

# MINITEL:

## LE MEDIA ELECTRONIQUE

**Au même titre que le téléphone, le Minitel fait désormais partie de notre vie de tous les jours. Mais, si tout un chacun parvient, au bout d'un bref apprentissage, à se servir correctement de l'un et de l'autre, il est encore bien peu de personnes sachant comment ce terminal Teletel fonctionne. Aussi ne nous pencherons-nous pas sur la « pratique » du Minitel mais sur son fonctionnement.**

**A**insi sommes-nous sûrs qu'à la fin de cette étude vous comprendrez parfaitement comment fonctionne ce nouveau média. Que les amoureux du hard se rassurent, ils trouveront en annexe un montage très simple et très peu coûteux qui leur permettra d'utiliser leur appareil comme modem. A eux toutefois de développer le logiciel adéquat pour l'utiliser.

### Il était une fois... le codage de l'information

Chacun d'entre vous connaît le code ASCII, qui permet de transformer les états haut et bas du courant, rassemblés par groupes de huit sous forme d'octets, en symboles alpha-numériques et en codes de contrôle. Dans le cas du Minitel, c'est un peu la même chose. Le code utilisé par le Minitel a été défini dans la recommandation numéro 5 du CCITT (Comité Consultatif International pour le Télégraphe et le Téléphone). Il est constitué de mots de sept bits,

ce qui permet de représenter 128 caractères. On y trouve :

- des instructions commandant les diverses fonctions du Minitel (couleur du fond et des caractères, signal d'envoi, passage en clignotement, loupe, etc.) ;
- les lettres de l'alphabet (majuscules et minuscules), soit 52 « mots » ;
- les dix chiffres ;
- ainsi que différents signes typographiques et des caractères dits alpha-mosaïques (caractères semi-graphiques, très utiles pour créer des masques d'écran).

Mais, tout comme ce qui se passe dans une transmission asynchrone (c'est-à-dire lorsqu'on envoie les bits les uns à la suite des autres, et non en même temps comme dans une liaison parallèle), il est nécessaire d'adjoindre d'autres bits pour signaler quand commence un mot et quand il prend fin. C'est pourquoi chaque caractère se voit précédé d'un bit de « start » et suivi d'un bit de « stop ». Mais ce n'est pas tout. En réalité, le huitième bit de chaque octet sert à vérifier si le code transmis l'a été correctement. On appelle ce bit « bit de parité » car il prend la valeur zéro si le nombre de bits à 1 du caractère est pair, et la valeur 1 si ce nombre est impair. Ainsi l'ordinateur a-t-il un moyen sûr de vérifier si chaque caractère reçu est bien similaire à celui qui a été transmis. S'il y a une erreur de transmission, le Minitel la détecte donc aussitôt et affiche, à la place du caractère incorrect (quand il ne s'agit pas d'un caractère de contrôle), un point d'interrogation à l'envers. Nous reviendrons plus en détail sur les codes de contrôle du Minitel, mais auparavant il nous faut vous expliquer comment s'effectue la transmission des octets.

Pour transmettre des données par le canal téléphonique, on utilise des ondes sinusoïdales. Le réseau électrique fonctionne en effet avec un courant alternatif. Ce courant a comme propriété de voir varier sa tension un certain nombre de fois par seconde. Dans le cas de l'EDF, cette variation se produit 50 fois par seconde. Cette variation, encore appelée fréquence, est donc de 50 Hz (contre 60 pour le réseau nord-américain). La durée de cette variation s'appelle la période ; elle est inversement proportionnelle à la fréquence. Plus la fréquence est élevée et plus les périodes qui la composent sont courtes. Ce qui nous donne la relation suivante :

$$P = 1/F$$

Sur le téléphone, la tension varie du fait du courant produit par la bobine associée au microphone. Lorsque vous parlez, le son de votre voix fait vibrer la membrane du micro, qui excite la bobine, produit un courant, qui excite la bobine du récepteur de votre correspondant, récepteur dont la membrane se met à son tour à vibrer, reproduisant le son de votre voix. Mais ici surgit un petit problème. Sur le réseau téléphonique, on ne peut pas faire passer n'importe quelle fréquence. En réalité, les ondes transmises sont cantonnées entre deux valeurs extrêmes qui délimitent ainsi ce que l'on appelle une bande de fréquence. C'est pourquoi il n'est pas possible de faire transiter les données à trop grande vitesse (sauf sur des lignes spécialisées) si l'on ne veut pas récupérer un splendide charabia à la réception. Pour transmettre les messages du Minitel, on associe au symbole 0 un signal sinusoïdal de durée T et de fréquence  $f_0$  et au symbole 1 un signal sinusoïdal identique au précé-

*Le minitel n'est pas une entité unique : il est constitué de quatre organes.*



Photo SIC-PTT

dent mais de fréquence différente  $f_1$ . Ce déplacement de fréquences est identique à celui que produit votre voix lorsque vous modulez des sons pour les transformer en phonèmes. C'est pourquoi on parle aussi ici de modulation. Mais vous savez très bien que les signaux transmis par un ordinateur (ou un terminal) sont des signaux carrés et non pas sinusoïdaux. Aussi s'avère-t-il nécessaire de les transformer, à savoir de les moduler. Cette opération est effectuée par le modulateur du Minitel. De même, la réception d'un signal modulé doit être démodulée pour être intelligible par le terminal. Dans le Minitel, ces deux opérations s'exécutent au moyen d'un appareil modulateur-démodulateur. Il semble alors évident que ce modem transmet et reçoit à une certaine vitesse. Cette vitesse dépend de la durée de transmission d'un élément binaire, c'est-à-dire le temps pendant lequel sont émises (ou reçues) les fréquences  $f_0$  ou  $f_1$ . Cette rapidité est l'inverse de cette durée. Elle s'exprime en bits/seconde soit :

$$R = 1/T$$

Le modem du Minitel fonctionne selon l'avis V23 du CCITT, c'est-à-dire qu'il autorise des échanges simultanés (full-duplex) à 1 200 bauds dans le sens réception et 75 bauds (7,5 bits/seconde) dans le sens de l'émission. La modulation s'effectue par déplacement d'une fréquence de  $\pm 30$  Hz sur une fréquence constante de 420 Hz, appelée porteuse. Ce sont ces pics et ces

crêtes qui donneront nos symboles 1 et 0. Encore faut-il que le niveau sonore de l'émission soit suffisant. En réception, le démodulateur transforme (nous verrons comment dans quelques instants) une fréquence de  $\pm 400$  Hz placée sur une porteuse à 1 700 Hz en symbole 1 ou 0. Cette transformation des déplacements de modulation en données significatives est assurée par un coupleur asynchrone qui peut être soit logiciel soit matériel, et qui constitue ou reconstitue nos paquets de 10 octets (un bit de start (0), 7 bits de données, un bit de parité, un bit de stop (1)).

Notons au passage que le premier bit reçu est le bit de poids faible. Ce qu'il vous faut savoir également c'est que, dès la mise sous tension du Minitel, le modem est disponible, le microprocesseur l'obligeant en permanence à boucler sur lui-même. Si bien qu'il réémet tout ce qu'il reçoit. Pour vous en assurer, il vous suffit d'enfoncer une touche du clavier alphabétique, et vous verrez s'afficher le caractère émis à l'écran, ce qui constitue un parfait moyen pour savoir si votre modem fonctionne correctement. Ce type de fonctionnement est appelé « écho ». C'est aussi ce qui se passe lorsque le Minitel est connecté à un serveur ; ce que vous tapez est envoyé au modem qui l'envoie à la banque de données, celle-ci renvoyant les caractères tapés vers le modem qui les dirige alors vers l'écran. Ainsi est-on sûr de ce que reçoit le serveur. Sachez également qu'un logiciel de correction d'erreur

est implanté dans le modem. Mais voyons ce qui se passe lorsque l'on connecte, puis déconnecte le modem.

La connexion peut provenir soit d'une demande tapée au clavier (touche bascule **CONNEXION/FIN**), soit d'une demande provenant de la prise péri-informatique du Minitel. Que se passe-t-il alors ? Le Minitel effectue les opérations suivantes :

- basculement de la ligne téléphonique vers le modem (c'est pourquoi votre poste devient « sourd ») ;
- affichage de la lettre C qui clignote dans le coin supérieur droit de l'écran ;
- détection d'une fréquence à 1 300 Hz, émise par le système central, pendant une durée minimum de 1,7 seconde ;
- émission permanente d'une fréquence à 390 Hz dès que la fréquence à 1 300 Hz a été détectée, ce qui entraîne la fin de la phase de détection et le passage en mode émission/réception. Pendant ce temps, le logiciel de gestion du Minitel, que l'on appelle le **Protocole**, effectue ce fameux changement d'état ; ceci se produit au plus tard 3,9 millisecondes après l'envoi de la fréquence à 390 Hz.

Quant à la déconnexion, elle peut survenir dans quatre cas :

- double appui de la touche **CONNEXION/FIN** ;
- lorsque le serveur ou un périphérique la commande ;
- lorsque l'on perd la porteuse ;
- enfin, tout simplement, si vous débranchez votre Minitel. Dans les trois premiers cas, le Protocole transmet l'ordre de déconnexion au modem et provoque l'affichage de la lettre F dans le coin supérieur droit de l'écran. Le modem se remet alors à boucler sur lui-même. Dans tous les cas, le poste téléphonique est à nouveau disponible.

Tout cela est bel et bien, nous direz-vous, mais le Minitel est un terminal et non un simple modem (si seul ce dernier créneau vous intéresse, reportez-vous au montage qui



Photo SIC-PTT

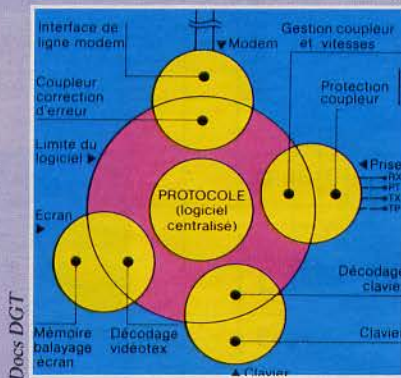


Fig. 1. - Architecture du minitel.

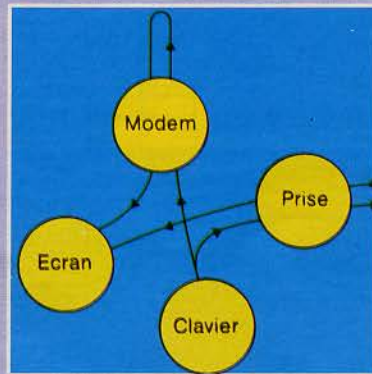


Fig. 2. - Aiguillage standard en mode local.

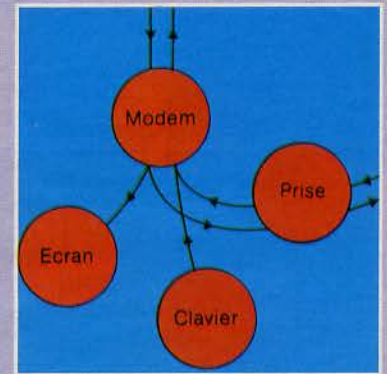


Fig. 3. - Aiguillage standard en mode connecté.

permet de s'en servir de cette manière). Tout à fait d'accord ! Aussi, voyons maintenant de quoi est faite la bête.

## Minitel : quatre organes pour un cerveau

En réalité, la « dissection » du Minitel nous apprend qu'il est constitué de quatre organes, reliés entre eux par un réseau « nerveux » à double sens, les nerfs moteurs et sensitifs étant eux-mêmes sous le contrôle du cerveau déjà mentionné, à savoir le Protocole. Nous venons d'examiner le premier de ces organes, le modem. Les trois autres sont le clavier, l'écran et la prise péri-informatique. Le cerveau assure l'envoi des données vers un ou plusieurs organes, tant et si bien que l'on peut assimiler le Minitel à un réseau en étoile dont la gestion serait centralisée (fig. 1). Disons tout de suite que le « cerveau » réagit différemment selon que l'appareil est en mode local (ce que l'on pourrait appeler son « dialogue intérieur ») ou en mode connecté. Dans le premier cas (fig. 2) le modem boucle sur lui-même et les trajets d'information sont les suivants :

- clavier vers modem ;
- modem vers écran ;
- clavier vers prise péri-informatique ;
- prise péri-informatique vers écran.

Dans le second cas, les nerfs de notre « étoile de mer » transmettront l'influx informatique de cette façon (fig. 3) :

- clavier vers modem ;
- modem vers écran ;
- modem vers prise péri-informatique ;

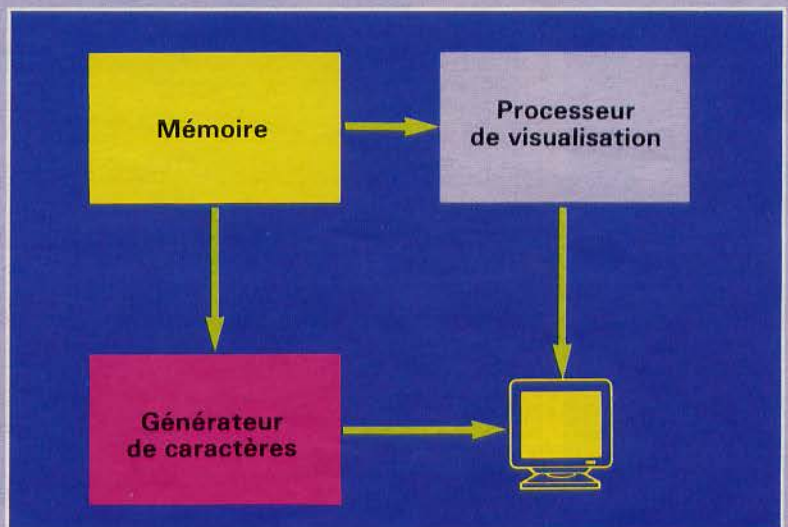


Fig. 4. - Schéma conceptuel d'un automate de visualisation.

● prise péri-informatique vers modem.

Vous remarquerez tout de suite qu'ici il est parfaitement possible de multiplexer les données reçues via la prise péri-informatique avec celles entrées au clavier. Rien n'empêche donc le petit malin que vous êtes de se servir de la conjonction ordinateur Minitel pour réaliser un système d'interrogation télématique biposte. A vous, toutefois, de développer le logiciel nécessaire pour ce faire. Mais revenons à nos organes et voyons plus avant leur structure et leur mode de fonctionnement.

## Pleine page sur l'écran

Dès sa mise sous tension, le Minitel fonctionne en mode pleine page. Il existe également d'autres modes tels que le mode rouleau, le mode loupe, etc., mais pour fonctionner ils

doivent être spécifiquement activés.

A dire vrai, notre organe écran est un ensemble composé de trois sous-organes : le tube cathodique, un automate de visualisation, lui-même tripartite, et un logiciel de décodage videtex.

L'automate possède trois parties :

- une mémoire qui a une capacité semblable au nombre d'octets que peut contenir une page de l'écran (soit 25 rangées de 40 emplacements caractères, chaque emplacement occupant 2 octets, ce qui nous donne une taille mémoire de 2 Ko, et une capacité de 1 000 caractères). Cette mémoire est mise à jour en permanence avec les données que lui transmet le logiciel de décodage et est lue 50 fois par seconde par le processeur de visualisation (cette fréquence correspond au balayage de l'écran) ;
- un générateur de caractères, en ROM contenant les 127 symboles visualisables ;

Le minitel dispose de trois jeux de caractères dont 64 mosaïques semi-graphiques.

				b7	0	0	1	1	1	1	
				b6	1	1	0	0	1	1	
				b5	0	1	0	1	0	1	
					2	3	4	5	6	7	
b4	b3	b2	b1								
0	0	0	0	0		0	@	P		p	
0	0	0	1	1		!	1	A	Q	a	q
0	0	1	0	2		"	2	B	R	b	r
0	0	1	1	3		#	3	C	S	c	s
0	1	0	0	4		\$	4	D	T	d	t
0	1	0	1	5		%	5	E	U	e	u
0	1	1	0	6		&	6	F	V	f	v
0	1	1	1	7		'	7	G	W	g	w
1	0	0	0	8		(	8	H	X	h	x
1	0	0	1	9		)	9	I	Y	i	y
1	0	1	0	10		*	:	J	Z	j	z
1	0	1	1	11		+	;	K	[	k	
1	1	0	0	12		.	<	L	\	l	
1	1	0	1	13		-	=	M	]	m	
1	1	1	0	14		.	>	N	†	n	
1	1	1	1	15		/	?	O		o	

Docs DGT

Fig. 5. - Le tableau G0 fournit les codes des caractères alphanumériques standards du minitel.

				b7	0	0	1	1	1	1
				b6	1	1	0	0	1	1
				b5	0	1	0	1	0	1
					2	3	4	5	6	7
b4	b3	b2	b1							
0	0	0	0	0						
0	0	0	1	1						
0	0	1	0	2						
0	0	1	1	3						
0	1	0	0	4						
0	1	0	1	5						
0	1	1	0	6						
0	1	1	1	7						
1	0	0	0	8						
1	0	0	1	9						
1	0	1	0	10						
1	0	1	1	11						
1	1	0	0	12						
1	1	0	1	13						
1	1	1	0	14						
1	1	1	1	15						

Fig. 6. - Le tableau G1 indique les codes de caractères semi-graphiques du minitel.

• un processeur de visualisation qui va lire dans la mémoire de page les caractères fournis par le générateur et les transformer en signaux RVB envoyés à l'écran. C'est aussi lui qui, comme nous l'avons déjà dit, va s'occuper de gérer le balayage de l'écran. Pour bien saisir ceci, reportez-vous à la **figure 4**.

Nous allons maintenant examiner en détail le logiciel de décodage vidéotex, qui, en outre, nous permettra de mieux comprendre comment, avec 7 bits, on peut obtenir non seulement un premier jeu de 128 caractères (ceux dont nous vous avons déjà parlé) mais aussi deux jeux annexes, dont l'un est constitué de caractères semi-graphiques.

Si l'écran du Minitel comprend 25 lignes de 40 colonnes, un peu à la manière des ordinateurs familiaux de la précédente génération, la matrice d'une de ces cases n'est pas tout

à fait semblable. Elle est constituée par une série de 10 lignes de 8 points. La ligne doit être ici entendue comme l'unité de balayage horizontal d'une image télé. Nous avons vu que chaque case équivalait à 2 octets en mémoire de page, c'est donc qu'il est nécessaire d'opérer un transcodage pour passer de 7 bits à 2 octets ! Ceci est réalisé au moyen d'une mémoire ROM qui interprète les codes reçus depuis le modem, la prise péri-informatique ou le clavier, et les transforme en caractères visualisables. Mais au juste que peut-on voir apparaître sur un Minitel ? A dire vrai, trois jeux de caractères (**fig. 5, 6 et 7**), le premier comprenant toute une série de codes de contrôle, les lettres de l'alphabet non accentuées, quelques signes de ponctuation ainsi que les « portes » qui permettent de passer d'un jeu de caractères à l'autre. Le second jeu est

composé de caractères mosaïques qu'on obtient en divisant la matrice en six morceaux. Petite réflexion comme ça en passant : comment diviser 80 par 6 et tomber juste ? Très simple : on divise 80 par 6, ce qui donne des morceaux de 3 pixels de côté par 4 de long ; reste une ligne de 8 pixels. Bon sang, mais c'est bien sûr ! Pour faire « joli », on ajoute cette ligne aux morceaux du milieu. Pour ceux qui n'ont toujours rien compris, reportez-vous à la **figure 8**. Le dernier jeu est à nouveau alphabétique, il contient les caractères accentués, la cédille du ç, l'œ, des flèches, des signes de fractions, bref, tout ce qu'il faut pour écrire correctement dans la langue de Rabelais. Pour s'y retrouver parmi tous ces codes, le CCITT les a alignés dans des tableaux (ceux des **fig. 5, 6 et 7**). Comment savoir à quel code correspond le A ? Très simple : on va tout

# DOSSIER

				b7	0	0	1	1	1	1
				b6	1	1	0	0	1	1
				b5	0	1	0	1	0	1
a4	b3	b2	b1		2	3	4	5	6	7
0	0	0	0	0		°				
0	0	0	1	1		±	-			
0	0	1	0	2			-			
0	0	1	1	3		£	-			
0	1	0	0	4		\$				
0	1	0	1	5						
0	1	1	0	6		#				
0	1	1	1	7						
1	0	0	0	8			÷	-		
1	0	0	1	9						
1	0	1	0	10					œ	œ
1	0	1	1	11				°		
1	1	0	0	12		-	¼			
1	1	0	1	13		↑	½			
1	1	1	0	14		→	¾			
1	1	1	1	15		↓				

Fig. 7. - Le tableau G2 correspond aux différents caractères spéciaux affichés.

d'abord regarder à quelle rangée il appartient. En regard de cette rangée, on trouve un paquet de 4 bits qui correspondent aux bits de poids faible de notre mot de 7 bits ; comme vous l'avez deviné, la colonne de notre A nous donne les 3 bits restants. Ce qui nous donne pour le A : 1000001, soit 65 en décimal (chic ! c'est de l'Ascii). Mais pourquoi faire simple quand on peut faire compliqué ? Au lieu de dire tout simplement « les caractères alphanumériques non accentués correspondent au standard ASCII et seront représentés comme tels », il est beaucoup plus « smart » de dire : *chaque code sera désigné par deux chiffres (ou un chiffre et une lettre pour ceux qui parlent couramment l'hexa), séparés par une barre de fraction. Si bien qu'au lieu de notre 65 nous aurons un magnifique 4/1 (avouez qu'ils l'ont fait exprès !). Bon, ce n'est pas tout ça, mais nos*

0 1 2 3 4 5 6 7

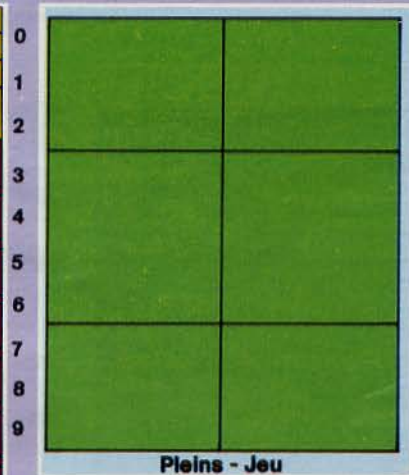


Fig. 8. - Les caractères semi-graphiques se décomposent en six points de taille inégale.

lettres accentuées ? Elles s'obtiennent en composant une suite de codes. Le premier de la séquence est un code de commande (les commandes occupent les deux premières colonnes du jeu 1), il a pour valeur 1/9 et permet au générateur de caractères de savoir qu'il doit aller chercher un accent dans le jeu G2 ; cette commande s'appelle SS2 (fig. 9). Le second code correspond à l'accent choisi dans le jeu de caractères d'accentuation. Tout comme sur une machine à écrire, on pose l'accent d'abord puis on pose le caractère sans avoir pour autant fait avancer le chariot de la machine. C'est exactement ce qui se produit ici ! Si nous voulons par exemple obtenir un ç, il nous faudra la séquence suivante :

1/9      4/11      6/3  
(SS2)    (cédille)    (c)

Deux choses encore à vous préciser avant d'aller plus loin : d'une part le programme contenu dans la ROM du générateur de caractères dispose d'une sécurité, si bien qu'il vous sera impossible de mettre une cédille au u. De même, les lettres liées ne comptent pas pour deux codes mais pour un seul. Par exemple l'e dans l'o sera codé

SS2 7/10

Passons maintenant aux caractères graphiques (fig. 6). Pour les obtenir, il suffit de composer le code S0 (0/14), ce qui a pour effet de vous faire basculer dans le jeu graphique. Vous y resterez jusqu'à ce que vous décidiez de passer la porte baptisée S1, dont le code se trouve en 0/15. Bien entendu, si vous êtes en mode semi-graphique et que vous essayez de repasser par SS2, vous vous heurterez à un mur.

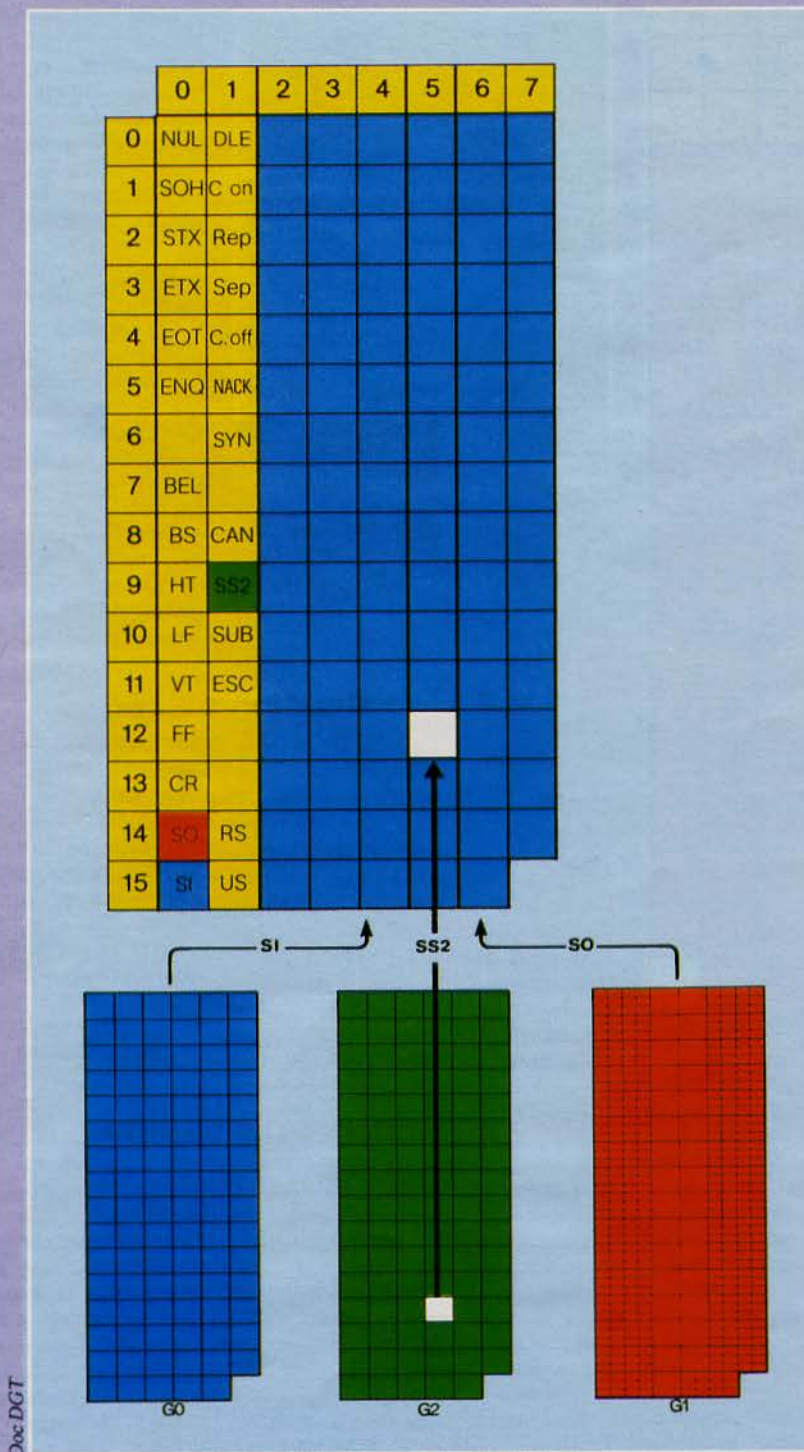
C'est bien beau tout ça, mais cela risque d'être un peu terne si cela demeure en noir et blanc et si on ne peut pas élargir ou allonger les caractères. Car, ô surprise, le Minitel dispose d'un codage des couleurs des lettres, et du fond, même s'il est en noir et blanc. Les couleurs, dans ce cas, se traduiront par une brillance plus ou moins intense. Mais que peut-on faire pour égayer le « décor » ?

On peut, au niveau des caractères alphabétiques, les souligner, les faire clignoter, les allonger, les élargir, et les faire passer en vidéo inverse. Quant au fond, on en détermine à loisir la couleur. Il est encore possible de masquer toute une suite de caractères et, bien sûr, de souligner en bloc. Les caractères semi-graphiques, eux, peuvent clignoter, changer de couleur, être disjoints ou masqués. Mais il ne faut surtout pas croire que l'on est obligé de déterminer pour chaque caractère toute une tribu d'attributs (sic). Il est tout à fait possible de procéder par zones. Une zone est une suite d'emplacements de caractères qui appartiennent à la même rangée physique (mais pas à la même colonne). Elle est délimitée au début de la rangée. Si vous n'avez pas plusieurs zones s'enchaînant les unes les autres, vous n'aurez pas besoin d'en placer un autre à la fin de chaque rangée. Ce délimiteur est signalé en mémoire de page par un mot bien particulier qui contient son identification, ainsi que les informations de changement d'attributs assignés à la zone. Par exemple : zone verte, soulignée, clignotante.

Il se trouve que le délimiteur est codé, en transmission, comme un espace, tant et si bien que, répété sur deux emplacements consécutifs d'une même rangée, il signalera le passage de la zone en double largeur, et, répété sur deux emplacements consécutifs d'une même colonne, il signalera la double hauteur de cette zone. Autre petite chose à savoir : lorsque vous effacez l'écran de votre Minitel, il s'ensuit un remplissage de l'écran par des pavés noirs semi-graphiques. Pourquoi ? Tout simplement pour empêcher que le lignage ou la couleur de fond n'anticipent sur l'inscription des caractères de la rangée courante.

Mais revenons à nos attributs. Ceux-ci sont codés au moyen d'une séquence « Escape » suivie d'un code d'appel. Tous ces codes sont ensuite

Il est possible de définir de nombreux attributs pour chaque caractère.



Doc DGT

Fig. 9. - Les trois jeux de caractères sont sélectionnés par l'un des codes suivants : SI, SS2 ou SO.

filtrés par le module écran du Minitel. Une fois définis au niveau du caractère, ils sont traités *en parallèle*, c'est-à-dire qu'ils n'apparaissent que conjointement à un caractère. Ainsi chaque signe peut avoir ses propres attributs.

Il n'est d'ailleurs pas nécessaire de coder chaque symbole, puisque les attributs de visualisation s'appliquent par défaut à tous les caractères qui les suivent.

Dans quels cas les attributs cessent-ils de s'appliquer ?

Tout d'abord, quand on rencontre une nouvelle batterie d'attributs ; là, pas de surenchère, mais un passage au nouveau jeu d'attributs pour la suite de caractères qui leur succède ; ensuite, lorsque l'on rencontre un séparateur bien particulier qui délimite des portions de page-écran. Ce séparateur est codé 1/F. Dans le tableau représenté à la figure 9, il est désigné par l'abréviation US (Unit Separator). Il signale la disponibilité d'une portion de l'écran pour recevoir de nouveaux caractères (quel que soit leur mode) et les attributs qui les accompagnent. Par défaut, cette portion de l'image repassera en mode alphanumérique et avec les couleurs (ou valeurs) codées lors de l'initialisation du terminal. Si bien qu'il est évident de dire que les attributs cessent de s'appliquer lorsque le terminal est réinitialisé. En tout, ce sont 28 codes qui sont utilisés pour définir les attributs. Listons-les rapidement :

- 8 codes de couleur de caractère (noir 4/0, rouge 4/1, vert 4/2, jaune 4/3, bleu 4/4, magenta 4/5, cyan 4/6, blanc 4/7 ;
- 8 codes de couleur de fond (mêmes couleurs que ci-dessus, mais les trois bits de poids fort donnent 5 comme valeur ; ainsi le fond bleu est codé 5/4) ;
- 4 codes définissant la taille du caractère (4/C grandeur normale, 4/D double hauteur, 4/E double largeur, 4/F double grandeur) ;
- 2 codes pour la fixité du caractère ou son clignotement (4/9 fixe, 4/8 caractère clignotant) ;
- 2 codes pour le début ou la fin de soulignement, car il est en effet possible de souligner toute une zone occupée par des caractères (début de lignage 5/A, fin de lignage 5/9) ;
- 2 codes concernant le fond ou son inversion (fond normal 5/C, fond inversé 5/D) ;
- 2 codes concernant le masquage ou le démasquage (5/8 masquage, 5/F démasquage). Le masquage empêche la visualisation des caractères qui auront été codés au moyen de cet attribut. Si le masquage est validé, les caractères ou les zones qui en dépendent apparaissent à l'écran sous forme de fond local noir. C'est une possibilité très intéressante quand on veut envoyer un texte de questions à choix multiples avec les réponses qui ne seront démasquées qu'en fonction du choix de l'utilisateur.

Là, vous aurez tout intérêt à faire

des repérages au moyen du séparateur US. Celui-ci est en effet suivi de deux caractères qui, de toute manière, ne sont jamais visualisés. Si les octets correspondant à ces caractères appartiennent aux colonnes 6 et 7, ils représentent respectivement, sous forme de nombres à 6 bits, le numéro de rangée et le numéro de colonne du premier caractère du sous-article. Si ils ont une autre valeur, la fonction US est ignorée. Ce qu'il faut savoir, c'est que la réception d'un séparateur d'article ou de sous-article remet les fonctions de visualisation dans un état initial, ce qui a donc pour effet d'enlever le masquage de notre zone. Seul petit problème : n'agrémentez pas vos réponses d'attributs tels que soulignement ou caractères graphiques, car US fait repasser le Minitel dans le jeu alphanumérique, si bien que vous risquez un joli mélange si vous n'y prenez garde.

Il est aussi possible de masquer et de démasquer l'écran d'un seul coup. Pour effectuer un masquage, employez la séquence ESC, 2/3, 2/0, 5/8. Le démasquage ne change que pour le troisième code, qui passe de 5/8 à 5/F.

## Questions de taille

Nous parlions précédemment de la taille des caractères. A ce propos, il est intéressant de voir comment l'automate de visualisation analyse les attributs permettant de passer en double grandeur ou en double hauteur. Lors de la lecture d'un attribut de double taille ou de double hauteur, il y a affectation immédiate aux deux rangées caractère, de deux caractéristiques, l'une pour la partie haute, l'autre, bien évidemment pour la partie basse, caractéristiques qui demeureront inchangées jusqu'à la fin de rangée. Notons tout de suite qu'il n'est pas possible d'utiliser ce type de caractères en rangée 1. Pourquoi ? Tout simplement parce que la rangée 0 est consacrée aux messages de service (par exemple, le « C qui vous indique que vous êtes connecté). Nous vous conseillons de n'utiliser cette rangée qu'avec parcimonie. Il est en effet préférable de la réserver à l'inscription de messages de contrôle. Pour y accéder, vous emploierez la séquence US (notre fameux délimiteur d'article) 4/0 X/Y (où Y désigne le numéro de colonne, donc prenez soin de laisser X à 0!). Un petit truc : pour repas-



/11 DOUBLES TAILLES/

FF,US,\$30,\$32,ESC,DT

/DOUBLE/

ESC,DLIGN,DEL

/TAILLE/

US,\$30,\$35,ESC,DT

/DOUBLE/

ESC,FINV,ESP

/TAILLE/

US,\$30,\$37,ESC,FINV,\$61,\$65,\$6F

SS2,\$41,\$61,SS2,\$42,\$65,SS2,\$43,\$6F,SS2,\$4B,\$63

ESC,FN,\$61,\$65,\$6F

SS2,\$41,\$61,SS2,\$42,\$65,SS2,\$43,\$6F,SS2,\$4B,\$63

ESC,DLIGN,DEL,\$61,\$65,\$6F

SS2,\$41,\$61,SS2,\$42,\$65,SS2,\$43,\$6F,SS2,\$4B,\$63

US,\$30,\$39,ESC,DT,ESC,FINV,\$61,\$65,\$6F

SS2,\$41,\$61,SS2,\$42,\$65,SS2,\$43,\$6F,SS2,\$4B,\$63

ESC,FN,\$61,\$65,\$6F

SS2,\$41,\$61,SS2,\$42,\$65,SS2,\$43,\$6F,SS2,\$4B,\$63

ESC,DLIGN,DEL,\$61,\$65,\$6F

SS2,\$41,\$61,SS2,\$42,\$65

US,\$31,\$31,ESC,DLIGN,DEL,ESC,DT

SS2,\$43,\$6F,SS2,\$4B,\$63

US,\$31,\$33,ESC,FINV,\$41,\$45,\$4F

SS2,\$41,\$41,SS2,\$42,\$45,SS2,\$43,\$4F,SS2,\$4B,\$43

ESC,FN,\$41,\$45,\$4F

SS2,\$41,\$41,SS2,\$42,\$45,SS2,\$43,\$4F,SS2,\$4B,\$43

ESC,DLIGN,DEL,\$41,\$45,\$4F

SS2,\$41,\$41,SS2,\$42,\$45,SS2,\$43,\$4F,SS2,\$4B,\$43

US,\$31,\$35,ESC,DT,ESC,FINV,\$41,\$45,\$4F

SS2,\$41,\$41,SS2,\$42,\$45,SS2,\$43,\$4F,SS2,\$4B,\$43

ESC,FN,\$41,\$45,\$4F

SS2,\$41,\$41,SS2,\$42,\$45,SS2,\$43,\$4F,SS2,\$4B,\$43

ESC,DLIGN,DEL,\$41,\$45,\$4F

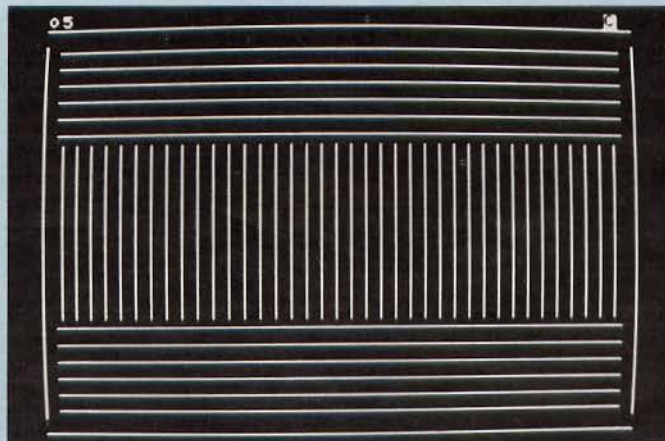
SS2,\$41,\$41,SS2,\$42,\$45

US,\$31,\$37,ESC,DLIGN,DEL,ESC,DT

SS2,\$43,\$4F,SS2,\$4B,\$43

Utilisation de la double taille.

Le langage Protocole permet d'affecter 8 couleurs aux caractères et au fond.



/05/

FF, RS, \$7E, REP, \$67, \$7B	\$7C, REP, \$65, \$7D, \$7B
\$60, REP, \$65, \$7D, \$7B	\$7C, REP, \$65, \$7D, \$7B
\$60, REP, \$65, \$7D, \$7B	\$7C, REP, \$65, \$7D, \$7B
\$60, REP, \$65, \$7D, \$7B	\$7C, REP, \$65, \$7D, \$7B
\$60, REP, \$65, \$7D, \$7B	\$7C, REP, \$65, \$7D, \$7B
\$60, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$60, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$60, REP, \$65, \$7D, \$7B
\$7C, REP, \$65, \$7D, \$7B	\$5F, REP, \$67

Exemple de remplissage d'écran.

ser à la ligne où se trouvait le curseur avant cette incursion, il suffit d'envoyer le code du saut de ligne, LF, soit 0/A. Par ailleurs, vous pourrez aussi quitter la rangée 0 en envoyant le code US. Revenons maintenant, après cette petite digression, à notre double taille. Si un autre caractère est désigné comme en double taille sur la rangée immédiatement inférieure ou supérieure aux rangées de notre double rangée, on ne visualisera que sa partie haute ou sa partie basse. Ça paraît peut-être un peu ésotérique mais, si vous y réfléchissez trente secondes, cela vous expliquera pourquoi, sur la version actuelle de votre Minitel, il ne vous est pas possible d'obtenir des caractères double taille qui soient placés en quinconce. Votre double taille, ainsi que tous les attributs définis au niveau du caractère, se retrouvera de plus transportée avec tous les déplacements d'écriture

dans l'écran, ceci jusqu'à ce que soit rencontré un code US ou FF (Form Feed, saut de page) ou encore RS (Record Separator). Dernier point avant de passer à autre chose : ce que fait en permanence l'automate de visualisation, c'est une analyse des séquences d'attribution au fur et à mesure qu'elles lui sont envoyées. Tant et si bien qu'on peut considérer l'envoi des attributs comme étant réalisé de façon sérielle. Ceci explique pourquoi l'automate néglige les séparateurs que sont les espaces tant que ceux-ci ne sont pas accompagnés par un quelconque changement d'attribut. Il les mémorise, mais n'en tient pas compte tant qu'il ne tombe pas sur un ESCape. Autre conséquence de cette attribution à la chaîne, c'est que l'on n'est pas obligé d'allouer un espace à la codification de chaque attribut, et c'est parfaitement normal. Imaginez que vous vouliez des caractères double largeur, bleus, clignotants. Si l'on appliquait cette méthode, on aurait

besoin de trois espaces pour réunir les conditions exigées, d'où une cruelle perte de place. Ceci explique aussi pourquoi la réception d'un caractère mosaïque permettant de valider l'attribut de couleur de fond lui fait jouer également le rôle de délimiteur pour une chaîne de caractères alphanumériques. Poussons le bouchon encore un peu plus loin. Que se passerait-il si nous avions, par exemple, le texte suivant :

VOUS QUI ENTREZ ICI ABANDONNEZ TOUTE ESPERANCE  
avec le VOUS en bleu, le QUI en rouge, le ENTREZ en noir, le ICI en magenta, et le reste à nouveau en rouge, et que nous introduisions ensuite des tirets à la place des espaces, ce qui donnerait :

VOUS-QUI-ENTREZ-ICI-  
ABANDONNEZ-TOUTE-ESPERANCE

Enfer et damnation ! Toute la phrase est devenue bleue ! Ceci s'explique quand on sait que le Minitel ne relit pas la mémoire de page. Ce qui fait que les attributs série ne sont pas transportables lors d'un mouvement dans l'écran de la position courante. Le simple fait de remplacer un séparateur par un autre caractère annule purement et simplement les attributs qui lui avaient été alloués. Que fait alors l'automate ? Il se contente de prendre pour attributs ceux de la zone située immédiatement à gauche, et ceci pour une bonne et simple raison : la transformation du séparateur provoque la réunion des deux zones.

### A droite, à gauche, en haut, en bas

Mais ce n'est pas tout que de s'amuser avec les couleurs. Pour que votre texte (ou votre dessin) ait vraiment l'air présentable, encore faut-il l'aérer, le disposer correctement, etc. Bref, le mettre en page. La page du Minitel est constituée de 25 lignes de 40 colonnes, exactement la taille du feuillet dans certains journaux. C'est peut-être pourquoi les « djinns » de la DGT l'ont baptisée du doux nom d'article. Enfin, peut importe, l'essentiel est de pouvoir traiter ce dernier de la même façon qu'un imprimeur traiterait le contenu d'un texte soumis à composition, c'est-à-dire de pouvoir indenter les paragraphes, sauter quelques lignes en fin de section, et aller à la page suivante pour le prochain chapitre. Voyons rapide-



Mnémonique	Code	Syntaxe	Action
1 OFF	6/0	1/B, 3/B, 6/0, code récepteur, code émetteur	● Arrêt d'aiguillage
2 ON	6/1	1/B, 3/B, 6/1, code récepteur, code émetteur	● Aiguillage
3 TO	6/2	1/B, 3/A, 6/2, code récepteur ou code émetteur	● Demande de status d'un module
4 FROM	6/3	1/B, 3/B, 6/3, code récepteur ou code émetteur, octet de status	● Réponse à une demande de status ou acquittement
5 NON DIFFUSION	6/4	1/B, 3/A, 6/4, code récepteur	● Diffusion restreinte des acquittements protocole
6 DIFFUSION	6/5	1/B, 3/A, 6/5, code récepteur	● Diffusion systématique des acquittements protocole
7 TRANSPARENCE	6/6	1/B, 3/A, 6/6, nombre d'octets	● Mise en transparence du protocole
8 DECONNEXION	6/7	1/B, 3/9, 6/7	● Déconnexion physique du modem
9 CONNEXION	6/8	1/B, 3/9, 6/8	● Assure la connexion du modem
10 START	6/9	1/B, 3/A, 6/9, mode de fonctionnement	● Mise en route d'un fonctionnement particulier du terminal
11 STOP	6/A	1/B, 3/A, 6/A, mode de fonctionnement	● Arrêt du mode de fonctionnement
12 PROG	6/B	1/B, 3/A, 6/B, octet de programmation vitesse	● Programmation des vitesses par le périphérique
13 RET1	6/C	1/B, 3/9, 6/C	● Retournement du modem
14 RET2	6/D	1/B, 3/9, 6/D	● Retournement inverse du modem
15 ACRET	6/E	1/B, 3/9, 6/E	● Acquiescement de retournement
16 OPPO	6/F	1/B, 3/9, 6/F	● Retournement pour l'opposabilité
17 OPPORE	6F, 3/1	1/B, 3/A, 6/F, 3/1	● Passage du mode opposé à esclave
18 STATUS TERMINAL	7/0	1/B, 3/9, 7/0	● Demande de status terminal
19 REP STATUS TERMINAL	7/1	1/B, 3/A, 7/1, octet de status terminal	● Réponse à la demande
20 STATUS FONCTIONNEMENT	7/2	1/B, 3/9, 7/2	● Demande de status fonctionnement
21 REP STATUS FONCTIONNEMENT	7/3	1/B, 3/A, 7/3, octet de status fonctionnement	● Réponse à la demande
22 STATUS VITESSE	7/4	1/B, 3/9, 7/4	● Demande de status vitesse
23 REP STATUS VITESSE	7/5	1/B, 3/A, 7/5, octet de status vitesse	● Réponse à la demande
24 STATUS PROTOCOLE	7/6	1/B, 3/9, 7/6	● Demande de status protocole
25 REP STATUS PROTOCOLE	7/7	1/B, 3/A, 7/7, octet de status protocole	● Réponse à la demande
26 IDEN1	7/8	1/B, 3/9, 7/8	● En tête de téléchargement Bloc 1
27 IDEN2	7/9	1/B, 3/9, 7/9	● En tête de téléchargement Bloc 2
28 ENQ RAM	7/A	1/B, 3/9, 7/A	● Identification du bloc 2
29 ENQ ROM	7/B	1/B, 3/9, 7/B	● Identification du terminal

Doc SIC-PTT

### Commandes de protocole.

ment les fonctions dont dispose le Minitel pour faire tout cela.

Ce sont les codes allant de 0/8 à 0/D qui vont ici nous servir. Parole d'évangile, le retour chariot est codé 0/D. Les tabulations horizontales et verticales correspondent respectivement à 0/9 (HT), 0/B (VT) et 0/A (LF). Elles provoquent le déplacement du curseur d'un caractère vers la droite, vers le haut, et d'une ligne vers le bas.

A noter que le retour chariot correspond à celui d'une machine à écrire, il ne provoque pas de saut de ligne, comme le Line Feed, mais ramène tout simplement le curseur dans la première colonne de la rangée en cours d'édition. L'espacement arrière (Back Space) nous est donné par 0/8. A tout ceci, il faut ajouter le « Form Feed » qui passe à l'article suivant ; il provoque l'effacement complet de l'écran, de la rangée 1 à la rangée 24.

Pour ramener le curseur dans le coin supérieur gauche de l'écran, nous emploierons le séparateur d'enregistrement RS, codé 1/E. Là aussi, on ne regrime pas sur la rangée 0 mais sur la rangée 1. Ce code constitue par ailleurs un séparateur explicite d'article. Pour diviser un article en plusieurs sous-articles, on emploie US, 1/F. Ce code est suivi de deux caractères non visualisés qui, ainsi que nous le disions auparavant, correspondent aux coordonnées des caractères qui seront ensuite envoyés à l'automate de visualisation.

Cette fonction est très intéressante, car c'est grâce à elle que nous pourrions positionner le curseur où bon nous semble. Imaginons que nous voulions que le mot BEATRICE commence à la 25<sup>e</sup> colonne de la 5<sup>e</sup> rangée. Nous enverrons donc les codes suivants :

**1/F (US), 4/5, 5/9, 4/2, 4/5, 4/1, 5/4, 5/2, 4/9, 4/3, 4/5.**

Vous remarquerez tout de suite qu'il y a plusieurs 4/5 qui ne correspondent absolument pas à la même chose ; dans le premier cas, 4/5 désigne le numéro de rangée, tandis que dans les deux autres ils représentent la lettre E. Explication : le premier doublet correspond à un caractère non visualisé et chiffré sur six bits utiles, qui nous donne le numéro de la rangée, et n'a donc rien à voir avec le codage de la lettre E. Traduisons ce doublet sur 6 bits, cela nous donnera 000101, c'est-à-dire bien 5 en décimal. De même, 25 nous donne en binaire 011001, soit 5/9. Compris ?

Mais vous aurez peut-être envie de faire suivre BEATRICE non pas par un détective privé mais par des espaces présentant les mêmes attributs. Pour ce faire, vous emploieriez la fonction CAN, codée 1/8. Notons au passage que ce code ne sert pas de délimiteur ; aussi les caractères envoyés ensuite, et qui débiteront à la première colonne de la rangée suivante (colonne 0 de la rangée 6 dans notre exemple) auront les mêmes attributs que BEATRICE.

Quelques mots encore sur le curseur. Il est tout à fait possible d'arrêter la visualisation de celui-ci en envoyant le code 1/4 (baptisé DC4 certainement par un passionné d'aéronautique !). Pour faire réapparaître le petit carré clignotant, il suffit d'envoyer le code DC1 (1/1). Ceci nous amène à vous parler d'un petit truc bien utile quand on veut faire de jolis en-têtes à l'écran. Nous vous avons parlé longuement du séparateur de sous-article US. Le terminal l'utilise dans un autre cas que pour positionner le curseur. Si vous faites ESC 6/1, il s'en servira pour vous retourner la valeur des coordonnées qui suivent US. Ainsi, si vous avez déplacé votre curseur jusqu'à la cinquième colonne de la cinquième rangée, la séquence ESC 6/1 matérialisera à l'écran les codes 4/5 5/9.

### Les autres fonctions

Le code SP (2/10) déplace le curseur d'un espace vers la droite. Il inhibe alors les attributs de visualisation de zone précédente et laisse apparaître la couleur du fond. Mais, pour les titres, par exemple, il est préférable de laisser les espaces de la même couleur que les caractères. Est-ce à dire que vous devrez composer une séquence ESCAPE pour chaque espace à colorier ? Que nenni ! La fonction d'oblitération DEL (7/F) s'en chargera à votre

## Il est aisé de réaliser un dispositif auto-répondre à l'aide d'un minitel.

place. Ainsi, si nous voulons obtenir BEATRICE en caractères rouges de taille normale entrecoupés de pavés de la même couleur, nous aurons la séquence suivante :

**1/B** (ESC) **4/1** (rouge) **4/9** (caractères fixes) **4/C** (taille normale) **4/2** (B) **7/F** (espace rouge) **4/5** (E) **7/F** (espace rouge) **4/1** (A) **7/F** (espace rouge) **5/4** (T) **7/F** (espace rouge) **5/2** (R) **7/F** (espace rouge) **4/9** (I) **7/F** (espace rouge) **4/3** (C) **7/F** (espace rouge) **4/5** (E).

La répétition d'un caractère est également possible. Pour ce faire, il faut employer la fonction REP (1/2) x, où x désigne sur 6 bits utiles le nombre de répétitions désirées. Si, par exemple, nous voulons afficher qu'un certain liquide bien connu pour ses bulles est FFFF~~FOU~~, nous aurons la séquence suivante :

**4/6** (F), **1/2** (REP), **4/3** (000011 en binaire), **4/F** (0), **5/5** (U).

C'est bien gentil tout ça, direz-vous, mais 000011 en binaire ça n'a jamais fait 4 mais plutôt 3 ! Tout à fait d'accord ! Mais comme il faut bien avoir un modèle sous la main pour pouvoir le répéter, le code du caractère provoquera le premier affichage, et il faudra donc ôter 1 du nombre de fois que vous désirez voir reproduire le caractère (cqfd).

Enfin, last but not least, votre Minitel possède une sonnerie. Vous l'activerez en employant la fonction BEL (0/7). A la réception de ce code, une sonnerie se déclenche pendant une durée inférieure à une seconde. On peut la prolonger en répétant le code BEL. D'où l'idée complètement loufoque de transmettre par votre Minitel non pas des caractères visualisables mais du morse, en vous servant des caractères « . » et « - », auxquels vous adjoindrez une sonnerie brève et une sonnerie longue (le mieux dans ce cas est la séquence BEL REP 4/2).

Certaines séquences provoquent un fonctionnement particulier de l'écran. Vous pourrez notamment faire passer votre Minitel en mode rouleau en tapant la séquence suivante :

ESC, 3/A, 6/9, 4/3.

Pour repasser en mode plein écran, il suffira de retaper la même séquence en remplaçant juste 6/9 par 6/A.

Deux autres couples de séquences ESCAPE commandent la mise en œuvre et l'arrêt du mode loupe. Ce mode particulier permet de grossir soit la moitié supérieure soit la moi-

tié inférieure de l'écran. Pour passer en mode loupe haut, il faut faire :

ESC, 3/A, 6/9, 4/6

puis

ESC, 3/A, 6/A, 4/6

pour repasser en mode normal.

De même, pour passer en loupe basse, vous utiliserez :

ESC, 3/A, 6/9, 4/7

puis

ESC, 3/A, 6/A, 4/7

pour désactiver cette fonction. Ces diverses séquences appartiennent au Protocole que nous allons étudier maintenant (nous laisserons en effet de côté l'étude du module clavier qui, dans le cadre de ce dossier, n'offre pas un grand intérêt).

### Le protocole

Les quatre organes du Minitel sont pilotés par un logiciel spécifique baptisé Protocole. C'est lui qui, à l'instar d'un chef de gare, assure l'aiguillage des données d'un organe à un autre. Cette position de « grand répartiteur » fait que ses instructions ne sont pas soumises au mécanisme des aiguillages. Seuls les modules le sont, leur mise en relation étant assurée selon qu'ils sont en émission ou en réception. Pour chacun de ces deux modes, les modules disposent d'un code distinct. Résumons ceux-ci rapidement :

Module	Emission	Réception
Ecran	5/0	5/8
Clavier	5/1	5/9
Modem	5/2	5/A
Prise	5/3	5/B

Ce qu'il faut également savoir, c'est que tous les ordres du langage Protocole nécessitent l'emploi de séquences Escape. Ces séquences sont analysées uniquement par le logiciel. Il y a au total 29 instructions dans le langage Protocole.

Notons toutefois que certaines séquences de type SEP, X sont également émises par le Protocole. Une fois la commande reçue, le Protocole garantit sa reconnaissance en envoyant une séquence d'acquiescement, un peu à la manière de l'ACKnowledge transmis par une imprimante à l'ordinateur.

A la mise sous tension du terminal, ce dernier est en aiguillage standard, c'est-à-dire qu'aucun mode particulier n'est activé. A ce stade, une séquence SEP, 5/3 est diffusée vers tous les récepteurs.

Lorsque le terminal change d'état,

il envoie une séquence spécifique vers tous les récepteurs.

SEP, 5/0 marque le changement d'état résultant de la connexion du terminal à un serveur.

SEP, 5/1 est le message qui est diffusé quand le modem change de vitesse.

Lors de la détection de la portuse, c'est la séquence SEP, 5/3 qui est expédiée !

Enfin, quand la prise péri-informatique est en service, le terminal diffuse la séquence SEP, 5/4.

### Le terminal dans tous ses états

Nous savons que notre Minitel fonctionne en mode local lors de sa mise sous tension. Mais qu'en est-il quand il passe en mode terminal ? Son état physique, lors de la demande, se traduit par la séquence suivante : ESC, 3/9, 7/0. La réponse à cette mise en état est donnée par la séquence suivante : ESC, 3/A, 7/1, octet de status terminal.

Ici, deux remarques s'imposent :

- Tout d'abord, il est indispensable de noter que les séquences Protocole peuvent comporter un ou plusieurs octets après le code Escape. Les créateurs de Protocole ont donc décidé de baptiser PRO1 les séquences n'ayant qu'un octet comme argument, PRO2 celles qui en disposent de deux, et PRO3 celles qui nécessitent un triptyque. En fait, le nombre d'arguments est donné par la valeur de l'octet qui suit le code Escape. 3/9 signale un seul octet, 3/A deux octets, 3/B trois octets. Ce qui, résumé, sous forme de tableau nous donnera ceci :

PRO1, X = ESC, 3/9, X  
PRO2, X, Y = ESC, 3/A, X, Y  
PRO3, X, Y, Z = ESC, 3/B, X, Y, Z

- Ensuite, il nous faut préciser de quoi est constitué l'octet de status terminal. Celui-ci contient cinq bits significatifs. Il est constitué de la façon suivante :

P-1-0-PT-DP-0-VM-EC

P correspondant au bit de parité, PT indiquant l'état du fil du même nom sur la prise péri-informatique, le bit étant à 1 lorsque le périphérique est en transmission,

DP étant le bit qui signale la détection ou la non-détection de la portuse (1 signale la connexion), VM indiquant la vitesse du modem dans le sens « Paris-Provence », en-

tendez par là le sens serveur vers terminal (1 indique une vitesse de 1 200 bauds), EC signalant l'état du terminal lors de la connexion (la mise à 1 de ce bit sert dans le cas de retournement du modem).

Moralité : on ne dispose de pas moins de cinq indications par simple consultation de cet octet !

Mais, vous vous en doutez, il existe d'autres instructions, ne serait-ce que pour faire passer le modem de son bouclage permanent, lorsque le terminal est en mode local, en émission-réception dès qu'il y a détection d'une porteuse. On voit donc combien il est facile de réaliser un dispositif autorépondeur en se servant d'un tout petit programme. Pour ce faire, il suffit en effet de concevoir une fonction de scrutation permanente de l'octet de status qui boucle tant que le bit DP ne passe pas à 1. Dès le passage à 1, il faudra envoyer la séquence Escape correspondant à la connexion du modem, c'est-à-dire PRO1, 6/8. La déconnexion sera réalisée par la séquence PRO1, 6/7.

## De module en module

Nous avons vu tout au début de cette étude comment étaient configurés les aiguillages entre les organes du Minitel (cf. fig. 1). Mais ce qu'il faut savoir, c'est que le serveur ou le périphérique connecté au terminal peut engendrer des commandes Protocole qui permettent de relier un module émetteur à plusieurs modules récepteurs. Voici les aiguillages possibles à partir d'un module émetteur.

récepteur	→	émetteurs			
prise	→	modem	→	clavier	
modem	→	prise	→	clavier	
clavier					
écran	→	prise	→	modem	→
					clavier

Ces aiguillages sont effectués en utilisant une séquence ayant cette forme :

PRO3, COMMANDE D'AIGUILLAGE, CODE RECEPTION DU MODULE RECEPTEUR, CODE EMISSION DU MODULE EMETTEUR.

Les codes des commandes d'aiguillage sont les suivants :

6/1 : ON, la liaison entre le module émetteur et le module récepteur est établie.

6/0 : OFF, la liaison est déconnectée.

Ainsi, par exemple, si nous voulons mettre en relation la prise péri-informatique et le modem, nous aurons la séquence suivante :

PRO3, 6/0, 5/A, 5/3.

Il est également possible de bloquer temporairement les communications à destination d'un module. C'est généralement le cas lorsque le module émetteur et le module récepteur sont identiques.

Tous ceux d'entre vous qui ont, peu ou prou, touché aux réseaux, savent bien qu'une des principales tâches du serveur consiste à régler les accès au disque. C'est un peu la même chose qui se passe ici avec les aiguillages. Chaque module peut faire une demande de status à n'importe quel moment afin de savoir soit l'état d'un module en tant qu'émetteur ou récepteur, soit l'état général des aiguillages. Bien évidemment, toute demande obtient réponse. Pour connaître l'état d'un module récepteur ou émetteur, on emploie la séquence :

PRO2, 6/2 (TO), CODE RECEPTION OU EMISSION

La réponse (encore baptisée séquence d'acquiescement) se présente sous la forme suivante :

PRO3, 6/3 (FROM), CODE RECEPTION OU EMISSION DU MODULE, SUIVI DE L'OCTET DU STATUS D'AIGUILLAGE RECEPTEUR OU EMETTEUR DU MODULE.

Admettons que le module écran veuille connaître l'état du clavier, il enverra :

PRO2, TO, 5/1.

Ce dernier, une fois sollicité, lui expédiera son acquiescement sous cette forme :

PRO3, FROM, 5/1, 11000010.

Tout comme l'octet de status terminal, l'octet de status associé à un module véhicule une foule d'informations.

Le bit de poids fort (bit 7) est le bit de parité, le suivant est à 1, les bits 4 et 5 ne servent pas et sont maintenus à 0, le bit 3 nous renseigne sur une possible liaison avec la prise péri-informatique (1, la liaison est établie), le 2 nous renseigne sur le modem, le 1 sur le clavier, et le 0 sur l'écran.

Lorsqu'il reçoit une instruction

d'aiguillage, le Protocole l'exécute et répond par un acquiescement, c'est-à-dire qu'il fournit le nouveau status récepteur du module récepteur concerné après l'exécution de la commande.

Le format de l'acquiescement est : PRO3, FROM, CODE RECEPTION DU MODULE RECEPTEUR, STATUS D'AIGUILLAGE DU MODULE RECEPTEUR CONCERNE.

Cet acquiescement est systématiquement envoyé au module initiateur de la commande d'aiguillage.

Mais est-il possible d'envoyer ces acquiescements à d'autres modules qui désireraient savoir ce qui se passe ? Tout à fait ! On utilise pour ce faire des commandes de diffusion. Le format de celles-ci est le suivant :

DIFFUSION (6/5) : le module, désigné par le code réception, recevra toutes les séquences d'acquiescement quel que soit le module initiateur ;

NON DIFFUSION (6/4) : le module, désigné par le code réception, ne recevra que les séquences d'acquiescement dont il est l'initiateur. Donc, il faut non pas prendre cet ordre comme une pure et simple non diffusion mais plutôt comme un ordre de diffusion restreinte.

Quand nous retrouvons-nous en diffusion restreinte ? Dans plusieurs cas. Tout d'abord à la mise sous tension du terminal, ensuite lors d'une connexion ou d'une déconnexion du modem, ou encore lors d'une fin de session sur la prise en mode local.

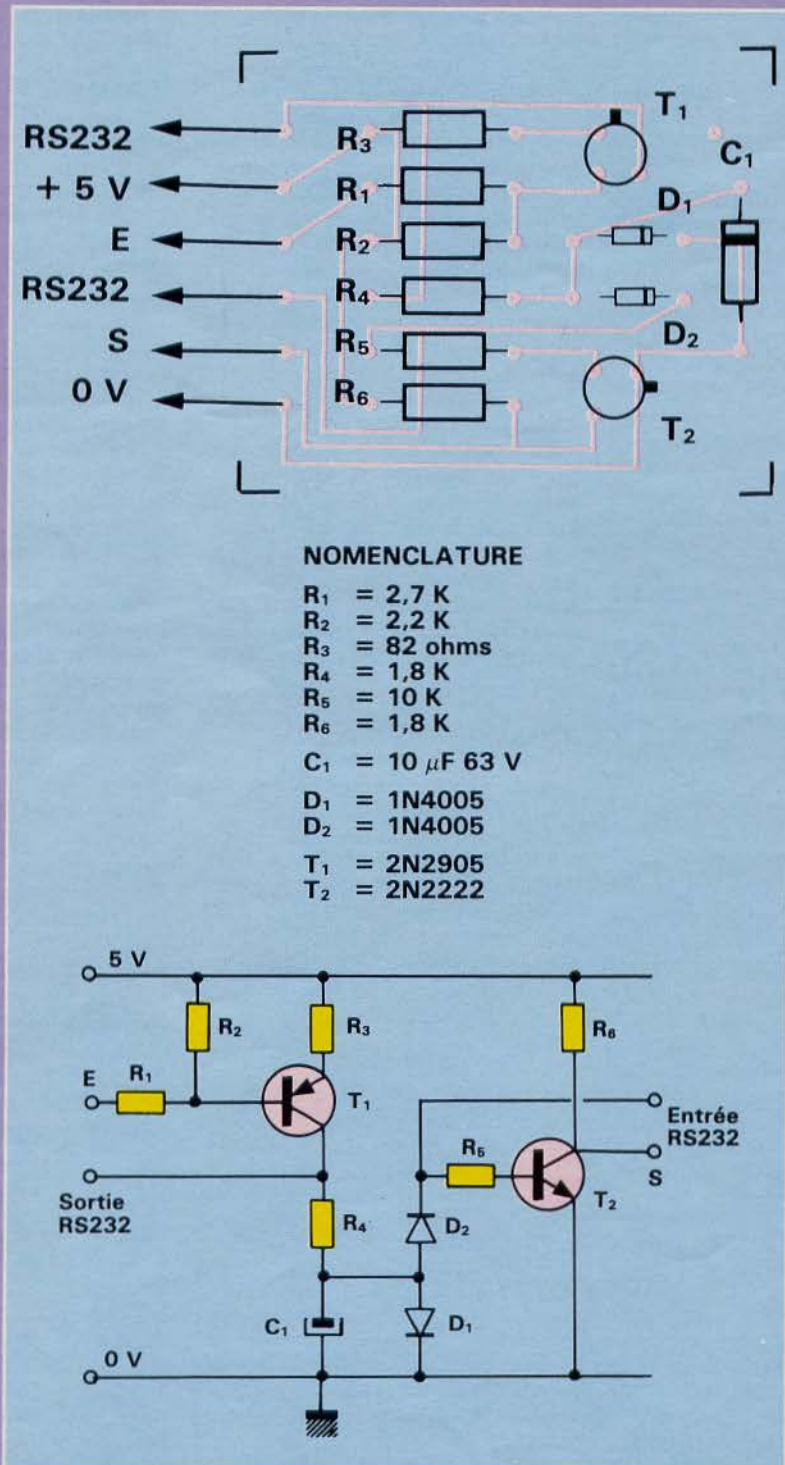
Là encore, on a un octet de statut Protocole ; seuls les deux premiers bits de poids faible sont ici importants. Si le premier bit est à 0, c'est que les acquiescements sont diffusés vers le modem. Si le second bit est à 0, c'est que les acquiescements sont envoyés vers la prise.

Nous allons passer dans quelques instants à l'étude de la prise péri-informatique, ce qui nous permettra de vous proposer un petit montage, très simple à réaliser, et qui vous autorisera de faire communiquer votre Minitel avec la sortie RS 232 C de votre ordinateur. Pour toutes les autres commandes Protocole, reportez-vous à la figure 10.

## La prise péri-informatique

Cette fameuse prise péri-informatique est, à notre sens, un des organes essentiels pour une bonne exploita-

Une adaptation RS 232 C de la prise péri-informatique connecte votre ordinateur au minitel.



**NOMENCLATURE**

- R<sub>1</sub> = 2,7 K
- R<sub>2</sub> = 2,2 K
- R<sub>3</sub> = 82 ohms
- R<sub>4</sub> = 1,8 K
- R<sub>5</sub> = 10 K
- R<sub>6</sub> = 1,8 K
- C<sub>1</sub> = 10 µF 63 V
- D<sub>1</sub> = 1N4005
- D<sub>2</sub> = 1N4005
- T<sub>1</sub> = 2N2905
- T<sub>2</sub> = 2N2222

Examinons plus en détail ces différents signaux. Les signaux TX et RX ont 7 bits de données plus un bit de parité paire.

Le signal PT indique l'occupation du réseau des périphériques raccordés au Minitel. Le signal TP n'a guère d'importance dans la mesure où il est activé sur toutes les versions du Minitel. Guère d'importance pour l'instant, mais nous savons de source sûre que ce fil sera employé sur les prochaines versions pour alimenter des périphériques à faible consommation (quelques watts sous 5 V).

La vitesse des échanges est également programmable grâce aux signaux RX et TX. On peut aussi changer la vitesse et passer en 1200-1200 et 300-300. Pour ce faire, on utilise la séquence suivante :

ESC, 3/A, 6/B, OCTET VITESSE

Cet octet a cette configuration :  
PT-1-E2-E1-E0-R2-R1-R0

On remarque tout de suite deux champs de trois bits, l'un dénommé E (pour émission), l'autre R (pour réception). Voici les valeurs que prennent ces champs :

- 001 = 75 bauds
- 010 = 300 bauds
- 100 = 1 200 bauds

Toutefois, ces échanges ne sont validés que lorsque la prise a reçu une séquence d'acquittement de ce type :  
ESC, 3/A, 7/5.

Encore un mot du signal PT. Celui-ci conserve les aiguillages et les vitesses d'échanges entre les périphériques lorsqu'il passe à 0. Lorsqu'il passe à 1, le Minitel revient en mode standard. Il y aurait encore bien d'autres choses à dire sur le Minitel, mais nous vous laissons pour l'instant le soin de les découvrir par vous-même. Voici donc, en guise de « dessert », le montage promis.

**Conclusion : un adaptateur Minitel RS232**

Ce circuit, très simple à réaliser, permet de relier la fiche péri-informatique du Minitel à la prise RS 232 de n'importe quel micro-ordinateur. Nous ne sommes pas encore en mesure de vous fournir le logiciel permettant d'exploiter cette liaison. Espérons que des lecteurs actifs nous apporteront rapidement une solution (langage machine Forth ou tout langage relativement « rapide »).

M. ROUSSEAU

Schéma de principe, liste des composants et circuit imprimé d'un adaptateur RS 232 C / prise péri-informatique.

- la fiche 1 assure la réception des données par le terminal (signal RX) ;
- la fiche 2 est la masse ;
- la fiche 3 assure l'émission des données par le terminal (signal TX) ;
- la fiche 4 s'occupe du périphérique en transmission (signal PT) ;
- enfin, la fiche 5 signale si le terminal est prêt (signal TP).

# PROGRAMME

## JEU

Vous aimez les jeux de réflexion, mais malheureusement il vous est toujours difficile de trouver un adversaire. Si vous possédez un Oric Atmos, vous êtes maintenant sauvé : après avoir chargé ce programme, votre ordinateur favori deviendra un adversaire toujours disponible pour vous affronter au Tiouk-Tiouk.

de J.-C. RIAT

Ordinateur :

Oric Atmos

Langage :

Basic

Vous vous demandez sans doute ce qui se cache derrière ce nom exotique. Les règles de ce jeu de pure réflexion sont très simples. Au départ, sur un échiquier 8 x 8 sont disposés sur la première rangée 8 pions blancs et sur la trentième rangée 8 pions noirs. Chaque adversaire dispose des huit pions d'une même couleur et les blancs commencent toujours. Chacun des deux joueurs, à tour de rôle, déplace l'un de ses pions d'un nombre de cases quelconque, en avant ou en arrière, mais dans une seule colonne. Il doit se poser sur une case libre et ne peut pas sauter par-dessus le pion adverse situé dans la même colonne que le sien. Pour gagner, il vous faudra (et même, il vous suffira !) de bloquer votre adversaire.

La simplicité des règles vous a sans doute déjà donné envie de jouer ! Il vous faut donc savoir comment utiliser le programme. Une fois chargé, lancez son exécution par « RUN ». Après vous avoir salué et s'être présenté, Oric vous posera différentes questions auxquelles il vous faudra répondre avant de commencer. Si vous avez réussi à trouver un compagnon de jeu et que vous optez pour la solution de jouer contre un adversaire humain, Oric se contentera de gérer sur l'écran la représentation du jeu tout en vérifiant la validité de vos coups. En revanche, si vous désirez vous mesurer à lui, il vous faudra encore, avant de pouvoir l'affronter, décider de son niveau de jeu. Un conseil : si vous débutez, ne choisissez pas les niveaux 8 ou

# TIOUK-TIOUK



9, sinon vos échecs répétés face à la « machine » risqueraient de vous démoraliser ! Si au contraire, vous battez facilement Oric aux niveaux faibles, essayez donc le niveau 9 avant de vous croire imbattable ! Pour entrer vos coups, il vous suffit d'indiquer la case d'arrivée du pion. N'ayez pas peur de bloquer le programme en entrant un coup impossible car Oric les vérifie tous avant de les accepter (on n'est jamais assez prudent face aux humains !). Vous pouvez également vous faire aider en entrant « CONSEIL », sur quoi Oric vous indiquera le coup qu'il jouerait s'il était à votre place.

Enfin, si vous jugez votre situation désespérée, en entrant « ABANDON », vous mettrez fin à vos souffrances.

### L'algorithme

Si, fasciné, vous désirez laisser tout son mystère à ce jeu, ne lisez pas ces lignes car elles révèlent le secret. En effet, le

programme a un « truc » pour gagner : il sait évaluer si une position est perdante ou gagnante. Pour cela, il traduit en binaire le nombre de cases libres entre les pions pour chaque colonne, ce qui donne le tableau B (7,2), (la première composante étant celle du numéro de la colonne considérée : 0 pour A, 1 pour B, etc.). Par exemple, si dans la colonne A, les pions sont situés sur les rangées 2 et 6, on aura : B(0,0) = 0 ; B(0,1) = 1 ; (0,2) = 1.

Ensuite, dans le sous-programme de 500 à 600, il additionne ces huit nombres binaires selon une méthode particulière : il pose l'addition comme à l'école primaire et applique les règles suivantes :  $0 + 0 = 0$  ;  $0 + 1 = 1$  ;  $1 + 0 = 1$  ;  $1 + 1 = 0$

Si le résultat final, contenu dans L est nul, celui dont c'est le tour de jouer est perdant, sinon il est gagnant.

Avec ces quelques explications, la méthode de jeu utilisée par Oric devient alors très sim-

### LISTE DES VARIABLES

- A variable de boucle et utilisation temporaire
- B utilisation temporaire
- C C=1, vous jouez contre Oric  
C=2, vous jouez contre un autre adversaire humain
- D niveau de Oric (de 0 à 9)
- E E=1, Oric ou l'autre adversaire humain (fixé d'après C) commence  
E=-1, vous commencez
- F F=1, c'est les blancs qui jouent  
F=-1, c'est les noirs qui jouent
- G numéro de colonne du coup contenu dans E\$ (A donne 0, B donne 1, etc.)
- H numéro de ligne du coup contenu dans E\$ auquel on retranche 1
- I numéro du coup en cours
- J compteur utilisé quand Oric joue un coup au hasard (souvent perdant)
- K compteur utilisé pour regarder si le jeu est presque fini (les pions de chaque colonne étant alors contigus) : dans ce cas K = 0
- L résultat de l'évaluation de la position (si L = 0, celui dont c'est le tour de jouer est perdant, sinon il est gagnant)
- M utilisation temporaire quand Oric joue un coup gagnant
- A\$ réponses aux diverses questions
- B\$ prénom du joueur (4 lettres)
- C\$ message à afficher en haute résolution
- D\$ prénom de l'autre adversaire humain (seulement si C = 2)
- E\$ coups joués par les deux adversaires

ple à comprendre. Quand c'est à son tour de jouer, il choisit selon le niveau s'il cherche à jouer un coup gagnant ou s'il

# PROGRAMME

## JEU

### STRUCTURE DU PROGRAMME

- 500 à 600 sous-programme évaluant la position comme expliqué dans l'article qui précède
- 1000 à 1060 mouvements réels des pions : la ligne 1040 efface l'ancienne position et la ligne 1060 affiche la nouvelle
- 1100 à 1160 sous-programme étudiant si le jeu est presque fini, c'est-à-dire si les pions de chaque colonne se trouvent sur des cases contiguës. Dans ce cas celui dont c'est le tour de jouer a pratiquement perdu
- 1200 à 1300 modification des tableaux A (2,7) et B (7,2)
- 1500 à 1590 sous-programme d'affichage des coups à droite de l'écran
- 1700 à 1840 sous-programme regardant si c'est fini, c'est-à-dire si celui dont c'est le tour de jouer a tous ses pions bloqués
- 2000 à 2350 dessin du support du jeu : le joueur humain a toujours les pions de la couleur qu'il a choisie face à lui, c'est-à-dire en bas de l'écran. De plus, au début, les pions blancs sont situés sur la rangée 1
- 2500 à 2580 sous-programme d'affichage sur l'écran haute résolution. En faisant GOSUB 2500, on affiche le message contenu dans C\$ à la position 12,190. Pour l'afficher à la position x, y quelconque, il suffit de faire CURSET Y, Y, 0 puis GOSUB 2540
- 2600 à 2660 sous-programme d'effaçage sur l'écran haute résolution des messages situés à la position 12,190
- 3000 à 3330 règles du jeu
- 3500 à 3720 le joueur effectue ses différents choix
- 4000 à 4120 initialisation des variables et des tableaux
- 4500 à 4530 orientation du programme : si l'Atmos joue et doit commencer, le programme va en 6000
- 5000 à 5840 introduction du coup humain et vérification de sa validité
- 6000 à 6070 fin gagnante pour Oric : si avant le coup de l'humain, les pions de chaque colonne se trouvaient sur des cases contiguës, l'Oric répond au coup de l'humain par un coup qui le remplace dans une telle position
- 7000 à 7620 Oric joue un coup au hasard (souvent perdant)
- 7100 à 7140 Oric recule son pion sur la colonne choisie dans G
- 7500 à 7620 Oric avance son pion sur la colonne choisie dans G
- 8000 à 8280 Oric joue un coup gagnant (il est sûr que ce coup existe, car il a évalué sa position gagnante dans le sous-programme de 500 à 600)

### DENOMINATION DES TABLEAUX

- A(2,7) : représentation principale de la position
- A(0,x) : numéro de ligne du pion blanc de la colonne x
- A(1,x) : numéro de ligne du pion noir de la colonne x
- A(2,x) : nombre de cases vides entre les pions ad-

- verses de la colonne x
- B(7,2) : « traduction binaire de la position »
- C(2) : utilisé pour l'évaluation de la position
- D(2) : utilisation temporaire pour stocker une ligne de B(7,2) quand Oric joue un coup gagnant

joue au hasard : si le niveau est n, il a n/9 chance de chercher à jouer un bon coup. S'il doit jouer un coup gagnant, il évalue sa position ; si elle est perdante, il joue au hasard car il n'existe pas de coup gagnant. Dans le cas contraire, il essaie tous les coups possibles en évaluant à chaque fois la position jusqu'à

ce qu'il en trouve une perdante pour son adversaire. Ce coup sera alors gagnant pour lui !

En fait, le programme, pour rendre le jeu encore plus attrayant, ajoute à cette méthode générale quelques petits compléments que découvriront sans doute ceux qui l'étudieront en détail. ■

```
10 REM TIOUK-TIOUK
20 REM
30 REM RIAT JEAN-CHRISTOPHE
40 REM
50 REM 10/07/84
60 REM
70 TEXT
80 CLS
90 PAPER 2:INK 0
100 GOTO 3000
500 REM
510 REM EVALUATION DE LA POSITION
520 REM
530 C(0)=0:C(1)=0:C(2)=0
540 FOR A=0 TO 7
550 C(0)=ABS(C(0)-B(A,0))
560 C(1)=ABS(C(1)-B(A,1))
570 C(2)=ABS(C(2)-B(A,2))
580 NEXT A
590 L=C(0)*4+C(1)*2+C(2)
600 RETURN
1000 REM
1010 REM MOUVEMENTS REELS DES PIONS
1020 REM
1030 CURSET 72+42#E-12#E#G,102-42#E+12#E#(K,5-F/2,G)-1,0
1040 FILL 6,1,54
1050 CURSET 72+E#42-12#E#G,102-42#E+12#E#H,0
1060 FILL 6,1,127+64#F
1100 REM
1110 REM C'EST PRESQUE FINI ?
1120 REM
1130 K=0
1140 FOR A=0 TO 7
1150 IF R(2,A)=0 THEN K=K+1
1160 NEXT A
1200 REM
1210 REM MODIFICATIONS DES TABLEUX
1220 REM
1230 R(,5-F/2,G)=H+1
1250 R(2,G)=R(1,G)-R(0,G)-1
1260 A=INT(R(2,G)/2)
1270 B(G,2)=R(2,G)-2#A
1280 B=INT(A/2)
1290 B(G,1)=A-2#B
1300 B(G,0)=B
1500 REM
1510 REM AFFICHAGE DES COUPS
1520 REM
1530 I=1-(F=1)
1540 IF F=1 THEN C$=STR$(I)+") "+E# ELSE C$=E#
1550 IF 1/16<>INT(1/16) OR F=-1 THEN 1580
1560 CURSET 148,46,0
1570 FILL 128,13,64
1580 CURSET 181-27#F+6*(I#F>9),46+B*(1-INT(1/16))*16,0
1590 GOSUB 2540
1700 REM
1710 REM REGARDE SI C'EST FINI
1720 REM
1730 B=0
1740 FOR A=0 TO 7
1750 IF R(0,A)=7 THEN B=B+1
1760 IF R(1,A)=2 THEN B=B-1
1770 NEXT A
1780 IF ABS(B)>>8 THEN RETURN
1790 IF C=1 AND B#E=8 THEN C$=" Desole, mais J'ai gagne !"
1800 IF C=2 AND E#B=8 THEN C$=" Bravo "+D$+"! Vous avez gagne."
1810 IF E#B=-8 THEN C$=" Bravo "+B$+"! Vous avez gagne."
1820 GOSUB 2500
1830 INPUT " Desirez-vous faire une autre partie " :A#
1840 IF A#="OUI" THEN RUN ELSE END
2000 REM
2010 REM DESSIN DE L'ECHIQUIER
2020 REM
2030 HIRE
2040 PAPER 2:INK 0
2050 C$="TIOUK-TIOUK"
2060 CURSET 90,6,0
2070 GOSUB 2540
2080 CURSET 90,14,0
2084 CURSET 90,14,0
2085 FILL 2,11,192
2090 FOR A=0 TO 8
2100 B=26+12#A
2110 CURSET B,56,1
2120 DRAW 0,97,1
```

Listing du programme.

# PROGRAMME

## JEU

```

2130 CURSET B+1,56,1
2140 DRAW 0,97,1
2150 CURSET 26,B+30,1
2160 DRAW 97,0,1
2170 CURSET 26,B+31,1
2180 DRAW 97,0,1
2190 NEXT A
2200 FOR A=0 TO 7
2210 B=30+12*A
2220 CURSET B,60,0
2230 FILL 6,1,127+64*XE
2240 CURSET B,144,0
2250 FILL 6,1,127+64*XE
2260 CURSET B,47,0
2270 CHAR 68,5+3,5*XE-A*E,0,1
2280 CURSET B,156,0
2290 CHAR 68,5+3,5*XE-A*E,0,1
2300 CURSET 18,B+30,0
2310 CHAR 52,5-E*3,5+A*E,0,1
2320 CURSET 127,B+30,0
2330 CHAR 52,5-E*3,5+A*E,0,1
2340 NEXT A
2350 RETURN
2500 REM
2510 REM SS PGME D'AFFICHAGE EN HIRES
2520 REM
2530 CURSET 12,190,0
2540 FOR A=1 TO LEN(C#)
2550 CHAR ASC(MID$(C#,A)),0,1
2560 CURMOV 6,0,0
2570 NEXT A
2580 RETURN
2600 REM
2610 REM SS PGME D'EFFACAGE EN HIRES
2620 REM
2630 CLS
2640 CURSET 12,190,0
2650 FILL 8,32,64
2660 RETURN
3000 REM
3010 REM REGLES DU JEU
3020 REM
3030 PRINT:PRINT "BONJOUR !"
3040 PRINT:INPUT "Je m'appelle ORIC et vous (<4 lettres>):"B#
3045 IF LEN(B#)<>4 THEN 3040
3050 PRINT:INPUT "Voulez-vous consulter les regles "Y#
3060 IF A#(">OUI") THEN 3540
3065 E=-1
3070 GOSUB 2000
3080 C#="Regardez Puis Pressez une touche."
3100 GOSUB 2500
3110 GET A#
3120 TEXT
3130 PRINT @ 14,1;"TIOUK-TIOUK"
3140 PRINT TAB(14);"*****"
3150 PRINT
3160 PRINT"REGLES"
3170 PRINT"*****"
3180 PRINT "Vous disposez, a votre choix, des 8 Pions blancs ou noirs.";
3190 PRINT "Je Prends les autres.Les blancs commencent toujours."
3200 PRINT
3210 PRINT "Chaque Joueur a son tour de Place l'un de ses Pions d'un nombre "
3220 PRINT "de cases quelconque, en avant ou en arriere, mais dans une "
3230 PRINT "seule colonne,il doit seposer sur une case libre et ne peut "
3240 PRINT " sauter Par-dessus un autre Pion."
3250 PRINT
3260 PRINT "Pour gagner il suffit (<1> de bloquer l'adversaire."
3270 PRINT
3280 PRINT "Entrez vos coups en n'indiquant que lacase d'arrivee (ex'D4)."
3290 PRINT "Pour avoir une aide, entrez 'CONSEIL'."
3300 PRINT "Pour arreter, entrez 'ABANDON'."
3310 PRINT
3320 PRINT "Pressez une touche Pour continuer."
3330 GET A#
3500 REM
3510 REM CHOIX DU NIVEAU DE JEU
3520 REM
3530 CLS
3540 PRINT:PRINT "Desirez-vous jouer contre:"
3550 PRINT TAB(6);"1) moi"
3560 PRINT TAB(6);"2) un autre adversaire humain"
3570 PRINT:INPUT "Votre choix entre 1 et 2 SVP "J#
3575 IF C#>1 AND C#>2 THEN 3570
3580 IF C#=2 THEN PRINT:INPUT "Comment s'appelle-t-il (< 4 lettres ) "J#
3585 IF C#=2 AND LEN(J#)<>4 THEN GOTO 3580
3590 IF C#=2 THEN 3630
3600 PRINT:PRINT "Quel niveau de force m'accordez-vous (<de 0 a 9 ?"
3610 PRINT:INPUT "Votre choix SVP (Par Pitie un Peu de Generosite )" J#
3615 D=INT(D#)
3620 IF D<0 OR D>9 THEN 3605
3630 PRINT:INPUT "Voulez-vous commencer "Y#
3640 IF A#="OUI" THEN E=-1 ELSE E=1
3650 GOSUB 2000
3660 IF C#=2 THEN 3700
3670 IF E=1 THEN C#="ORIC "+B# ELSE C#="B# "+ORIC
3680 GOTO 3710
3700 IF E=1 THEN C#="D# "+B# ELSE C#="B# "+D#
3710 CURSET 172,30,0
3720 GOSUB 2540
4000 REM
4010 REM INITIALISATION DES VARIABLES
4020 REM
4030 DIM A(2,7)
4040 DIM B(7,2)
4050 FOR A=0 TO 7

```

```

4060 A(B,A)=1:A(1,A)=B:A(2,A)=6
4070 B(A,0)=1:B(A,1)=1:B(A,2)=0
4080 NEXT A
4100 DIM D(2)
4110 DIM D(2)
4120 F=-1
4500 REM
4510 REM ORIENTATION
4520 REM
4530 IF C#E THEN 6000
5000 REM
5010 REM VERIFICATION D'UN COUP HUMAIN
5020 REM
5030 F=-F
5033 A#B#
5036 IF EXP#C=2 THEN A#D#
5040 C#=" A vous, "+A#+"."
5050 GOSUB 2500
5060 INPUT " Votre coup SVP "E#
5065 GOSUB 2600
5070 IF E#="ABANDON" THEN B=-B#GOTO 1790
5080 IF E#<>"CONSEIL" THEN 5500
5090 C#=" Je cherche..."
5100 GOSUB 2500
5110 GOSUB 500
5120 IF L<>0 THEN 8050
5140 C#="E#+" "abandonner !"
5150 PING
5160 GOSUB 2600
5170 GOTO 5810
5500 IF LEN(E#)<>2 THEN 5000
5510 G=ASC(LEFT$(E#,1))-65
5520 H=VAL(RIGHT$(E#,1))-1
5530 IF G<0 OR G>7 OR H<0 OR H>7 THEN 5000
5540 IF A(.5-F/2,G)+H+1 OR (H+1)*F=F#A(.5+F/2,G) THEN 5000
5550 GOSUB 1000
5560 IF C#=2 THEN 5000 ELSE 6000
5000 C#="Coup astucieux mais impossible !"
5010 GOSUB 2500
5020 WAIT 100
5030 GOSUB 2600
5040 GOTO 5040
6000 REM
6010 REM FIN GAGNANTE POUR ORIC
6020 REM
6030 F=-F
6033 C#=" A moi..."
6036 GOSUB 2500
6040 IF K<>0 THEN 7000
6050 H=H-F
6060 E#="LEFT$(E#,1)+CHR$(H+49)
6070 GOTO 7600
7000 REM
7010 REM COUP PERDANT JOUE PAR ORIC
7020 REM
7030 IF RND(1)*9<D OR K>5 THEN 8000
7040 G=INT(RND(1)*8)
7045 G=G+1+B*(G#7)
7050 IF A(2,G)<>0 THEN 7500
7060 J=J+1
7070 IF (J<B AND RND(1)*2) OR A(.5-F/2,G)+4.5-3.5*F THEN 7045
7100 REM
7110 REM ORIC DOIT RECULER
7120 REM
7130 H=-INT(RND(1)*ABS(A(.5-F/2,G)+3.5*F-4.5))-1
7140 GOTO 7540
7500 REM
7510 REM ORIC PEUT AVANCER
7520 REM
7530 H=INT(RND(1)*A(2,G))+1
7540 H=A(.5-F/2,G)+H*F-1
7550 E#="CHR$(65+G)+CHR$(H+49)
7600 GOSUB 2600
7605 PING
7610 GOSUB 1000
7620 GOTO 5000
8000 REM
8010 REM COUP GAGNANT JOUE PAR ORIC
8020 REM
8030 GOSUB 500
8040 IF L#0 THEN 7040
8050 G=INT(RND(1)*8)
8060 G=G+1+B*(G#7)
8080 IF A(2,G)#0 THEN 8060
8090 H=A(.5-F/2,G)
8100 D(0)=B(G,0):D(1)=B(G,1):D(2)=B(G,2)
8110 H=A(.5+F/2,G)-F-1
8120 A(.5-F/2,G)+H+1
8130 A#A(1,G)-A(B,G)-1
8140 B=INT(A/2)
8150 B(G,2)=A-2*B#
8160 A=INT(B/2)
8170 B(G,1)=B-2*A#
8180 B(G,0)=A
8190 GOSUB 500
8200 IF L#0 THEN 8230
8210 H=H-F
8220 IF H<>(M-1) THEN 8120
8230 A(.5-F/2,G)=M
8240 B(G,0)=D(0):B(G,1)=D(1):B(G,2)=D(2)
8250 IF L<>0 THEN 8060
8260 IF E#<>"CONSEIL" THEN 7550
8270 C#="E# "+CHR$(65+G)+CHR$(H+49)
8280 GOTO 5150

```

Listing du programme (suite et fin).