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9C8E'   C9           RET;FIN
;
;*****
;
;
;*****
;* ENVOI D'UNE LIGNE PUIS *
;*   PASSAGE A LA LIGNE   *
;* SUIVANTE SUR LE MINITEL*
;*****
;
PRINT:
9C8F'   CD 9C5B'    CALL SENCAR;DE MEME QUE
; POUR UNE CHAINE NORMALE
9C92'   3E 0A       LD A,0AH;LINE FEED
9C94'   CD 9C51'    CALL EMASC;EN PLUS
9C97'   3E 0D       LD A,0DH;CARRIAGE RETURN
9C99'   CD 9C51'    CALL EMASC;EN PLUS
9C9C'   C9         RET;ET FIN
;
;*****
;
;
;*****
;* RECEPTION DU CODE      *
;* ASCII D'UN CARACTERE*
;* EMIS PAR LE MINITEL *
;*****
;
RECEPT:
9C9D'   FE 01      CP 01H;UN PARAMETRE SEULEMENT?
9C9F'   C0         RET NZ;NON ALORS RETOUR
;
9CA0'   DD 6E 00   LD L,(IX+00H);ADRESSE BASSE
9CA3'   DD 66 01   LD H,(IX+01H);ADRESSE HAUTE
; DE LA VARIABLE ENTIERE TRANSMISE
9CA6'   E5        PUSH HL;A SAUVER
9CA7'   F3        DI;INHIBER INTERRUPTION
9CA8'   CD 9D66'   CALL RE7BIT;RECEVOIR UN CARACTERE
; SUR LA LIGNE
9CAB'   FB        EI;AUTORISER INTERRUPTION
9CAC'   E1        POP HL;RECUPERER ADRESSE
9CAD'   77        LD (HL),A;ET SAUVER
; LE CODE DU CARACTERE RECU
9CAE'   23        INC HL;ADRESSE SUIVANTE
; POUR POINTER L'OCTET HAUT
9CAF'   AF        XOR A;ET ANNULER
9CB0'   77        LD (HL),A;CET OCTET HAUT
9CB1'   C9        RET;ET FIN
;
;*****
;
;
;*****
;* RECEPTION D'UNE CHAINE *
;* DE CARACTERES DONT LA  *
;* FIN EST SIGNALEE PAR  *
;* LA TOUCHE ENVOI      *
;* FRAPPEE SUR LE MINITEL*

```

```

;*****
;
RECAR:
9CB2' DD 6E 00 LD L,(IX+00H);ADRESSE DU
9CB5' DD 66 01 LD H,(IX+01H);POINTEUR
9CB8' 06 00 LD B,00;ZERO CARACTERES A
;L'INITIALISATION
9CBA' E5 PUSH HL
9CBB' 23 INC HL
9CBC' 5E LD E,(HL)
9CBD' 23 INC HL
9CBE' 56 LD D,(HL)
9CBF' EB EX DE,HL
9CC0' D1 POP DE
;POINTE LE PREMIER
;CARACTERE DE LA CHAINE
;
RECAR1:
9CC1' EB EX DE,HL
9CC2' 70 LD (HL),B;RANGE LE NOMBRE
;DE CARACTERES ACTUELS
9CC3' EB EX DE,HL
9CC4' F3 DI
9CC5' CD 9D66' CALL RE7BIT;CHERCHER CARACTERE
9CC8' FE 13 CP 013H;TOUCHE FONCTION?
9CCA' 28 21 JR Z,FONCTI
9CCC' FE 19 CP 019H;ACCES JEU SS2?
9CCE' FB EI
9CCF' 28 11 JR Z,SS2
9CD1' 04 INC B;UN CARACTERE DE PLUS
9CD2' 77 LD (HL),A;RANGER LE CARACTERE
;DANS LA CHAINE POINTEE PAR HL
9CD3' 23 INC HL
9CD4' EB EX DE,HL
9CD5' 70 LD (HL),B
9CD6' EB EX DE,HL
9CD7' C5 PUSH BC
9CD8' D5 PUSH DE
9CD9' E5 PUSH HL
9CDA' CD 9C51' CALL RMASC;EFFECTUER L'ECHO
;DU CARACTERE SUR LE MINITEL
9CDD' K1 POP HL
9CDE' D1 POP DE
9CDF' C1 POP BC
9CE0' 18 DF JR RECAR1;ET CONTINUER
;
SS2: ;ACCES JEU G2
9CE2' E5 PUSH HL
9CE3' D5 PUSH DE
9CE4' C5 PUSH BC
9CE5' CD 9D66' CALL RE7BIT;ANNULER
;LE PROCHAIN CARACTERE POUR
;IGNORER LES ACCENTS
9CE8' C1 POP BC
9CE9' D1 POP DE
9CEA' K1 POP HL
9CEB' 18 D4 JR RECAR1
;
FONCTI: ;TOUCHE FONCTION

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9CED'   CD 9D66'   CALL RE7BIT;CHERCHER LE
;CODE DE LA TOUCHE FONCTION
9CF0'   FB        EI
9CF1'   FE 41     CP 041H;TOUCHE ENVOI?
9CF3'   C8        RET Z;OUI ALORS FIN
9CF4'   FE 47     CP 047H;TOUCHE CORRECTION ?
9CF6'   28 06     JR Z,CORREC;EFFECTUER CORRECTION
9CF8'   FE 45     CP 045H;TOUCHE ANNULATION
9CFA'   28 30     JR Z,ANNULE;EFFECTUER ANNULATION
9CFC'   18 C3     JR RECAR1;SINON RETOURNER
;ATTENDRE NOUVEAU CARACTERE
;
;*** CORRECTION ***
;
CORREC:
LD A,B; NOMBRE DE CARACTERES
9CFE'   78        CP 00;PAS DE CARACTERES?
9CFF'   FE 00     JR NZ,CORREC1;SI, ALORS SUITE
9D01'   20 0D     ERRRE:
;ERREUR DE CORRECTION
9D03'   C5        PUSH BC
9D04'   D5        PUSH DE
9D05'   E5        PUSH HL
9D06'   3E 07     LD A,07H;EMMETTRE UN BIP
9D08'   CD 9C51' CALL EMASC
9D0B'   E1        POP HL
9D0C'   D1        POP DE
9D0D'   C1        POP BC
9D0E'   18 B1     JR RECAR1;RECOMMENCER
;
CORREC1:
9D10'   C5        PUSH BC
9D11'   D5        PUSH DE
9D12'   E5        PUSH HL
9D13'   3E 08     LD A,08H;RETOUR ARRIERE
9D15'   CD 9C51' CALL EMASC
9D18'   3E 20     LD A,020H;EFFACER LE
;CARACTERE PRECEDENT
9D1A'   CD 9C51' CALL EMASC
9D1D'   3E 08     LD A,08; ET REVENIR UN
;CARACTERE EN ARRIERE
9D1F'   CD 9C51' CALL EMASC
9D22'   E1        POP HL
9D23'   D1        POP DE
9D24'   C1        POP BC
9D25'   05        DEC B
9D26'   EB        EX DE,HL
9D27'   70        LD (HL),B
9D28'   EB        EX DE,HL
9D29'   2B        DEC HL
9D2A'   18 95     JR RECAR1;ET CHERCHER A NOUVEAU
;
;*** ANNULATION LIGNE ***
;
ANNULE:
9D2C'   78        LD A,B;NB CARACTERES
9D2D'   FE 00     CP 00
9D2F'   28 D2     JR Z,ERRRE;SI ZERO ALORS ERREUR
;

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9D31'   C5
9D32'   D5
9D33'   E5
9D34'   C5
9D35'   3E 08
9D37'   C5
9D38'   CD 9C51'
9D3B'   C1
9D3C'   10 F7
9D3E'   C1
9D3F'   C5
9D40'   3E 20
9D42'   3E 20
9D44'   C5
9D45'   CD 9C51'
9D48'   C1
9D49'   10 F5
9D4B'   C1
9D4C'   3E 08
9D4E'   C5
9D4F'   CD 9C51'
9D52'   C1
9D53'   10 F7
9D55'   E1
9D56'   D1
9D57'   C1
9D58'   06 00
9D5A'   D5
9D5B'   EB
9D5C'   70
9D5D'   23
9D5E'   5E
9D5F'   23
9D60'   56
9D61'   EB
9D62'   D1
9D63'   C3 9CC1'

ANNUL1:
PUSH BC
PUSH DE
PUSH HL
PUSH BC;SAUVE NB CARACTERES
ANNUL2:
LD A,08;RETOUR ARRIERE
PUSH BC
CALL EMASC; A EMMETTRE
POP BC
DJNZ ANNUL2;JUSQU'A CE QUE
;TOUT LE TEXTE SOIT "REMONTE"
POP BC;REPREND LE NB DE CARACTERES
PUSH BC; ET RESAUVE
ANNUL3:
LD A,020H;CETTE FOIS CI
;EFFACER AVEC DES ESPACES
LD A,020H
PUSH BC
CALL EMASC
POP BC
DJNZ ANNUL3;TOUT EFFACE?
POP BC;RECUPERE NB CARACTERES
ANNUL4:
;EQU

LD A,08H;RETOUR ARRIERE
;POUR RETOURNER EN DEBUT DE LIGNE
PUSH BC
CALL EMASC
POP BC
DJNZ ANNUL4;ENCORE?!
POP HL
POP DE
POP BC
LD B,00;PLUS DE CARACTERES
PUSH DE
EX DE,HL
LD (HL),B
INC HL
LD E,(HL)
INC HL
LD D,(HL)
EX DE,HL
POP DE
JP RECAR1;ALLER EN RECHERCHER
;
;*****
;
;
;*** ZONE DES SOUS-PROGRAMMES ***
;
;
;*****
;* RECEPTION ET RANGEMENT *
;* D'UN CARACTERE DANS *
;* LE REGFISTRE A *
;*****
;

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

RE7BIT:
;EQU

;
9D66'   CD 9D87'   CALL STWAIT;ATTENDRE BIT DE START
9D69'   C5         PUSH BC;SAUVER BC
9D6A'   06 07     LD B,7;SEPT BIT A RECEVOIR
;
9D6C'   CD 9DA5'   CALL LECTUR;LIRE UN BIT
9D6F'   F5         PUSH AF;ET LE SAUVER
;DANS LA PILE PARAMETRES
9D70'   CD 9DE6'   CALL TEMPO8;ATTENDRE 833 ms
9D73'   10 F7     DJNZ RE7BI1;ENCORE UN BIT A LIRE
9D75'   CD 9DE6'   CALL TEMPO8;ELIMINE PARITE
9D78'   CD 9DE6'   CALL TEMPO8;ELIMINE BIT STOP
9D7B'   06 07     LD B,7;SEPT CARACTERES
;A RECUPERER DE LA PILE
RE7BI1:
9D7D'   F1         POP AF;RECUPERE UN BIT
9D7E'   CB 11     RL C;ET RANGER DANS C
9D80'   10 FB     DJNZ RE7BI2;ENCORE UN BIT
9D82'   79         LD A,C;SINON REPRENDRE DANS A
9D83'   E6 7F     AND 07FH;ET GARDER LES
;SEPT BITS UTILES
9D85'   C1         POP BC;RECUPERER BC
9D86'   C9         RET;ET FIN
;
;*****
;
;*****
;* ATTENTE DU BIT DE START *
;* PRECEDENT LE CODE SERIE *
;* D'UN CARACTERE ENVOYE *
;* PAR LE MINITEL *
;*****
;
STWAIT:
9D87'   CD 9DA5'   CALL LECTUR;LIRE L'ETAT
9D8A'   38 FB     ;DE LA LIGNE BUSY
JR C,STWAIT;SI PAS START
;ALORS RECOMMENCER
9D8C'   CD 9D98'   CALL TEMPO4;ATTENDRE UN
;DEMI BIT DE START
9D8F'   CD 9DA5'   CALL LECTUR;ET RELIRE
9D92'   38 F3     JR C,STWAIT;SI CE N'EST PLUS
;LE BIT DE START ALORS RECOMMENCER
9D94'   CD 9DE6'   CALL TEMPO8;SINON ATTENDRE
;TEMPO POUR BIT SUIVANT
9D97'   C9         RET;ET RETOUR
;
;*****
;
;

```


Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

;*****
;* TEMPORISATION D'UN DEMI *
;* BIT POUR VERIFICATION *

;* DU BIT DE START *
;*****
;
TEMPO4:
9D98'   E5      PUSH HL;SAUVER HL
9D99'   F5      PUSH AF;ET AF
9D9A'   21 0036 LD HL,TEMP4;OCTET POUR LA
;DUREE DE LA TEMPORISATION
;
TEMP41:
9D9D'   2B      DEC HL;UNE BOUCLE DE MOINS

9D9E'   7D      LD A,L;HL
9D9F'   B4      OR H;EGAL A ZERO
9DA0'   20 FB   JR NZ,TEMP41;NON ALORS RECOMMENCER
9DA2'   F1      POP AF;RECUPERER
9DA3'   E1      POP HL;LES REGISTRES
9DA4'   C9      RET;ET FIN
;
;*****
;
;*****
;* LECTURE D'UN BIT SUR *

;* LA LIGNE BUSY DU PORT *
;* IMPRIMANTE *
;*****
;
LECTUR:
9DA5'   C5      PUSH BC;SAUVER BC
9DA6'   01 F500 LD BC,PORBUS;ADRESSE LIGNE BUSY

9DA9'   ED 78   IN A,(C);LIRE ETAT DU PORT
9DAB'   17      RLA;RECUPERER LE
9DAC'   17      RLA;BIT DANS CARRY
9DAD'   C1      POP BC;RECUPERER BC
9DAE'   C9      RET;ET RETOUR
;
;*****
;
;*****
;* GENERATION DE LA *

;* PARITE SUR LE 8IEME *
;* BIT DU CARACTERE ASCII *
;*****
;
PARITE:
9DAF'   E6 7F   AND 07FH;ANNULE BIT 7
9DB1'   EA 9DB6' JP PE,PARIT1;PARITE PAIRE?
9DB4'   F6 80   OR 080H;NON ALORS METTRE
;UNE PARITE PAIRE SUR BIT 7
PARIT1:

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

9DB6'   C9           ;PARITE PAIRE
                    RET;FIN
                    ;
                    ;*****
                    ;
                    ;*****
                    ;* CREATION D'UNE TRAME *
                    ;* DE DIX BIT A EMETTRE *
                    ;* DANS LE REGISTRE HL *
                    ;*****
                    ;
                    TRAME:
9DB7'   C5           PUSH BC
9DB8'   2E 00       LD L,0;TOUT
9DBA'   26 00       LD H,0;A ZERO
9DBC'   06 08       LD B,08H;8 BIT A TRAITER
                    ;
                    TRAM1:
9DBE'   1F           RRA
9DBF'   ED 6A       ADC HL,HL
9DC1'   10 FB       DJNZ TRAM1;PAS FINI?
9DC3'   37           SCF;FLAG A 1
9DC4'   ED 6A       ADC HL,HL;POUR BIT DE STOP
9DC6'   06 06       LD B,6;ENCORE 6 FOIS
                    TRAM2:
9DC8'   29           ADD HL,HL ;A DECALER SANS CARRY
9DC9'   10 FD       DJNZ TRAM2;TERMINE?
                    ;
9DCB'   C1           POP BC
9DCC'   C9           RET
                    ;
                    ;*****
                    ;
                    ;*****
                    ;* EMISSION D'UNE TRAME *
                    ;
                    ;* DE DIX BIT CONTENUE *
                    ;* DANS HL SUR LE BIT DE *
                    ;
                    ;* POIDS FAIBLE DU PORT *
                    ;* IMPRIMANTE D0 *
                    ;*****
                    ;
                    EMETTRE:
9DCD'   C5           PUSH BC;SAUVER
9DCE'   D5           PUSH DE;LES
                    ;
9DCF'   F5           PUSH AF;REGISTRES
9DD0'   F3           DI;INHIBER LES INTERRUPTIONS
                    ;
9DD1'   01 EF00     LD BC,PORIMP ;ADRESSE PORT
                    ; IMPRIMANTE
9DD4'   1E 0A       LD E,0AH;DIX CARACTERES A
                    ;EMETTRE
                    EMET1:
9DD6'   AF           XOR A;ANNULE A ET CARRY
9DD7'   29           ADD HL,HL;ROTATION DU BIT

```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```

;FORT DE H DANS CARRY
9DD8' 17 RLA;RECUPERE CARRY SUR
;BIT FAIBLE DE A
9DD9' ED 79 OUT (C),A;SORTIR LE

;BIT SUR LE PORT CONCERNE
9DDB' CD 9DE6' CALL TEMPO8;ATTENTE POUR
;PROCHAIN BIT
9DDE' 1D DEC K;PROCHAIN BIT
9DDF' 20 F5 JR NZ,EMET1;S'IL Y A?
9DE1' FB EI;AUTORISER A
;NOUVEAU LES INTERRUPTIONS
9DE2' F1 POP AF;RECUPERER
9DE3' D1 POP DE;LES
9DE4' C1 POP BC;REGISTRES

9DE5' C9 RET;FIN
;
;*****
;
;
;*****
;* TEMPORISATION PERMETTANT *
;* L'ATTENTE D'UN DELAI DE *
;* 833 MICROSECONDES ENTRE *

;* DEUX BITS ENVOYES EN *

;* SERIE SUR LE PORT *
;* DE L'IMPRIMANTE *
;*****
TEMPO8:
PUSH HL;SAUVER HL
PUSH AF;ET AF

9DE6' E5 LD HL,TEMP8;NOMBRE DE BOUCLES
9DE7' F5 ;POUR UNE TEMPO DE 833 ms ;
TEMP81:
DEC HL;UNE BOUCLE DE MOINS
LD A,L;HL
9DEB' 2B OR H;EGAL ZERO?
9DEC' 7D JR NZ,TEMP81;NON ALORS RECOMMENC
9DED' B4
9DEE' 20 FB

9DF0' F1 POP AF;RECUPERER
9DF1' E1 POP HL;AF ET HL
9DF2' C9 RET;RETOUR
;
;*****
;
;
;*** MESSAGES D'ERREUR ***

;
;*****
;* MESSAGE D'ERREUR DE *
;* LOCALISATION SUR LE *

;* MINITEL *
    
```

;*****

MESLOC:

```

9DF3' 0A 0D          DEFB 0AH,0DH
9DF5' 45 52 52 45   DEFB "ERREUR"
9DF9' 55 52
9DFB' 20 44 45      DEFB " DE"
9DFE' 20 43 4F 4F   DEFB " COORDONNEES"
9E02' 52 44 4F 4E
9E06' 4E 45 45 53
9E0A' 0A 0D          DEFB 0AH,0DH
9E0C' 00            DEFB 00H

```

MESINK:

```

9E0D' 0A 0D          DEFB 0AH,0DH
9E0F' 45 52 52 45   DEFB "ERREUR"
9E13' 55 52
9E15' 20 43 4F 55   DEFB " COULEUR"
9E19' 4C 45 55 52
9E1D' 20 44 45      DEFB " DE"
9E20' 20 43 41 52   DEFB " CARACTERE"
9E24' 41 43 54 45
9E28' 52 45
9E2A' 0A 0D          DEFB 0AH,0DH
9E2C' 00            DEFB 00H

```

MESPAP:

```

9E2D' 0A 0D          DEFB 0AH,0DH
9E2F' 45 52 52 45   DEFB "ERREUR"
9E33' 55 52
9E35' 20 43 4F 55   DEFB " COULEUR"
9E39' 4C 45 55 52
9E3D' 20 44 45      DEFB " DE"
9E40' 20 46 4F 4E   DEFB " FOND"
9E44' 44
9E45' 0A 0D          DEFB 0AH,0DH
9E47' 00            DEFB 00

```

END

Macros:

Symbols:

AFFICH	BB5A	ALLCLS	9A88'	ANNUL1	9D31'	ANNUL2	9D35'
ANNUL3	9D40'	ANNUL4	9D4C'	ANNULE	9D2C'	BELL	9BC7'
BS	9B9D'	CANCEL	9BC1'	CLIGNO	9B5B'	CLS	9A82'
COREC1	9D10'	CORREC	9CFE'	CUROFF	9AEE'	CURSON	9AE8'
DBG	9B87'	DBH	9B7C'	DBL	9B71'	DEBUT	9900'
DISJ	9B45'	EMASC	9C51'	EMCA1	9C6A'	EMET1	9DD6'
EMETTR	9DCD'	ERRIN1	9C17'	ERRINK	9C14'	ERRLO2	9ADA'
ERRLO3	9ADE'	ERRLOC	9ADB'	ERRPA1	9C41'	ERRPAP	9C3E'
ERRRE	9D03'	FIXE	9B66'	FONCTI	9CED'	G0	9B09'
G1	9B0F'	G2	9B15'	HOME	9BBB'	HT	9BA3'
INK	9BF7'	KERNEL	9A55'	KLOGTX	BCD1	LECTUR	9DA5'
LF	9BA9'	LOCATE	9AA2'	LOWER	9B30'	LOWVID	9AF3'
MASKOF	9A77'	MASKON	9A6C'	MESINK	9E0D'	MESLOC	9DF3'
MESPAP	9E2D'	NORMH	9B92'	NORVID	9AFE'	PAGE	9BE2'
PAPER	9C21'	PARIT1	9DB6'	PARITE	9DAF'	PORBUS	F500
PORIMP	EF00	PRINT	9C8F'	RC	9BB5'	RE7BI1	9D6C'
RE7BI2	9D7D'	RE7BIT	9D66'	RECAR	9CB2'	RECAR1	9CC1'
RECEPT	9C9D'	REPEAT	9A59'	ROULE	9BCD'	SENCAR	9C5B'
SEND	9C4B'	SENSE1	9C7B'	SENSE2	9C86'	SENSEQ	9C76'
SOULOF	9B50'	SOULON	9B45'	SS2	9CE2'	STWAIT	9D87'
TABLE	998F'	TEMP4	0036	TEMP41	9D9D'	TEMP8	006D
TEMP81	9DEB'	TEMPO4	9D98'	TEMPO8	9DE6'	TRAM1	9DBE'
TRAM2	9DC8'	TRAME	9DB7'	UPPER	9B1B'	VECTEU	990F'
VT	9BAF'						

No Fatal error(s)

Ce programme est implanté à partir de l'adresse &9900. Afin de vous permettre d'éventuelles modifications, si vous jugez utile d'en apporter, nous vous donnons ci-après les différences adresses concernant les procédures de gestion :

:ALLCLS	&9A88 à &9AA1
:BELL	&9BC7 à &9BCC
:BS	&9B9D à &9BA2
:CANCEL	&9BC1 à &9BC6
:CLIGNOTE	&9B5B à &9B65
:CLS	&9A82 à &9A87
:CURSOFF	&9AEE à &9AF2
:CURSON	&9AE8 à &9AED
:DBG	&9B87 à &9B91
:DBH	&9B7C à &9B86

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

:DBL	&9B71 à &9B7B
:DISJ	&9B45 à &9B44
:FIXE	&9B66 à &9B70
:G0	&9B09 à &9B0E
:G1	&9B0F à &9B14
:G2	&9B15 à &9B1A
:HOME	&9BBB à &9BC0
:HT	&9BA3 à &9BA8
:INK	&9BF7 à &9C20
:LF	&9BA9 à &9BAE
:LOCATE	&9AA2 à &9AE7
:LOWER	&9B30 à &9B44
:LOWVIDEO	&9AF3 à &9AFD
:MASKOFF	&9A77 à &9A81
:MASKON	&9A6C à &9A76
:NORMH	&9B92 à &9B9C
:NORMVIDEO	&9AFE à &9B08
:PAGE	&9BE2 à &9BF6
:PAPER	&9C21 à &9C4A
:PRINT	&9C8F à &9C9C
:RC	&9BB5 à &9BBA
:RECAR	&9CB2 à &9D65
:RECEPT	&9C9D à &9CB1
:REPEAT	&9A59 à &9A6B
:ROULEAU	&9BCD à &9BE1
:SEND	&9C4B à &9C75
:SENDSEQ	&9C76 à &9C8E
:SOULOFF	&9B50 à &9B5A
:SOULON	&9B45 à &9B4F
:UPPER	&9B1B à &9B2F
:VT	&9BAF à &9BB4
RE7BIT	&9D66 à &9D86
STWAIT	&9D87 à &9D97

Les messages d'erreurs se situent entre &9DF3 et &9E47.

Vous trouverez en fin de listing la table des symboles, classés par ordre alphabétiques, pour vous permettre de retrouver facilement un éventuel sous-programme.

III. Utilisation du programme Assembleur

Après avoir entré le programme Assembleur, vous pourrez l'assembler, et sauvegarder le code créé sur disquette (nous avons utilisé le nom "RSXMINIT").

Les adresses utiles sont comprises entre &9900 et &9E47.

Pour charger à partir du Basic, il suffira d'effectuer les instructions dans l'ordre suivant :

```
10 MEMORY &98FF
20 OPENOUT "ESSAI.":
30 REM LA DISQUETTE NE DOIT PAS
40 REM ETRE PROTEGEE EN ECRITURE
50 MEMORY HIMEN - 1
60 CLOSEOUT
70 LOAD"RSXMINIT.BIN",&9900
80 CALL &9900 : REM INITIALISE LES RSX
```

IV. Le chargeur Basic

Pour les programmeurs Basic, nous vous proposons ci-dessous le logiciel permettant de charger les codes hexadécimaux en mémoire.

```
10 REM *****
20 REM *** CREATION D'INSTRUCTIONS ***
30 REM *** DE TYPE RSX POUR LA ***
40 REM *** GESTION D'UNE LIAISON ***
50 REM *** SERIE ENTRE UN AMSTRAD ***
60 REM *** ET UN MINITEL PAR ***
70 REM ** LA FICHE PERI-INFORMATIQUE **
80 REM *****
90 REM
100 REM *** CHARGEUR BASIC ***
110 MEMORY &98FF
120 REM
130 ADR = &9900
140 I = 0
150 SOMME = 0
160 RESTORE 430
170 READ A$
180 IF A$ = "FIN" THEN 320
190 B$ = "&" + A$
200 B = VAL (B$)
210 SOMME = SOMME + B
220 POKE ADR,B
230 ADR = ADR + 1
240 I = I + 1
250 PRINT I;CHR$(13);
260 GOTO 170
270 REM
280 STOP
290 REM
300 REM *** SAUVEGARDE ***
310 REM
320 MODE 1
330 PRINT "POUR SAUVEGARDER LA ROUTINE R
N BINAIRE"
340 PRINT "SAVE ";CHR$(34);"RSXMINIT.BIN
";CHR$(34);".B.&9900,";RIGHT$(STR$(I),LE
N(STR$(I))-1)
350 PRINT
360 PRINT
370 PRINT "CHARGEMENT PAR"
380 PRINT
390 PRINT "MEMORY &98FF"
400 PRINT "LOAD ";CHR$(34);"RSXMINIT.BIN
";CHR$(34);".&98FF"
410 PRINT
420 REM
430 REM *** CODES HEXADECIMAUX ***
440 REM
450 REM *** INSTALLATION RSX ***
460 DATA 21,55,9A,01,0F,99,CD,D1
```



```
470 DATA BC,3E,C9,32,00,99,C9,8F
480 DATA 99
490 REM
500 REM *** TABLE DES SAUTS ***
510 DATA C3,4B,9C,C3,5B,9C,C3
520 DATA 76,9C,C3,8F,9C,C3,9D,9C
530 DATA C3,B2,9C,C3,A2,9A,C3,E8
540 DATA 9A,C3,EE,9A,C3,F3,9A,C3
550 DATA FE,9A,C3,F7,9B,C3,21,9C
560 DATA C3,82,9A,C3,88,9A,C3,09
570 DATA 9B,C3,0F,9B,C3,15,9B,C3
580 DATA 1B,9B,C3,30,9B,C3,45,9B
590 DATA C3,50,9B,C3,5B,9B,C3,66
600 DATA 9B,C3,71,9B,C3,7C,9B,C3
610 DATA 87,9B,C3,92,9B,C3,9D,9B
620 DATA C3,A3,9B,C3,A9,9B,C3,AF
630 DATA 9B,C3,B5,9B,C3,BB,9B,C3
640 DATA C1,9B,C3,C7,9B,C3,CD,9B
650 DATA C3,E2,9B,C3,59,9A,C3,6C
660 DATA 9A,C3,77,9A,C3,45,9B
670 REM
680 REM *** DEFINITION DES RSX ***
690 DATA 53
700 DATA 45,4E,C4,57,52,49,54,C5
710 DATA 53,45,4E,44,53,45,D1,50
720 DATA 52,49,4E,D4,52,45,43,45
730 DATA 50,D4,52,45,43,41,D2,4C
740 DATA 4F,43,41,54,C5,43,55,52
750 DATA 53,4F,CE,43,55,52,53,4F
760 DATA 46,C6,4C,4F,57,56,49,44
770 DATA 45,CF,4E,4F,52,4D,56,49
780 DATA 44,45,CF,49,4E,CB,50,41
790 DATA 50,45,D2,43,4C,D3,41,4C
800 DATA 4C,43,4C,D3,47,B0,47,B1
810 DATA 47,B2,55,50,50,45,D2,4C
820 DATA 4F,57,45,D2,53,4F,55,4C
830 DATA 4F,CE,53,4F,55,4C,4F,46
840 DATA C6,43,4C,49,47,4E,4F,54
850 DATA C5,46,49,58,C5,44,42,CC
860 DATA 44,42,C8,44,42,C7,4E,4F
870 DATA 52,4D,C8,42,D3,48,D4,4C
880 DATA C6,56,D4,43,D2,48,4F,4D
890 DATA C5,43,41,4E,43,45,CC,42
900 DATA 45,4C,CC,52,4F,55,4C,45
910 DATA 41,D5,50,41,47,C5,52,45
920 DATA 50,45,41,D4,4D,41,53,4B
930 DATA 4F,CE,4D,41,53,4B,4F,46
940 DATA C6,44,49,53,CA
950 REM
960 REM *** RESERVATION POUR KERNEL ***
970 DATA 30,30,30,30
```

```
980 REM
990 REM *** INSTRUCTION REPEAT ***
1000 DATA FE,01,C0,DD,7E,00,F5
1010 DATA 3E,12,CD,51,9C,F1,C6,40
1020 DATA CD,51,9C,C9
1030 REM
1040 REM *** INSTRUCTION MASKON ***
1050 DATA 3E,1B,CD,51
1060 DATA 9C,3E,58,CD,51,9C,C9
1070 REM
1080 REM *** INSTRUCTION MASKOFF ***
1090 DATA 3E
1100 DATA 1B,CD,51,9C,3E,5F,CD,51
1110 DATA 9C,C9
1120 REM
1130 REM *** INSTRUCTION CLS ***
1140 DATA 3E,0C,CD,51,9C,C9
1150 REM
1160 REM *** INSTRUCTION ALLCLS ***
1170 DATA 3E,1F,CD,51,9C,3E,40,CD
1180 DATA 51,9C,3E,41,CD,51,9C,3E
1190 DATA 18,CD,51,9C,3E,0C,CD,51
1200 DATA 9C,C9
1210 REM
1220 REM *** INSTRUCTION LOCATE ***
1230 DATA DD,6E,00,DD,66,01
1240 DATA 7C,FE,00,20,2E,7D,FE,00
1250 DATA 28,29,FE,29,30,25,F5,DD
1260 DATA 6E,02,DD,66,03,7C,FE,00
1270 DATA 20,18,7D,FE,19,30,13,F5
1280 DATA 3E,1F,CD,51,9C,F1,C6,40
1290 DATA CD,51,9C,F1,C6,40,CD,51
1300 DATA 9C,C9,F1,21,F3,9D,7E,FE
1310 DATA 00,C8,CD,5A,BB,23,18,F6
1320 REM
1330 REM *** INSTRUCTION CURSON ***
1340 DATA 3E,11,CD,51,9C,C9
1350 REM
1360 REM *** INSTRUCTION CURSOFF ***
1370 DATA 3E,14,CD,51,9C
1380 REM
1390 REM *** INSTRUCTION LOWVIDEO ***
1400 DATA 3E,1B,CD,51,9C
1410 DATA 3E,5D,CD,51,9C,C9
1420 REM
1430 REM *** INSTRUCTION NORMVIDEO ***
1440 DATA 3E,1B
1450 DATA CD,51,9C,3E,5C,CD,51,9C
1460 DATA C9
1470 REM
```

```
1480 REM *** INSTRUCTION G0 ***
1490 DATA 3E,0F,CD,51,9C,C9
1500 REM
1510 REM *** INSTRUCTION G1 ***
1520 DATA 3E,0E,CD,51,9C,C9
1530 REM
1540 REM *** INSTRUCTION G2 ***
1550 DATA 3E,19,CD,51,9C,C9
1560 REM
1570 REM *** INSTRUCTION UPPER ***
1580 DATA 3E,1B,CD,51,9C
1590 DATA 3E,3A,CD,51,9C,3E,6A,CD
1600 DATA 51,9C,3E,45,CD,51,9C,C9
1610 REM
1620 REM *** INSTRUCTION LOWER ***
1630 DATA 3E,1B,CD,51,9C,3E,3A,CD
1640 DATA 51,9C,3E,69,CD,51,9C,3E
1650 DATA 45,CD,51,9C,C9
1660 REM
1670 REM *** INSTRUCTION DISJ ***
1680 REM *** ET SOULON ***
1690 DATA 3E,1B,CD
1700 DATA 51,9C,3E,5A,CD,51,9C,C9
1710 REM
1720 REM *** INSTRUCTION SOULOFF ***
1730 DATA 3E,1B,CD,51,9C,3E,59,CD
1740 DATA 51,9C,C9
1750 REM
1760 REM *** INSTRUCTION CLIGNOTE ***
1770 DATA 3E,1B,CD,51,9C
1780 DATA 3E,48,CD,51,9C,C9
1790 REM
1800 REM *** INSTRUCTION FIXE ***
1810 DATA 3E,1B
1820 DATA CD,51,9C,3E,49,CD,51,9C
1830 DATA C9
1840 REM
1850 REM *** INSTRUCTION DBL ***
1860 DATA 3E,1B,CD,51,9C,3E,4E
1870 DATA CD,51,9C,C9
1880 REM
1890 REM *** INSTRUCTION DBH ***
1900 DATA 3E,1B,CD,51
1910 DATA 9C,3E,4D,CD,51,9C,C9
1920 REM
1930 REM *** INSTRUCTION DBG ***
1940 DATA 3E
1950 DATA 1B,CD,51,9C,3E,4F,CD,51
1960 DATA 9C,C9
1970 REM
```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

```
1980 REM *** INSTRUCTION NORMH ***
1990 DATA 3E,1B,CD,51,9C,3E
2000 DATA 4C,CD,51,9C,C9
2010 REM
2020 REM *** INSTRUCTION BS ***
2030 DATA 3E,08,CD,51,9C,C9
2040 REM
2050 REM *** INSTRUCTION HT ***
2060 DATA 3E,09,CD,51,9C,C9
2070 REM
2080 REM *** INSTRUCTION LF ***
2090 DATA 3E,0A,CD,51,9C,C9
2100 REM
2110 REM *** INSTRUCTION VT ***
2120 DATA 3E,0B,CD,51,9C,C9
2130 REM
2140 REM *** INSTRUCTION RC ***
2150 DATA 3E,0D,CD,51,9C,C9
2160 REM
2170 REM *** INSTRUCTION HOME ***
2180 DATA 3E,1E,CD,51,9C,C9
2190 REM
2200 REM *** INSTRUCTION CANCEL ***
2210 DATA 3E,18,CD,51,9C,C9
2220 REM
2230 REM *** INSTRUCTION BEEP ***
2240 DATA 3E,07,CD,51,9C,C9
2250 REM
2260 REM *** INSTRUCTION ROULEAU ***
2270 DATA 3E,1B,CD
2280 DATA 51,9C,3E,3A,CD,51,9C,3E
2290 DATA 69,CD,51,9C,3E,43,CD,51
2300 DATA 9C,C9
2310 REM
2320 REM *** INSTRUCTION PAGE ***
2330 DATA 3E,1B,CD,51,9C,3E
2340 DATA 3A,CD,51,9C,3E,6A,CD,51
2350 DATA 9C,3E,43,CD,51,9C,C9
2360 REM
2370 REM *** INSTRUCTION INK ***
2380 DATA DD
2390 DATA 66,01,DD,6E,00,7C,FE,00
2400 DATA 20,12,7D,FE,08,30,0D,F5
2410 DATA 3E,1B,CD,51,9C,F1,C6,40
2420 DATA CD,51,9C,C9,21,0D,9E,7E
2430 DATA FE,00,C8,CD,5A,BB,23,18
2440 DATA F6
2450 REM
2460 REM *** INSTRUCTION PAPER ***
2470 DATA DD,66,01,DD,6E,00,7C
```

```
2480 DATA FE,00,20,12,7D,FE,08,30
2490 DATA 0D,F5,3E,1B,CD,51,9C,F1
2500 DATA C6,50,CD,51,9C,C9,21,2D
2510 DATA 9E,7E,FE,00,C8,CD,5A,BB
2520 DATA 23,18,F6
2530 REM
2540 REM *** INSTRUCTION SEND ***
2550 DATA FE,01,C0,DD,7E
2560 DATA 00,CD,AF,9D,CD,B7,9D,CD
2570 DATA CD,9D,C9
2580 REM
2590 REM *** INSTRUCTION WRITE ***
2600 DATA FE,01,C0,DD,6E
2610 DATA 00,DD,66,01,46,23,5E,23
2620 DATA 56,EB,7E,E5,C5,CD,51,9C
2630 DATA C1,E1,23,10,F5,C9
2640 REM
2650 REM *** INSTRUCTION SENDSEQ **
2660 DATA FE,00
2670 DATA C8,47,57,DD,7E,00,F5,DD
2680 DATA 23,DD,23,10,F6,42,F1,C5
2690 DATA CD,51,9C,C1,10,F8,C9
2700 REM
2710 REM *** INSTRUCTION PRINT ***
2720 DATA CD
2730 DATA 5B,9C,3E,0A,CD,51,9C,3E
2740 DATA 0D,CD,51,9C,C9
2750 REM
2760 REM *** INSTRUCTION RECEIPT ***
2770 DATA FE,01,C0
2780 DATA DD,6E,00,DD,66,01,E5,F3
2790 DATA CD,66,9D,FB,E1,77,23,AF
2800 DATA 77,C9
2810 REM
2820 REM *** INSTRUCTION RECAR ***
2830 DATA DD,6E,00,DD,66,01
2840 DATA 06,00,E5,23,5E,23,56,EB
2850 DATA D1,EB,70,EB,F3,CD,66,9D
2860 DATA FE,13,28,21,FE,19,FB,28
2870 DATA 11,04,77,23,EB,70,EB,C5
2880 DATA D5,E5,CD,51,9C,E1,D1,C1
2890 DATA 18,DF,E5,D5,C5,CD,66,9D
2900 DATA C1,D1,E1,18,D4,CD,66,9D
2910 DATA FB,FK,41,C8,FE,47,28,06
2920 DATA FE,45,28,30,18,C3,78,FE
2930 DATA 00,20,0D,C5,D5,E5,3E,07
2940 DATA CD,51,9C,E1,D1,C1,18,B1
2950 DATA C5,D5,E5,3E,08,CD,51,9C
2960 DATA 3E,20,CD,51,9C,3E,08,CD
2970 DATA 51,9C,E1,D1,C1,05,EB,70
```

```
2980 DATA EB,2B,18,95,78,FE,00,28
2990 DATA D2,C5,D5,E5,C5,3E,08,C5
3000 DATA CD,51,9C,C1,10,F7,C1,C5
3010 DATA 3E,20,3E,20,C5,CD,51,9C
3020 DATA C1,10,F5,C1,3E,08,C5,CD
3030 DATA 51,9C,C1,10,F7,E1,D1,C1
3040 DATA 06,00,D5,EB,70,23,5E,23
3050 DATA 56,EB,D1.C3,C1,9C
3060 REM
3070 REM ** ZONE DES SOUS-PROGRAMMES *
3080 REM
3090 REM *** RE7BIT ***
3100 DATA CD,87
3110 DATA 9D,C5,06,07,CD,A5,9D,F5
3120 DATA CD,E6,9D,10,F7,CD,E6,9D
3130 DATA CD,E6,9D,06,07,F1,CB,11
3140 DATA 10,FB,79,E6,7F,C1.C9
3150 REM
3160 REM *** STWAIT ***
3170 DATA CD
3180 DATA A5,9D,38,FB,CD,98,9D,CD
3190 DATA A5,9D,38,F3,CD,E6,9D,C9
3200 REM
3210 REM *** TEMPO4 ***
3220 DATA E5,F5,21,36,00,2B,7D,B4
3230 DATA 20,FB,F1,E1,C9
3240 REM
3250 REM *** LECTUR ***
3260 DATA C5,01,00
3270 DATA F5,ED,78,17,17,C1,C9
3280 REM
3290 REM *** PARITE ***
3300 DATA E6
3310 DATA 7F,EA,B6,9D,F6,80,C9
3320 REM
3330 REM *** TRAME ***
3340 DATA C5
3350 DATA 2E,00,26,00,06,08,1F,ED
3360 DATA 6A,10,FB,37,ED,6A,06,06
3370 DATA 29,10,FD,C1,C9
3380 REM
3390 REM *** EMETTRE ***
3400 DATA C5,D5,F5
3410 DATA F3,01,00,EF,1E,0A,AF,29
3420 DATA 17,ED,79,CD,E6,9D,1D,20
3430 DATA F5,FB,F1,D1,C1,C9
3440 REM
3450 REM *** TEMPO8 ***
3460 DATA E5,F5
3470 DATA 21,6D,00,2B,7D,B4,20,FB
```

Après avoir entré le programme, sauvegardez-le sur disquette, et lancez-le.
Suivez les instructions de sauvegarde du fichier binaire ainsi créé.
Lors d'une utilisation ultérieure, utilisez le logiciel proposé dans le paragraphe précédent.

