

The Sinclair logo is displayed in a white, stylized, blocky font against a solid black rectangular background.

QL

Concepts

Ce Guide des Notions de Base décrit les notions concernant le SuperBASIC et les ordinateurs QL. Ce guide est à utiliser comme une source d'informations lorsqu'une question se pose au sujet du SuperBASIC ou du QL lui-même dans l'utilisation de l'ordinateur, ou concernant une autre partie du manuel. Ce Lexique de Notions peut apporter alors la réponse. Les Notions (ou Concepts) sont regroupées dans l'ordre alphabétique, sous le mot qui correspond le mieux. Si vous ne trouvez pas le sujet désiré, consultez l'index sur la page suivante : il devrait vous indiquer la bonne page.

Quand un exemple est donné, précédé d'un numéro, c'est un programme complet que vous pouvez introduire et utiliser.

Les exemples non numérotés sont de simples commandes, et il ne convient pas toujours de les introduire isolément dans l'ordinateur.

Les exemples de couleurs composées s'adaptent mal à un écran de téléviseur.

Table des matières

BASIC	3
Canaux.....	3
Cartouche enfichable.....	5
ROM.....	5
Commande directe	6
Communication	7
Comparaison des chaînes de caractères (règles).....	9
Conversions	9
Coordonnées par point.....	11
Pixel	11
Couleur	11
Démarrage	13
Données - variables	14
Ecran.....	15
Exprssions.....	16
Extension périphérique.....	17
Fenêtres.....	19
Fichiers	20
Fonctions et procédures	20
Fonctions mathématiques	21
Gestion des erreurs.....	22
Graphiques.....	24
Graphique de la tortue.....	26
Horloge	26
Identifiant.....	27
Interruption provoquées.....	27
Jeux de caractères	28
Touches du clavier	28
Manette de jeu.....	33
Mémoire centrale.....	34
Microlecteurs	35
Moniteur	37
Mots-clés.....	38
Opérateurs	39
Ordes du SuperBASIC	40
Périphériques	40
Programme	43
QDOS	44
Répétition	48
Réseaux.....	48
Son.....	50
Syntaxe - Définitions	52
Tableaux	53
Tableaux de chaînes – Variables de chaînes	53
Tableaux (découpage des).....	54

BASIC

Le SuperBASIC inclut la plupart des fonctions, procédures et constructions que l'on trouve dans les autres langages BASIC. La plupart de ces fonctions sont inutiles en SuperBASIC, mais on les conserve pour des raisons de compatibilité avec d'autres langages.

GOTO	utilisez IF, REPEAT, etc
GOSUB	utilisez DEFINE PROCEDURE
ON GOTO	utilisez SELECT
ON GOSUB	utilisez SELECT

Certaines commandes de BASIC sont incluses en SuperBASIC et correspondent à un emploi particulier :

READ
DATA
RESTORE
REMARK
PEEK
POKE
LIST
NEW RUN STOP
CLEAR

(Voir signification dans la partie « Lexique » des Mots Clés)

Certaines commandes n'existent pas. On peut les obtenir au moyen d'une fonction plus générale. Par exemple il n'y a pas de **LPRINT** ou **LLIST** en SuperBASIC, mais on peut obtenir la commande à l'imprimante en ouvrant le canal approprié par **OPEN # n** et en utilisant **PRINT** ou **LIST**.

LPRINT	utilisez PRINT #n,...
LLIST	utilisez LIST#n (,N° de ligne TO N° de ligne)
VAL	non nécessaire en SuperBASIC
STR\$	non nécessaire en SuperBASIC
IN	non applicable au Microprocesseur 68008
OUT	non applicable au Microprocesseur 68008

CANAUX

CHANNELS

Un canal est le moyen par lequel des données peuvent entrer ou sortir dans un dispositif QL. Avant d'être utilisé, il faut ouvrir le canal par la commande **OPEN**. Certains canaux doivent toujours être utilisés, ce sont les canaux réservés par défaut : ils permettent de communiquer avec le QL par le clavier et l'écran. Quand un canal n'est plus utile, il peut être fermé par la commande **CLOSE**.

Un canal s'identifie par un numéro. Quand un canal est ouvert, un dispositif se branche sur le canal numéroté, et le canal est initialisé. Par la suite, le canal est désigné seulement par son numéro.

Par exemple :

```
OPEN # 5,ser1
```

va connecter le port * Ser1 au canal 5. Pour fermer un canal il suffit d'indiquer le numéro du canal, par exemple :

```
CLOSE # 5
```

Pour ouvrir un canal il faut que le DRIVER (pilote) du périphérique puisse être activé. Pour cela plusieurs procédés sont possibles. Prenons le cas du réseau : ce dernier nécessite un numéro de station. Cette indication complémentaire est jointe au nom du dispositif, et utilisé avec la commande **OPEN** comme un paramètre (Cf. Les notions DISPOSITIF et EXTENSION PERIPHERIQUE)

Des données peuvent être transférées sur un canal par la commande **PRINT**. C'est la même manipulation qui permet de faire paraître ces données sur l'écran. **PRINT** transfère à un canal réservé par défaut.

Par exemple :

```
10 OPEN # 5, MDV1_FICHER_ESSAI
20 PRINT # 5, "Ce texte est dans le fichier essai"
30 CLOSE # 5
```

permettra d'écrire le texte : "Ce texte est dans le fichier essai". Il faut fermer (**CLOSE**) le fichier après que les instructions de sortie soient toutes effectuées, pour être sûr que les données soient écrites.

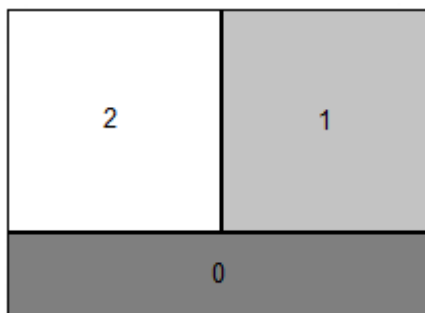
De la même façon, des données peuvent être extraites d'un fichier par la commande **INPUT** et des données peuvent être extraites d'un canal caractère par caractère, en utilisant la commande **INKEY\$**.

Un canal peut être ouvert comme canal de la console (Ecran + Clavier) La sortie se fait directement sur une fenêtre de l'écran QL et l'entrée se fait par le clavier. Quand on ouvre un canal de console, il faut préciser la dimension et la forme de la fenêtre. Si on travaille avec plus d'une fenêtre, il est possible de le faire en même temps. On choisit le canal voulu par la commande **CTRL C** pour mettre en attente les canaux. Le curseur sur la fenêtre du canal choisi s'éclairera.

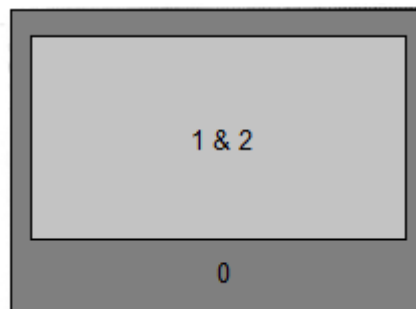
Le QL a trois canaux réservés par défaut qui s'ouvrent automatiquement. A chacun est attribuée une fenêtre sur l'écran du QL.

Canal 0 Canal de commandes et d'erreurs
 Canal 1 Canal de sorties et de graphiques
 Canal 2

Canal des listes de programmes



Moniteur



Télévision

Commandes

Fonctions

OPEN	Ouvre un canal pour une Entrée-Sortie
CLOSE	Ferme un canal déjà ouvert
PRINT	Sortie sur un canal
INPUT	Entrée à partir d'un canal
INKEY\$	Entrée d'un caractère à partir d'un canal

CARTOUCHE ENFICHABLE

ROM

CARTRIDGE SLOT

La cartouche QL ROM permet le chargement de logiciels dans le système du QL. La cartouche ROM peut contenir des logiciels qui modifient le Super-BASIC. La cartouche peut contenir :

1. Des logiciels en remplacement du SuperBASIC, par exemple :
 - assembleurs
 - compilateurs
 - debuggers
 - logiciels d'application
 - etc.
2. Des logiciels pour enrichir le SuperBASIC, par exemple :
 - procédures spéciales
 - etc.

Il n'est pas possible d'utiliser la cartouche ROM ZX sur le QL.

Points de sortie :

—	a	1	b	VDD
A12	a	2	b	A14
A7	a	3	b	A13
A6	a	4	b	A8
A5	a	5	b	A9
SLOT	a	6	b	SLOT
A4	a	7	b	A11
A3	a	8	b	ROMOEH
A2	a	9	b	A10
A1	a	10	b	A15
A0	a	11	b	D7
D0	a	12	b	D6
D1	a	13	b	D5
D2	a	14	b	D4
GND	a	15	b	D3

Signal	Fonction
A0..A15	Lignes d'adresses
D0..D7	Lignes de données
ROMOEH	Habilitation des sorties ROM
VDD	5V
GND	Terre

Attention !

Ne jamais brancher ou débrancher une cartouche ROM tant que le QL est sous tension.

COMMANDE DIRECTE

DIRECT COMMAND

Le SuperBASIC établit une distinction entre une instruction précédée d'un numéro de ligne et une instruction sans numéro de ligne.

Sans numéro de ligne, l'instruction est une **commande directe** et est exécutée directement par **l'interprète de commande** du SuperBASIC.

Par exemple, **RUN** est tapé, et le résultat immédiat est que le programme commence à fonctionner. Si une instruction est tapée avec un numéro de ligne, alors la syntaxe de la ligne est contrôlée et toute erreur est indiquée. Une ligne correcte entre en programme SuperBASIC et est stockée. Ces instructions constituent un programme SuperBASIC qui ne seront exécutés que si le programme commence par la commande **RUN** ou **GOTO(n)**.

Toutes les instructions en SuperBASIC n'ont pas forcément d'effet lorsqu'elles sont entrées comme commande directe. Par exemple : **END FOR, END DEFine...**

Commande	Fonction
RUN	exécute un programme
NEW	efface programme et variables
CLEAR	efface variables
FORMAT	formate une cartouche
SAVE	sauve un programme
LOAD	charge (load) un programme
MERGE	fusionne un programme en mémoire
LRUN	charge et exécute un programme
LIST	liste un programme
RENUM	renumérote un programme
DLINE	détruit les lignes du programme
PRINT	imprime des données
EXEC	exécute une tâche
EDIT	édite un programme
AUTO	entre un programme
COPY	copie les données

COMMUNICATIONS

RS-232-C

Le QL a deux ports séries (appelés SER1 et SER2) qui permettent la connexion à une installation qui utilise des communications EIA de type RS-232-C ou un modèle compatible.

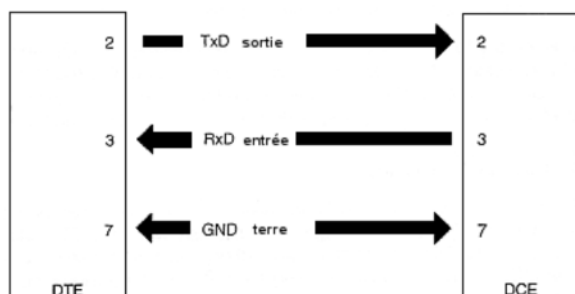
Le standard RS-232-C servait au début à désigner des ordinateurs capables d'envoyer et de recevoir des données par téléphone par l'usage d'un Modem. Actuellement, il désigne des ordinateurs reliés l'un à l'autre par divers équipements de périphériques : imprimantes, traceurs de courbes, etc...

Etant donné que le RS-232-C se présente sous une infinité de formes d'équipements différents, il est extrêmement difficile, même pour un spécialiste, d'établir une connexion entre deux équipements de modèle RS-232-C. Nous voulons ici répondre aux principaux problèmes de base que vous pouvez rencontrer.

Le RS-232-C correspond à deux types d'équipements :

- le DTE (Data Terminal Equipment) : pour transmettre des données à un Terminal.
- le DCE (Data Communication Equipment) : pour des échanges de données.

Normalement le Terminal (DTE) et le Modem (DCE) devraient avoir le même type de connecteur.



Le schéma ci-dessus montre que le DTE transmet des données sur le "pin" (élément de broche) 2 tandis que le DCE reçoit les données sur son "pin" 2 (appelé pin de transmission). De la même manière, le DTE reçoit les données sur le "pin" 3 (appelé pin de réception). C'est pour le moins ambigu. Le problème est que certains périphériques peuvent avoir la configuration de DCE ou de DTE.

Malheureusement, il arrive que certains constructeurs décident que leur ordinateur devrait avoir la configuration DCE, alors que d'autres préfèrent donner aux mêmes ordinateurs la configuration DTE. Dès lors il est très difficile de donner à chaque port de série sa configuration.

SER1 sur le QL est configuré DCE alors que SER2 l'est en DTE. Dès lors il devrait être possible de connecter au moins l'un des ports de série à un périphérique donné si l'un ou l'autre port est correctement branché. Le pin de sortie des ports de série est donné ci-dessous. Sinclair Research Limited peut vous fournir un câble pour connecter le QL à un connecteur à 25 broches de type D.

Note : En cas de difficulté, essayez successivement les deux ports SER1 et SER2.

SER1			SER2		
Pin	Nom	Fonction	Pin	Nom	Fonction
1	GND	Signal terre	1	GND	signal terre
2	TxD	Input	2	TxD	Output
3	RxD	Output	3	RxD	Input
4	DTR	Prêt input	4	DTR	Prêt output
5	CTS	Prêt output	5	CTS	Prêt input
6	-	+12V	6	-	+12V

TxD	Transmet données	DTR	Terminal prêt aux données
RxD	Reçoit données	CTS	Libre pour envoyer

Quand l'installation a été connectée au port "correct", la vitesse de transmission des données en "baud" doit être la même pour le QL et l'installation connectée. Le QL peut être réglé pour travailler à :

75
300
600
1200
2400
4800
9600

19200 baud. (Dans ce dernier cas, uniquement pour transmettre)

La vitesse de transmission en baud est réglée par la commande **BAUD**, et est valable pour les deux canaux. Les vitesses ne peuvent être réglées de façon indépendante.

La **parité** qui doit être utilisée par le QL doit être aussi établie pour convenir aux périphériques. La parité est normalement utilisée pour détecter les erreurs de transmission et peut être impaire, paire, un point, un espace ou sans parité (tous les 8 bits de l'octet sont utilisés pour les données).

Les bits d'arrêt indiquent la fin de transmission d'un octet ou d'un caractère. Le QL reçoit toujours les données avec un bit d'arrêt, un et demi ou deux bits d'arrêt. Il transmet les données avec au moins deux bits d'arrêt. Notez que si le QL est réglé à la vitesse de 9600 baud, il ne peut recevoir avec un seul bit d'arrêt.

Il peut être nécessaire de régler le "handshake" (protocole d'établissement de liaison matériel, appelé aussi "Ready busy") entre le QL et un élément qui lui est connecté. Cela permet au QL et à son périphérique de "monitorer" et de contrôler leur vitesse de communication. Cela peut être nécessaire si l'un d'entre eux ne peut accepter la vitesse de transmission des données. Le QL utilise deux lignes de "handshaking" (transmission) :

CTS Clear To Send (prêt à envoyer)
DTR Data Terminal Ready

Si le DTE ne peut s'adapter à la vitesse de transmission des données, il peut alors mettre en négatif la ligne DTR qui commande au DCE d'interrompre la transmission des données. Ensuite, quand le DTE a reçu les données, il dit au DCE par l'intermédiaire de la ligne DTR de recommencer la transmission des données. De la même manière, le DCE peut arrêter la transmission par le DTE en mettant en négatif la ligne CTS. Si des signaux de contrôle supplémentaires sont nécessaires, ils peuvent être transmis par l'additionneur de 12 V adaptable sur chaque port de série.

Bien que la communication soit souvent possible sans aucun "handshaking" le QL ne recevra pas correctement sans l'utilisation du CTS sur la SER 1 et du DTR sur la SER2.

Les communications sur le QL sont "full duplex", c'est à dire qu'il peut à la fois transmettre et recevoir.

La parité et le handshaking sont choisis lorsque le canal série est ouvert.

Commande	Fonction
BAUD	établit la vitesse de transmission
OPEN	ouvre le canal série
CLOSE	ferme le canal série

* Bit : position binaire

COMPARAISON DES CHAINES DE CARACTERES (REGLES)

STRING COMPARISON

Ordre (pour QL anglais) :

espace

chiffres ou nombres dans l'ordre numérique.

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

Le résultat de la comparaison de deux chaînes peut être :

- **égal** : Tous les caractères ou nombres sont les mêmes ou équivalents
- **plus petit** : Le 1^{er} caractère de la chaîne qui est différent est avant dans l'ordre des caractères.
- **plus grand** : Le 1^{er} caractère de la chaîne qui est différent est après dans l'ordre des caractères

Type de comparaison :

Type 0	Cas dépendant	Comparaison caractère par caractère
Type 1	Cas indépendant	Comparaison caractère par caractère
Type 2	Cas indépendant	Les nombres sont sortis dans l'ordre numérique
Type 3	Cas dépendant	Les nombres sont sortis dans l'ordre numérique

Usage :

- Le type 0 est utilisé normalement par le SuperBASIC.
- Le type 1 est utilisé pour les fichiers et les variables.
- Le type 2 : comparaisons logiques du SuperBASIC : <, <=, =, >=, >, INSTR et <>
- Le type 3 : Equivalence du SuperBASIC ==

CONVERSIONS

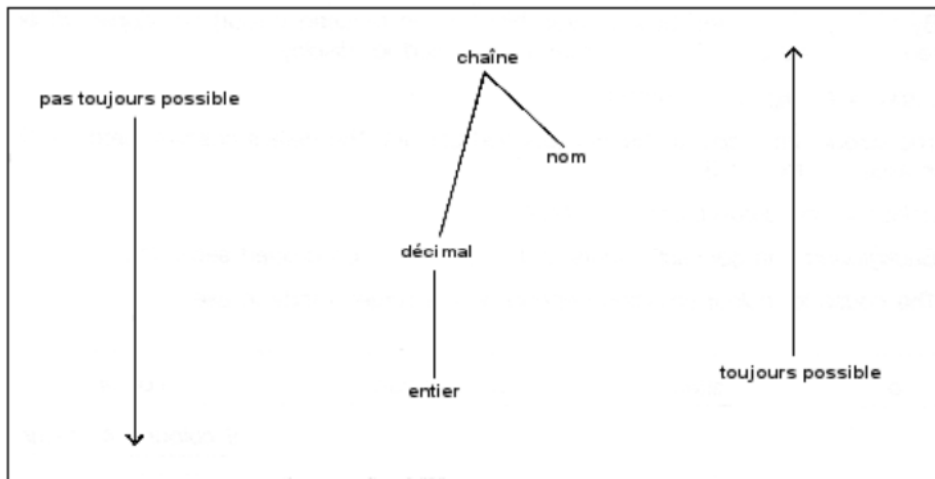
COERCION

Si nécessaire, le SuperBASIC transforme une donnée incompatible en une autre donnée qui permet l'opération spécifiée.

Les **opérateurs** utilisés déterminent la conversion nécessaire. Par exemple, si une opération nécessite un paramètre en chaîne de caractères, et que les paramètres sont donnés en chiffres, alors le SuperBASIC transformera le paramètre en chaîne de caractères. Il n'est pas toujours possible de changer les données sous la forme souhaitée. Dans ce cas l'ordinateur signale une erreur.

Le paramètre d'une **fonction** ou d'une **procédure** peut être aussi converti dans le mode correct. Par exemple, la commande SuperBASIC **LOAD** nécessite un paramètre de type **nom** (sans " ") mais peut accepter un paramètre de type chaîne de caractères (avec " ") et sera converti dans le type correct par la procédure elle-même. Les conversions de ce type sont toujours dépendantes de la façon dont la fonction ou la procédure a été "implémentée" (définie).

Il existe un ordre naturel des types de données sur le QL comme indiqué sur le schéma ci-dessous. La chaîne de caractères est le type le plus général car il peut représenter des noms, des valeurs décimales ou entières. La valeur décimale n'est pas aussi générale que la chaîne de caractères, mais elle est plus générale que la valeur entière, car la valeur décimale peut désigner une valeur entière (la plupart du temps de manière exacte). Le Schéma ci-dessous montre sous forme de diagramme que la donnée peut toujours être convertie si on remonte le diagramme, mais que ce n'est pas toujours possible en le descendant.



Exemple :

a = b + e

Aucune conversion n'est nécessaire pour le calcul de l'addition ni pour l'affectation du résultat.

a% = b + c

Aucune conversion n'est nécessaire pour le calcul de l'addition mais le résultat doit être converti en entier avant l'affectation.

a\$ = b\$ + c\$

b\$ et **c\$** sont convertis en valeur décimale si cela est possible avant l'addition. Si ce n'est pas possible parce que **b\$** ou **c\$** comporte des caractères non numériques, la commande est refusée. En cas d'acceptation, le résultat est converti en chaîne de caractères avant l'affectation.

LOAD "mdv1_data"

La chaîne "**mdv1_data**" est convertie en type nom par la procédure **LOAD** avant l'exécution.

N.B.

Des instructions peuvent être écrites en SuperBASIC qui provoqueraient des erreurs dans la plupart des autres langages. Il est, en général, possible de mélanger des types de données différentes de façon très souple :

- i. PRINT "1" + 2 + "3"
- ii. LET a\$ = 1 + 2 + a\$ + "4"

COORDONNEES PAR POINT

(Système de)

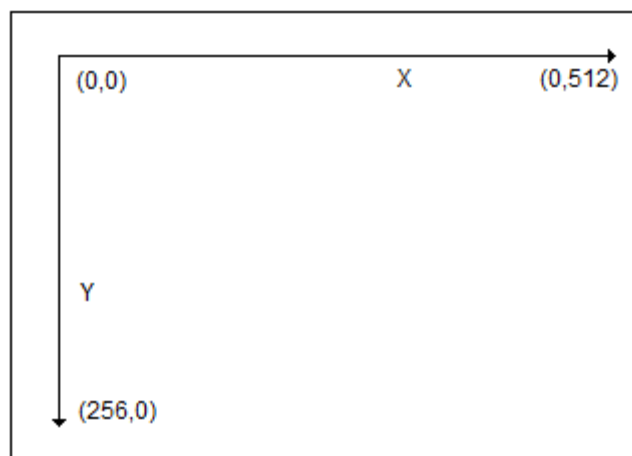
PIXEL

COORDINATES SYSTEM

Le système de coordonnées par points est utilisé pour définir les positions et les tailles des fenêtres, surfaces et position du curseur sur l'écran QL.

L'origine des coordonnées est située en haut et à gauche de l'écran, et les coordonnées sont toujours calculées comme si l'écran était en mode Haute Résolution (Mode 512). Le système utilise le point le plus proche disponible pour fixer le mode, rendant ainsi le système de coordonnées indépendant du mode de l'écran.

Certaines commandes sont toujours relatives à l'origine de la fenêtre par défaut, comme par exemple **WINDOW**, tandis que d'autres sont relatives à l'origine de la fenêtre en cours, comme par exemple **BLOCK**.



Le système de coordonnées par points

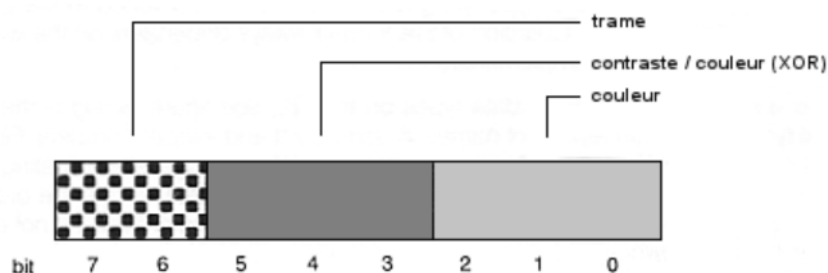
COULEUR

COLOUR

Les couleurs disponibles sur le QL peuvent être aussi bien être une couleur unie, une couleur tramée (en pointillés), un mélange de deux couleurs selon un modèle défini, ainsi la détermination des couleurs sur le QL peut prendre en compte ces trois éléments : une couleur, une couleur composée et une couleur tramée.

Argument unique

L'argument unique précise les trois composantes de la couleur choisie. La couleur principale est définie par les trois positions de droite de l'octet de couleur. Les 3 suivantes correspondent au ou exclusif (XOR) de la couleur principale et de la couleur de contraste. Les deux positions de gauche indiquent le type de la trame.



En spécifiant seulement les trois positions de droite (la couleur souhaitée) aucune trame ne sera utilisée et une couleur unie sera affichée.

Argument double

couleur = fond, contraste

La couleur est une trame de deux couleurs spécifiées. La trame 3 est utilisée par défaut.

Argument triple

Couleur = fond, contraste, trame

Fond, contraste et trame sont définis chacun séparément.

COULEURS

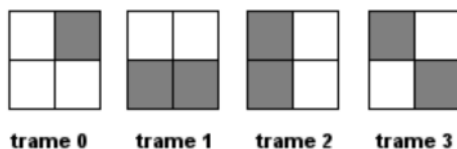
Code	Position binaire	composition	couleur	
			8 couleurs	4 couleurs
0	0 0 0		noir	noir
1	0 0 1		bleu	noir
2	0 1 0	rouge	rouge	rouge
3	0 1 1	rouge +	violet ⁽¹⁾	rouge
4	1 0 0	vert	vert	vert
5	1 0 1	vert +	bleu turquoise ⁽²⁾	vert
6	1 1 0	vert + rouge	jaune	blanc
7	1 1 1	vert + rouge +	bleu blanc	blanc

Composition des couleurs et Codes

⁽¹⁾ couleur appelée magenta ⁽²⁾ couleur appelée cyan

Trame

Une trame mélangée à un fond et une couleur de contraste donne un modèle de trame fine. Les trames peuvent être utilisées sur le QL de la même manière qu'une couleur unie, bien que les trames ne puissent être reproduites de façon correcte sur un téléviseur ordinaire. Il y a quatre modèles de trame :



La trame 3 est le modèle choisi par défaut.

Exemple :

- i. PAPER 255 : CLS
- ii. PAPER 2,4 :CLS
- iii. PAPER 0,2,0 : CLS

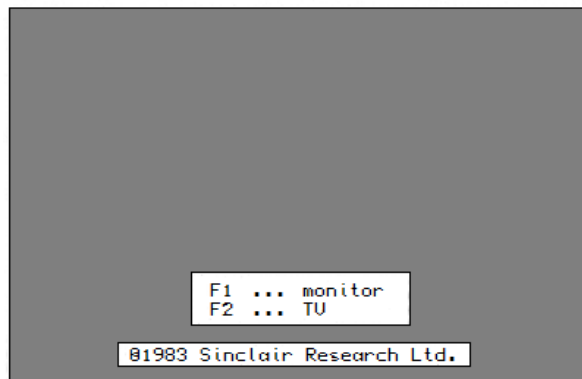
Attention

Il n'est pas possible de reproduire des trames de façon correcte sur des téléviseurs à prise UHF. Il ne faut donc utiliser ces trames que sur QL reliés à un moniteur.

DEMARRAGE

START UP

Immédiatement après la mise sous tension (ou reset : réinitialisation) le QL exécute un test de la RAM, ce qui provoque un dessin-brouillon sur l'écran. Si le test de la RAM est fait, alors l'écran s'éclaire et le copyright d'écran apparaît.



Ecran de démarrage

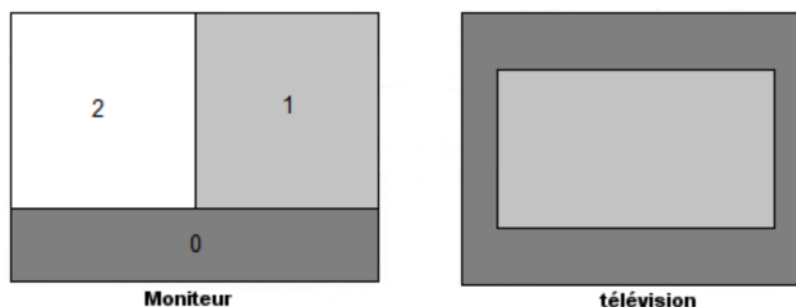
Le QL démarre après la mise sous tension dans un état initial. Il affiche le message copyright et détermine si le QL va être utilisé avec un écran de téléviseur ou un moniteur. Le QL prendra un mode initial d'écran différent et des dimensions de fenêtres différentes selon la réponse.

Appuyez sur F1 si vous utilisez un moniteur et F2 si vous utilisez l'écran d'un téléviseur.

Le QL est capable de démarrer lui-même sur un programme contenu soit dans la cartouche ROM soit dans le Microdrive 1 (microlecteur 1). Si la cartouche ROM contient un programme d'auto-démarrage, alors le démarrage continue sous le contrôle du programme de la ROM. Si rien de valable n'est trouvé, alors le QL teste si le Microdrive 1 a une cartouche. Si une cartouche est trouvée et qu'elle contient un fichier appelé BOOT il est chargé et exécuté.

Écran par défaut

Le QL a trois canaux par défaut qui sont reliés à 3 fenêtres par défaut.



Le canal 0 est utilisé pour la liste des commandes et des messages d'erreurs. Le canal 1 pour les programmes et les sorties graphiques. Le canal 2 pour les listes de programmes. Un canal par défaut peut être modifié en spécifiant un canal optionnel par une commande appropriée.

Attention

Il ne faut pas allumer le QL quand une cartouche est en position. Si un démarrage à partir du Microdrive est désiré, alors la cartouche doit être introduite entre l'allumage et la pression sur F1 ou

DONNEES VARIABLES

DATA TYPES VARIABLES

Entiers

Les entiers sont des nombres entiers allant de - 32767 à + 32768. Les variables sont considérées comme entiers si l'identifiant variable est terminé par le suffixe %. Il n'y a pas de constantes entières en SuperBASIC, toutes les constantes sont stockées comme décimales (floating point numbers).

Syntaxe : *identifiant%*

Exemples :

- i. compteur%
- ii. nombre_d_enfants%
- iii. ceci_est_un_entier_variable%

Nombres décimaux

Les nombres à virgule flottante dans l'étendue de +- (10^{-615} à 10^{+615}) avec 8 chiffres. Le nombre à décimale est le type par défaut des données en SuperBASIC. Toutes les constantes sont gérées en décimales et peuvent être entrées en utilisant une notation exponentielle.

Syntaxe : *identifiant*

Exemples :

- i. compte_courant
- ii. 76.23456
- iii. 234E56

Chaîne de caractères

Une chaîne de caractères est une suite autorisée de caractères SuperBASIC allant jusqu'à 32768 caractères. Les variables sont considérées de type chaîne de caractères (string) si elles sont terminées par \$. La chaîne de caractères est représentée en entourant les caractères SuperBASIC soit de '...' soit de "...".

Syntaxe : *identifiant\$*

Exemples :

- i. Variable_de_chaine\$
- ii. "Ceci est donné en chaîne de caractères"

Nom

Le type nom a la même forme que l'identifiant standard SuperBASIC et est utilisé par le système "Name" en fichier microdrive, dans les procédures.

Syntaxe : *identifiant*

Exemples :

- i. MDV1_fichier
- ii. SER1e

ECRAN

SCREEN

Mode 512

L'écran possède 512 points (pixels) en longueur et 256 points en hauteur. Seules les 4 couleurs suivantes peuvent être visualisées :

- noir
- rouge
- vert
- blanc

Mode 256

L'écran possède 256 points (pixels) en largeur et 256 points en hauteur. Les huit couleurs suivantes peuvent être visualisées :

- noir
- bleu
- rouge
- magenta
- vert
- cyan
- jaune
- blanc

Le mode basse résolution permet aussi le clignotement de l'écran.

Remarque : la réduction du nombre de couleurs augmente la résolution de l'écran.

Attention :

Un écran de téléviseur familial n'est pas toujours capable d'afficher l'écran complet QL, des parties de l'écran, en haut à gauche ne sont pas reproduites. La fenêtre initiale par défaut tient compte de cela et peut réduire la taille effective de l'image. Elle peut rétablir la taille réelle grâce à la commande

WINDOW.

EXPRESSIONS

Les expressions en SuperBASIC peuvent être *des chaînes de caractères, numériques, logiques ou un mélange*. Les données non conformes sont automatiquement converties en données conformes par le système chaque fois que c'est possible.

Définitions

expression | *expression* *opérateur* *expression*
| (*expression*)
| *atom*

atom = | *variable*
| *constante*
| *fonction*
| *tableau*
| *tranche* (1)

variable = | *identifiant*
| *identifiant*%
| *identifiant*\$

fonction = | *identifiant*
| *identifiant*%
| *identifiant*\$

constante = | **[chiffre]**
| **[chiffre]*, *[chiffre]**
| **[chiffre]*[.] *[chiffre]*E*[chiffre]**

La valeur finale retournée par l'évaluation de l'expression peut être un entier donnant une **expression_entière**, une chaîne de caractères donnant une **expression_de_chaîne_de_caractères** ou un chiffre décimal donnant une **expression_décimale**. Souvent les expressions entières et les expressions décimales sont équivalentes, on emploie alors le terme **expression_numérique**.

Des opérateurs logiques peuvent être inclus dans une expression si l'opération spécifiée est vraie. Dans ce cas un « un » est donné comme valeur de l'opération. Si l'opération est fausse, alors un « zéro » est donné. Bien que des opérateurs logiques puissent être utilisés dans n'importe quelle expression, on les utilise normalement si l'expression est conditionnelle (à base de **IF**)

Exemples :

- i. données + 23.3 + 5
- ii. "ahcdefghijklm, nopqrstuvwxyz" (2 à 4)
- iii. 32.1 + (couleur =1)
- iv. compteur = limite

(1) Partie d'un ensemble

ex : NOMS (5 to 10) du 5^e au 10^e caractère inclus.

EXTENSION PERIPHERIQUE

PERIPHERAL EXPANSION

Le connecteur d'extension permet de relier des périphériques spéciaux au QL. Les connexions disponibles sur le connecteur sont :

GND	a	1	b	GND
D3	a	2	b	D2
D4	a	3	b	D1
D5	a	4	b	D0
D6	a	5	b	ASL
D7	a	6	b	DSL
A19	a	7	b	RDWL
A18	a	8	b	DTACKL
A17	a	9	b	BGL
A16	a	10	b	BRL
CLKCPU	a	11	b	A15
RED	a	12	b	RESETCPUL
A14	a	13	b	CSYNCL
A13	a	14	b	E
A12	a	15	b	VSYNCH
A11	a	16	b	VPAL
A10	a	17	b	GREEN
A9	a	18	b	BLUE
A8	a	19	b	FC2
A7	a	20	b	FC1
A6	a	21	b	FC0
A5	a	22	b	A0
A4	a	23	b	ROMOEH
A3	a	24	b	A1
DBGL	a	25	b	A2
SP2	a	26	b	SP3
DSMCL	a	27	b	IPL0L
SP1	a	28	b	BERRL
SP0	a	29	b	IPL1L
VP12	a	30	b	EXTINTL
VM12	a	31	b	VIN
VIN	a	32	b	VIN

Un « L » à la fin du code Signal indique qu'il est actif en position basse.

Signal	Fonction
A0-A19	Adresses du 68008
RDWL	Lecture/Écriture
ASL	Adresse
DSL	Données
BGL	Accès Bus
DSMCL	
CLKCPU	Horloge des périphériques 6800
E	6800 peripherals clock
RED	Rouge
BLUE	Bleu
GREEN	Vert
CSYNCL	Synchro-composite
VSYNCH	Synchro verticale
ROMOEH	Sortie ROM en action
FC0	Status du processeur
FC1	Status du processeur
FC2	Status du processeur
RESETCPUL	Initialisation du CPU

Signaux de sortie QL

Signal	Fonction
DTACKL	Arbitrage sur le Bus
BRL	Appel du Bus
VPAL	Validation d'adresse périphérique
IPL0L	Priorité d'interruption Niveau 0
IPL1L	Priorité d'interruption Niveau 1
BERRL	Erreur Bus
EXTINTL	Interruption extérieure
DBGL	Sortie des données sur le Bus

Signaux d'entrée

Signal	Fonction
D0..D7	Lignes de données

Signaux bidirectionnels

Signal	Fonction
SP0..SP3	Affectation du périphérique 0 à 3
VIN	9 V continu - 500 mA maximum
VM12	-12V
VP12	+12V
GND	Terre

Signaux divers

Cette description n'est pas suffisante pour installer un périphérique d'extension, mais peut servir à acquérir des connaissances de base du principe des extensions. Les détails complets des exigences et des techniques utilisées peuvent être obtenues auprès de « Sinclair Research ».

Un ou plusieurs périphériques peuvent être ajoutés au QL, jusqu'à concurrence de 16 appareils. Un périphérique unique peut être branché directement à la broche d'extension, alors que plusieurs périphériques doivent être branchés sur le module d'extension du QL, qui lui-même est branché à la broche d'extension. Tous les détails pour ce type de branchement sont décrits dans la brochure « *QL Expansion Module Documentation* ».

FENETRES

WINDOWS

Les fenêtres sont des zones de l'écran, et on peut dire que chaque fenêtre ouverte est également un écran. Par exemple, une fenêtre permet de faire défiler des éléments et on peut effacer une fenêtre particulière par la commande **CLS**.

Des fenêtres peuvent être spécifiées et reliées au canal à son ouverture. La fenêtre courante peut être changée avec la commande **WINDOW** et le bord de la fenêtre peut être ajouté par la commande **BORDER**. Pour envoyer quelque chose sur une fenêtre, on écrit sur le canal correspondant. Des données peuvent provenir d'une fenêtre particulière en faisant l'entrée (**INPUT**) sur le canal correspondant. Si plus d'un canal est prêt pour l'entrée, alors l'entrée peut être aiguillée sur un autre canal en appuyant sur :

CTRL C

Le curseur clignotera dans la fenêtre sélectionnée.

Des fenêtres peuvent être utilisées pour des sorties graphiques et non graphiques en même temps. La sortie non graphique est relative à la position du curseur qui peut être positionné n'importe où et à n'importe quelle ligne ou colonne avec la commande **AT**. La sortie graphique est relative à un curseur graphique qui peut être positionné et manipulé avec les procédures graphiques.

Paramètres

Certaines commandes (CLS, PAN, etc...) acceptent des paramètres optionnels pour traiter une seule partie de la fenêtre. Ces paramètres sont définis ci-dessous :

Paramètre	Description
0	Ensemble de l'écran
1	Au-dessus, sans compter la ligne du curseur
2	Au-dessous, sans compter la ligne du curseur
3	Ensemble de la ligne du curseur
4	Colonnes à droite y compris celle du curseur

Commande	Fonction
WINDOW	Redéfinir une fenêtre
BORDER	Ajouter un entourage à la fenêtre
PAPER	Définir la couleur de fond de la fenêtre
INK	Définir la couleur de traçage de la fenêtre
STRIP	Définir une couleur pour les intersections (2 courbes qui se croisent)
PAN	Déplacement latéral du contenu de la fenêtre
SCROLL	Déplacement vertical du contenu de la fenêtre
AT	Positionne le curseur (ligne et colonne)
CLS	Efface le contenu de la fenêtre courante
CSIZE	Détermine la taille des caractères
FLASH	Règle le clignotement
RECOL	Change les couleurs à l'intérieur de la fenêtre courante

Sommaire des commandes

FICHIERS

FILE TYPES

FILES

Toute I/O (Entrée/Sortie) sur le QL va ou est en provenance d'un fichier logique.

Divers types de fichiers existent :

DATA

Les programmes SuperBASIC, les fichiers textes.

On les sauve en utilisant PRINT, SAVE -

On les charge en utilisant INPUT, INKEY, LOAD, etc

EXEC

Un programme transitoire exécutable.

On les sauve en utilisant SEXEC

On les charge en faisant EXEC, EXEC_W, etc

CODE

Données brutes en mémoire, images d'écran, etc...

On les sauve en faisant SBYTES

On les charge en faisant LBYTES

FONCTIONS ET PROCEDURES

FUNCTIONS AND PROCEDURES

En SuperBASIC les *fonctions et procédures* sont définies par :

```
DEFine FUNction et DEFine PROCEDURE
```

Pour activer (ou appeler) une fonction, on tape son nom en expression SuperBASIC. La fonction doit être incluse dans une expression, car elle réfère à une valeur, et cette valeur doit être utilisée. On appelle une procédure en tapant son nom dès le début d'une instruction SuperBASIC.

Les données peuvent être transmises dans une instruction ou une procédure en rajoutant une liste de **paramètres effectifs** après le nom de la fonction ou de la procédure. Cette liste est comparée à une liste semblable quand la fonction ou la procédure ont été définies. La seconde liste est appelée **paramètres formels** de la fonction ou procédure. Les paramètres formels doivent être des variables. Les paramètres effectifs doivent être un tableau, une tranche de tableau ou une expression SuperBASIC dont la forme la plus simple est une variable simple ou une constante.

Les paramètres effectifs sont des expressions effectives, il faut donc leur associer un type particulier. Les paramètres formels sont simplement utilisés pour indiquer comment les paramètres effectifs doivent être traités, on ne leur associe pas de type. Les éléments de chaque liste de paramètres sont « *paired off* » (appariés) de manière à ce que, lorsque les fonctions et procédures sont appelées, les paramètres formels puissent devenir équivalents aux paramètres effectifs.

On peut utiliser les paramètres de trois façons distinctes

- Si le paramètre effectif est une simple variable, alors si la donnée est affectée au paramètre formel dans la fonction ou la procédure, alors la donnée est aussi affectée au paramètre effectif correspondant.
- Si le paramètre effectif est une expression, alors les données affectées au paramètre formel correspondant n'auront pas d'effet hors de la procédure. Notez qu'une variable peut être changée en expression en la mettant entre parenthèses.
- Si le paramètre effectif est une variable, qui n'a pas été établie auparavant, alors les données affectées au paramètre formel correspondant rendront la variable paramètre effectif.

Il n'est jamais nécessaire de spécifier tous les éléments des paramètres effectifs. Par exemple l'instruction en SuperBASIC **PRINT** est implémentée comme une procédure SuperBASIC et peut accepter un nombre variable de paramètres de types différents.

Des variables peuvent être incluses dans une fonction ou une procédure par l'instruction **LOCAL**. Les variables internes n'ont pas d'effet sur d'autres variables en dehors de la fonction ou de la procédure dans lesquelles elles ont été définies. Cela permet une plus grande liberté dans le choix de noms de variables sans risquer d'altérer les variables externes. Les variables internes sont utilisables pour n'importe quelle fonction ou procédure à moins qu'elles ne soient définies comme internes.

En SuperBASIC, les fonctions et procédures sont récursives. Une fonction ou une procédure peut s'appeler elle-même directement ou indirectement.

Commande	Fonction
DEFine FuNction	Définit une fonction
DEFine PROCedure	Définit une procédure
RETurn	Quitte une fonction ou une procédure
LOCAl	Définit les variables locales dans une fonction ou une procédure.

FONCTIONS MATHÉMATIQUES

MATHS FUNCTIONS

Le SuperBASIC possède les fonctions standards en trigonométrie et en mathématiques.

Fonction	Nom
COS	cosinus
SIN	sinus
TAN	tangente
ATAN	arctangente
ACOT	arcotangente
ACOS	arc cosinus
ASIN	arc sinus
COT	cotangente
EXP	exponentiel
LN	logarithme naturel (népérien)
LOG10	logarithme commun (base 10)
INT	entier
ABS	valeur absolue
RAD	convertir en radians
DEG	convertir en degrés
PI	renvoie la valeur de PI
RND	génère un nombre aléatoire
RANDOMISE	réinitialise le générateur de nombres aléatoires

GESTION DES ERREURS

ERROR HANDLING

Les erreurs sont ainsi signalées en SuperBASIC :

At line line_number error_text

Le numéro de la ligne est celui où se trouve l'erreur, qui est ainsi signalée.

- 1. Not complete** **Opération non terminée**
Une opération a été interrompue prématurément
- 2. Invalid job** **Tâche invalide**
Il faut contrôler la multiprogrammation ou l'I/O (entrée/sortie)
- 3. Out of memory** **Fin de la capacité mémoire**
Le QDOS et/ou le SuperBASIC n'ont pas de mémoire suffisante
- 4. Out of range** **Sortie des limites**
On a tenté d'écrire hors des limites d'une fenêtre, ou les coordonnées d'un tableau sont incorrectes
- 5. Buffer full** **Tampon plein**
Une opération I/O a rempli le tampon avant qu'un disque terminal ait été retrouvé
- 6. Channel not open** **Canal non ouvert**
On a tenté d'écrire, de lire ou de fermer un canal non ouvert
- 7. Not found** **Non trouvé**
On ne trouve pas le fichier, le périphérique ou le medium.
Le SuperBASIC ne peut trouver un identifiant. Ceci peut provenir de structures mal emboîtées.
- 8. Already exists** **Existe déjà**
Le système de fichier a trouvé un fichier qui existe sous le nom avec lequel on veut ouvrir un nouveau fichier
- 9. In use** **En usage**
Le système de fichier a trouvé un fichier ou un périphérique déjà utilisé
- 10. End of file** **Fin de fichier**
Fin de fichier détectée pendant une entrée
- 11. Drive full** **Cartouche remplie**
Une cartouche de périphérique est remplie
- 12. Bad name** **Nom incorrect**
Le système de fichier a reconnu le nom, mais il y a une erreur de syntaxe ou de paramètre.
En SuperBASIC cela signifie que le nom a été utilisé hors du contexte. Par exemple, une variable a été utilisée comme procédure
- 13. Xmit error** **Erreur de transmission**
Erreur de parité de RS-232-C
- 14. Format failed** **Défaillance dans le formatage**
L'opération de formatage ne fonctionne pas. Il y a possibilité d'erreur de médium
- 15. Bad parameter** **Mauvais paramètre**

Erreur de paramètre dans une procédure ou l'appel d'une fonction. On a essayé de lire des données d'un périphérique prévu pour écrire seulement.

16. Bad medium Erreur de support

Possibilité d'erreur dans le support

17. Error in expression Erreur de l'expression

Une erreur a été détectée dans l'évaluation d'une expression

18. Overflow Dépassement de capacité

Dépassement arithmétique, division par zéro, racine carrée d'un nombre négatif, etc...

19. Non implemented Non « implémenté »

Commande non installée dans le système

20. Read only Lecture uniquement

On a tenté d'écrire un fichier uniquement prévu pour la lecture

21. Bad line Ligne incorrecte

Une erreur de syntaxe en SuperBASIC est survenue.

Cette erreur peut aussi survenir en cas d'erreur d'appel d'une fonction de procédure. Cela empêche d'écrire des lignes qui modifieraient le programme. Tapez **NEW** ou **CLEAR** pour débloquer la situation.

Correction d'Erreurs

Quand une erreur survient, le programme peut repartir sur la ligne suivante en tapant :

CONTINUE

Si l'erreur peut être corrigée sans changer le programme, ce programme peut être poursuivi sur la ligne où se trouvait l'erreur, en tapant :

RETRY

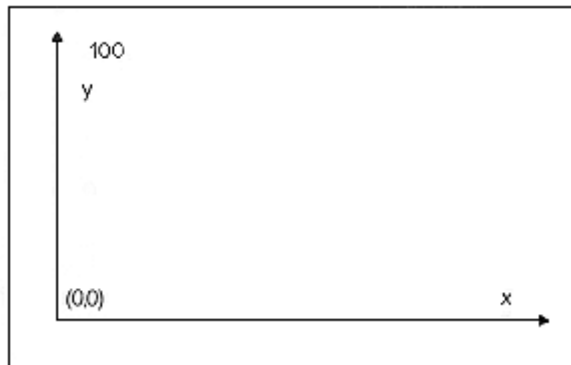
GRAPHIQUES

GRAPHICS

Il faut noter que l'écran du QL n'a pas de pixels carrés, et qu'en changeant de mode on changera la forme des pixels. Ainsi si les procédures de graphiques étaient simplement établies en pixels, ils représenteraient des figures différentes dans les deux modes. Par exemple, dans un mode on aurait un cercle, et cette même figure, dans l'autre mode, serait une ellipse.

Les procédures de graphiques permettent, dans n'importe quel mode d'écran, de produire des figures compatibles. Il n'est pas possible d'utiliser seulement les comptes en pixels pour indiquer les formes des figures. Par contre, les procédures de graphiques utilisent une échelle arbitraire et un système coordonné pour spécifier les formes et les positions des figures.

Les procédures de graphiques utilisent le **graphics coordinate system** (système coordonné de graphiques), c'est à dire un dessin correspondant au début des graphiques qui le coin supérieur gauche de la fenêtre spécifiée ou par défaut. Pour définir la position des fenêtres et des blocks, ce n'est pas le même que l'origine pixel qui est utilisé. Le point d'origine des graphiques permet d'utiliser un système coordonné cartésien. Voyez la figure. Un curseur de graphiques est mis à jour après chaque opération graphique, ainsi les opérations suivantes peuvent être relatives au curseur ou relatives.



Le système coordonné de graphiques

L'échelle est telle que la pleine distance dans la direction verticale dans la fenêtre spécifiée ou par défaut mesure 100 par défaut. L'échelle en x est égale à celle en y (abscisse = ordonnée), cependant la longueur de la ligne qui peut être tracée en x dépend de la forme de la fenêtre. En augmentant l'échelle on augmente la taille de la figure dans les limites de la taille de la fenêtre. Si la sortie du graphique se fait sur une taille différente de fenêtre, alors la dimension s'adapte à la taille de la nouvelle fenêtre. Si une figure déborde la fenêtre de sortie, alors la figure est correctement coupée.

Il est utile de considérer la fenêtre comme un espace permettant le dessin des figures. La commande **SCALE** (échelle) permet au graphique d'origine de se placer de telle manière que la fenêtre entoure l'espace du graphique.

Les sorties de procédures graphiques se font dans la fenêtre spécifiée ou dans le canal par défaut. La sortie est dessinée dans la couleur de l'encre spécifiée pour ce canal.

Commande	Fonction
CIRCLE	dessine une ellipse ou un cercle
LINE	dessine une ligne
ARC	dessine un arc de cercle
POINT	trace un point
CIRCLE_R	dessine une ellipse ou un cercle

LINE_R	dessine une ligne	}
ARC_R	dessine un arc de cercle	}
POINT_R	trace un point	}
		} relative
SCALE	établit l'échelle et l'origine du mouvement	
FILL	remplit une forme	
CURSOR	position du texte	

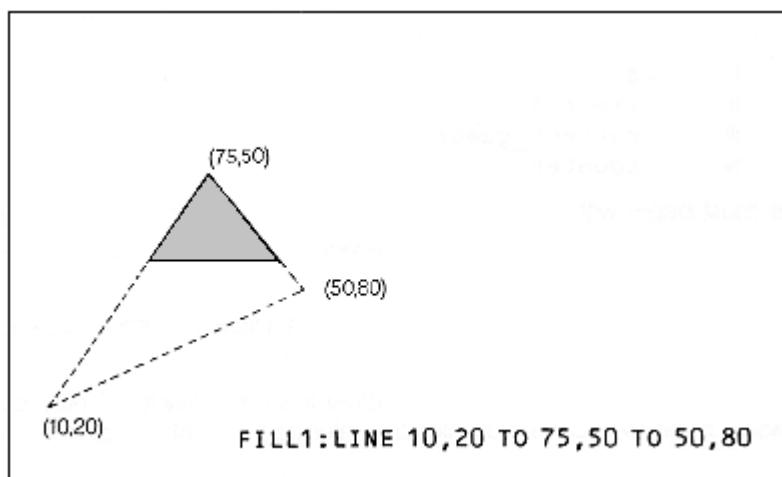
Remplissage de graphiques

Graphics fill

Les figures tracées grâce aux procédures graphiques ou de dessins peuvent être, si on le désire, remplies par une trame ou une couleur. En sélectionnant **FILL** la figure est remplie au fur et à mesure qu'elle est dessinée.

Le fichier algorithmique stocke une liste de points à imprimer au lieu de les marquer effectivement. Quand la figure est fermée, il y a deux points de la même figure sur la même ligne horizontale. Ces deux points sont reliés par une ligne de la couleur courante et le procédé se répète. Il faut toujours resélectionner « **FILL** » avant de dessiner une nouvelle figure pour être sûr que le tampon utilisé pour stocker la liste des points est vidé.

Le diagramme suivant illustre la fonction FILL :



Il est possible de spécifier une séquence utilisateur pour procéder au remplissage et pour remplir avec un procédé arbitraire.

Attention :

Il y a une restriction d'implémentation sur **FILL**.

FILL ne doit pas être utilisé sur des formes rentrantes (à formes concaves) Les formes rentrantes doivent être divisées en plus petites formes qui ne sont pas rentrantes, et chaque élément de la forme est alors rempli de façon indépendante.

GRAPHIQUE DE LA TORTUE

TURTLE GRAPHICS

Le SuperBASIC a un jeu de commandes pour le graphique de la Tortue.

Commande	Fonction
PENUP	arrêt du tracé
PENDOWN	démarrage du tracé
MOVE	mouvement de la tortue
TURN	changement de direction
TURNT0	changement dans une direction précise

Le jeu de commandes est réduit au minimum et normalement doit être utilisé avec d'autres procédures pour en étendre la gamme.

Par exemple :

```
10 DEFine PROCedure avance (distance)
20   MOVE distance
30 END DEFine
40 DEFine PROCedure recule      (distance)
50   MOVE - distance
60 END DEFine
70 DEFine PROCedure gauche (angle)
80   TURN angle
90 END DEFine
100 DEFine PROCedure droite (angle)
110  TURN - angle 1
120 END DEFine
```

Ceci définit quelques-unes des plus remarquables commandes de la tortue.

Au départ le stylo de la tortue est levé, et la tortue est dirigée à 0° (vers le côté droit de la fenêtre). La commande **FILL** peut aussi fonctionner pour les figures tracées avec la Tortue-Graphique. Aussi les graphiques ordinaires et les graphiques de « tortue » peuvent être mélangés bien que la direction de la Tortue ne soit pas toujours définie.

HORLOGE

CLOCK

Le QL contient une horloge intégrée, qui fonctionne lorsque l'ordinateur est en marche.

Le format utilisé pour indiquer la date et l'heure est le format standard ISO.

1985 JAN 01 12:09:10

N.B.

On peut avoir à part l'année, le mois, le jour et l'heure en découpant la chaîne de caractères par la commande **DATE\$** et on peut éventuellement les traduire en français.

Commande	Fonction
SDATE	entrer l'heure
ADATE	régler l'heure
DATE	renvoie la date sous la forme d'un nombre
DATE\$	renvoie la date sous la forme d'une chaîne de caractères
DAY\$	renvoie le jour de la semaine

IDENTIFIANT

IDENTIFIER

Définition

Un identifiant SuperBASIC est une séquence de lettres, de numéros et des soulignés (_).

lettre = | a..z
 | A..Z

numéro = | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |

identifiant = *lettre* * [| *lettre* | *numéro* | _ |]*

Exemples :

i. a
ii. 2 limit_1
iii. après_midi
iv. compteur

Un identifiant doit commencer par une lettre suivie par une séquence de lettres, de numéros et de soulignés, et peut être long de 255 caractères. Les majuscules et les minuscules sont équivalentes. Les identifiants sont utilisés en SuperBASIC pour identifier des variables, des procédures, des fonctions, des boucles de répétition, etc...

Attention :

La seule signification d'un identifiant est sa capacité d'identifier des constructions SuperBASIC. Il ne faut pas tirer de conclusion de la signification courante du nom de l'identifiant

INTERRUPTION PROVOQUEE

BREAK

Quand l'ordinateur ne peut pas répondre, ou que vous souhaitez arrêter un programme, maintenir appuyé :

et pressez sur la

Pour arrêter momentanément le Basic, pressez :

et

Pressez n'importe quelle touche pour le faire repartir.

JEUX DE CARACTERES

TOUCHES DU CLAVIER

Decimal	Hex	Touches	Affichage/Fonction
0	00	CTRL ESC	
1	01	CTRL A	
2	02	CTRL B	
3	03	CTRL C	Changement de canal d'entrée
4	04	CTRL D	
5	05	CTRL E	
6	06	CTRL F	
7	07	CTRL G	
8	08	CTRL H	
9	09	TAB	Champ suivant
10	0A	ENTER	New line / Validation de la commande
11	0B	CTRL K	
12	0C	CTRL L	
13	0D	CTRL M	Entrée
14	0E	CTRL N	
15	0F	CTRL O	
16	10	CTRL P	
17	11	CTRL Q	
18	12	CTRL R	
19	13	CTRL S	
20	14	CTRL T	
21	15	CTRL U	
22	16	CTRL V	
23	17	CTRL W	
24	18	CTRL X	
25	19	CTRL Y	
26	1A	CTRL Z	
27	1B	ESC	Abandon de la commande en cours
28	1C	CTRL SHIFT u accent grave	
29	1D	CTRL SHIFT e accent grave	
30	1E	CTRL SHIFT ^	
31	1F	CTRL SHIFT ESC	
32	20	SPACE	Espace
33	21	SHIFT 1	!
34	22	SHIFT 2	"
35	23	SHIFT 3	#
36	24	SHIFT 4	\$
37	25	SHIFT 5	%
38	26	SHIFT 7	&
39	27	SHIFT 6	'
40	28	SHIFT 9	(
41	29	SHIFT 0)
42	2A	SHIFT 8	*
43	2B	SHIFT =	+
44	2C	,	,
45	2D	-	-
46	2E	.	.
47	2F	SHIFT a accent grave	/
48	30	0	0
49	31	1	1

50	32	2	2
51	33	3	3
52	34	4	4
53	35	5	5
54	36	6	6
55	37	7	7
56	38	8	8
57	39	9	9
58	3A	SHIFT ;	:
59	3B	;	;
60	3C	SHIFT ,	<
61	3D	=	=
62	3E	SHIFT .	>
63	3F	SHIFT c cédille	?
64	40	CTRL 6	@
65	41	SHIFT A	A
66	42	SHIFT B	B
67	43	SHIFT C	C
68	44	SHIFT D	D
69	45	SHIFT E	E
70	46	SHIFT F	F
71	47	SHIFT G	G
72	48	SHIFT H	H
73	49	SHIFT I	I
74	4A	SHIFT J	J
75	4B	SHIFT K	K
76	4C	SHIFT L	L
77	4D	SHIFT M	M
78	4E	SHIFT N	N
79	4F	SHIFT O	O
80	50	SHIFT P	P
81	51	SHIFT Q	Q
82	52	SHIFT R	R
83	53	SHIFT S	S
84	54	SHIFT T	T
85	55	SHIFT U	U
86	56	SHIFT V	V
87	57	SHIFT W	W
88	58	SHIFT X	X
89	59	SHIFT Y	Y
90	5A	SHIFT Z	Z
91	5B	CTRL 9	[
92	5C	SHIFT e accent aigu	\
93	5D	CTRL 0]
94	5E	CTRL ^	^ (Symbole d'exponentiation du SuperBasic)
95	5F	SHIFT -	-
96	60	SHIFT u accent grave	£
97	61	A	a
98	62	B	b
99	63	C	c
100	64	D	d
101	65	E	e
102	66	F	f
103	67	G	g
104	68	H	h
105	69	I	i
106	6A	J	j
107	6B	K	k

108	6C	L	l
109	6D	M	m
110	6E	N	n
111	6F	O	o
112	70	P	p
113	71	Q	q
114	72	R	r
115	73	S	s
116	74	T	t
117	75	U	u
118	76	V	v
119	77	W	w
120	78	X	x
121	79	Y	y
122	7A	Z	z
123	7B	CTRL -	{
124	7C	CTRL 8	
125	7D	CTRL =	}
126	7E	CTRL u accent grave	~
127	7F	SHIFT ESC	©
128	80	SHIFT ^ puis a	ä
129	81	CTRL SHIFT 1	ã
130	82	CTRL SHIFT a accent grave	â
131	83	e accent aigu	é
132	84	CTRL SHIFT 4	ö
133	85	CTRL SHIFT 5	õ
134	86	CTRL SHIFT 7	ø
135	87	CTRL a accent grave	ü
136	88	C cédille	ç
137	89	CTRL SHIFT 0	ñ
138	8A	CTRL SHIFT 8	æ
139	8B	CTRL SHIFT =	œ
140	8C	CTRL ,	á
141	8D	a accent grave	à
142	8E	CTRL .	â
143	8F	CTRL c cédille	ë
144	90	E acent grave	è
145	91	CTRL SHIFT 3	ê
146	92	CTRL SHIFT 9	ï
147	93	CTRL 3	í
148	94	CTRL 4	ì
149	95	CTRL SHIFT ;	î
150	96	CTRL SHIFT e accent aigu	ó
151	97	CTRL 5	ò
152	98	CTRL SHIFT Z	ô
153	99	CTRL SHIFT V	ú
154	9A	U accent grave	ù
155	9B	CTRL ;	û
156	9C	CTRL SHIFT ,	ß
157	9D	CTRL 2	¢
158	9E	CTRL SHIFT .	¥
159	9F	CTRL SHIFT c cédille	`
160	A0	CTRL SHIFT 2	Ä
161	A1	CTRL SHIFT A	Å
162	A2	CTRL SHIFT B	Â
163	A3	CTRL SHIFT C	É
164	A4	CTRL SHIFT D	Ö

165	A5	CTRL SHIFT E	Ö
166	A6	CTRL SHIFT F	Ø
167	A7	CTRL SHIFT G	Ù
168	A8	CTRL SHIFT H	Ç
169	A9	CTRL SHIFT I	Ñ
170	AA	CTRL SHIFT J	Æ
171	AB	CTRL SHIFT K	Œ
172	AC	CTRL SHIFT L	α
173	AD	CTRL SHIFT M	ø
174	AE	CTRL SHIFT N	θ
175	AF	CTRL SHIFT O	λ
176	B0	CTRL SHIFT P	μ
177	B1	CTRL SHIFT Q	π
178	B2	CTRL SHIFT R	Φ
179	B3	CTRL SHIFT S	ı
180	B4	CTRL SHIFT T	¿
181	B5	CTRL SHIFT U	Ž
182	B6	CTRL SHIFT V	§
183	B7	CTRL SHIFT W	¶
184	B8	CTRL SHIFT X	«
185	B9	CTRL SHIFT Y	»
186	BA	CTRL 7	°
187	BB	CTRL I	÷
188	BC	CTRL e accent aigu	←
189	BD	CTRL e accent grave	→
190	BE	CTRL SHIFT 6	↑
191	BF	CTRL SHIFT _	↓
192	C0	GAUCHE	Déplace le curseur d'un caractère vers la gauche
193	C1	ALT GAUCHE	Place le curseur au début de la ligne
194	C2	CTRL GAUCHE	Efface le caractère situé à gauche du curseur
195	C3	CTRL ALT GAUCHE	Efface le début de la ligne jusqu'au curseur
196	C4	SHIFT GAUCHE	Déplace le curseur d'un mot vers la gauche
197	C5	SHIFT ALT GAUCHE	Déplace vers la gauche
198	C6	SHIFT CTRL GAUCHE	Efface le mot situé à gauche du curseur
199	C7	SHIFT CTRL ALT GAUCHE	
200	C8	DROITE	Déplace le curseur d'un caractère vers la droite
201	C9	ALT DROITE	Place le curseur à la fin de la ligne
202	CA	CTRL DROITE	Efface le caractère situé sous le curseur
203	CB	CTRL ALT DROITE	Efface la fin de la ligne depuis le curseur
204	CC	SHIFT DROITE	Déplace le curseur d'un mot vers la droite
205	CD	SHIFT ALT DROITE	Déplace vers la gauche
206	CE	SHIFT CTRL DROITE	Efface le mot situé sous et à droite du curseur
207	CF	SHIFT CTRL ALT DROITE	
208	D0	HAUT	Curseur à droite
209	D1	ALT HAUT	Déplace vers le haut
210	D2	CTRL HAUT	
211	D3	ALT CTRL HAUT	
212	D4	SHIFT HAUT	Haut de l'écran
213	D5	SHIFT ALT HAUT	
214	D6	SHIFT CTRL HAUT	
215	D7	SHIFT CTRL ALT HAUT	
216	D8	BAS	Curseur vers le bas
217	D9	ALT BAS	Déplace vers le bas
218	DA	CTRL BAS	
219	DB	ALT CTRL BAS	
220	DC	SHIFT BAS	Bas de l'écran
221	DD	SHIFT ALT BAS	
222	DE	SHIFT CTRL BAS	

223	DF	SHIFT CTRL ALT BAS	
224	E0	ALPHA	Bascule majuscules / minuscules
225	E1	ALT ALPHA	
226	E2	CTRL ALPHA	
227	E3	ALT CTRL ALPHA	
228	E4	SHIFT ALPHA	
229	E5	SHIFT ALT ALPHA	
230	E6	SHIFT CTRL ALPHA	
231	E7	SHIFT CTRL ALPHA	
232	E8	F1	
233	E9	CTRL F1	
234	EA	SHIFT F1	
235	EB	CTRL SHIFT F1	
236	EC	F2	
237	ED	CTRL F2	
238	EE	SHIFT F2	
239	EF	CTRL SHIFT F2	
240	F0	F3	
241	F1	CTRL F3	
242	F2	SHIFT F3	
243	F3	CTRL SHIFT F3	
244	F4	F4	
245	F5	CTRL F4	
246	F6	SHIFT F4	
247	F7	CTRL SHIFT F4	
248	F8	F5	
249	F9	CTRL F5	
250	FA	SHIFT F5	
251	FB	CTRL SHIFT F5	
252	FC	SHIFT space	Espace "Spécial"
253	FD	SHIFT TAB	
254	FE	SHIFT ENTER	
255	FF	Voir ci dessous	

La touche ALT pressée conjointement avec n'importe quelle autre touche que ALPHA ou les touches du curseur génère le code FF, suivi d'un octet indiquant ce qui se passerait si ALT n'était pas pressée.

MANETTE DE JEU

JOYSTICK

Les repères « joystick » sur les ports CTL1 et CTL2 permettent de connecter deux manettes de jeu au QL. Il faut utiliser un adaptateur spécial pour insérer les manettes de jeu.

Les manettes sont conçues simuler l'appui de touches lorsqu'elles sont positionnées de telle ou telle manière. Tout programme utilisant des manettes de jeu peut être capable de s'adapter à ces touches. Le clavier QL peut être lu directement en utilisant la commande **KEYROW**.

	CTL1	CTL2
mode	touche	touche
up	curseur haut	F4
down	curseur bas	F2
left	curseur gauche	F1
right	curseur droite	F3
fire	espace	F5

Pour obtenir « Fire » (Feu), appuyer sur le bouton de la manette.

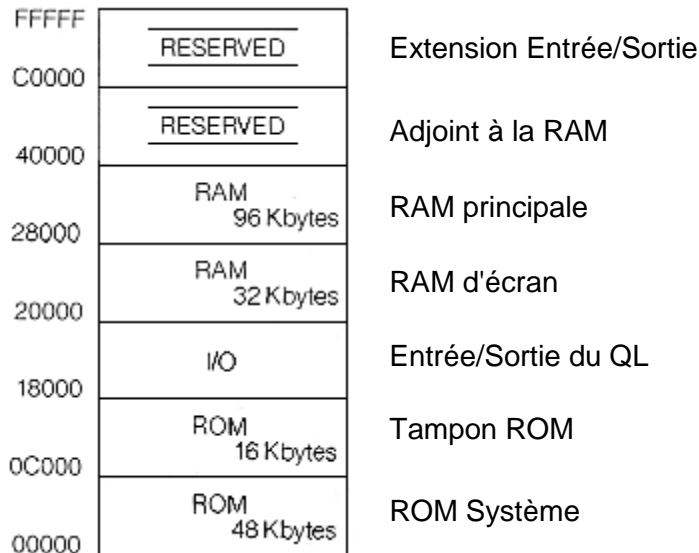
Commentaire :

Les ports de manettes de jeu permettent de brancher d'autres types de périphériques au QL.

MEMOIRE CENTRALE

MEMORY MAP

Le QL contient un microprocesseur Motorola 68008, qui peut adresser 1 Mégabyte (1 million d'octets) de mémoire, c'est à dire de 00000 à FFFFF. L'utilisation des adresses à l'intérieur de cette mémoire est ainsi définie par Sinclair Research.



L'écran RAM est organisé en séries de mots de 16 bits commençant à l'adresse Hex 20000 et progressant de droite à gauche sur chaque ligne d'écran, et du sommet au bas de la figure. Les bits à l'intérieur de chaque mot sont organisés de telle manière qu'un pixel à gauche est toujours plus significatif qu'un pixel à droite. (Le mode pixel sur l'écran ressemble au mode binaire). Cependant l'organisation de l'information de la couleur sur les deux modes d'écran est différente.

high byte AO=0	low byte AO=1	mode
GGGGGGGG	RRRRRRRR	512 mode (high res)
GFGFGFGF	RBRBRBRB	256 mode (low res)

Carte des couleurs

G : Vert - B : Bleu - R : Rouge - F : Flash (intermittence lumineuse)

Le mode Flash est toujours annulé au début de chaque ligne affichée. En mode de haute résolution, le rouge et le vert spécifiés ensemble sont interprétés par l'ordinateur comme blanc.

Attention :

Si on utilise des zones réservées dans la mémoire centrale, cela peut être la cause d'incompatibilité avec de futurs produits Sinclair. Une sortie inadaptée d'adresses au périphérique I/O d'adresses peut provoquer des résultats imprévisibles. Ces zones ne doivent pas servir à écrire et il ne faut pas les utiliser pour d'autres usages que ceux prévus. Prélever des zones en usage comme tampon de microprocesseur peut détruire les données du microprocesseur et il peut en résulter une perte d'informations. Si on prélève des zones utilisées comme table, cela peut détruire le système : il peut en résulter la disparition de données et de programmes.

Toute I/O (Entrée/Sortie) doit être exécutée, soit en utilisant les commandes correspondantes SuperBASIC, soit le dispositif operating system du QDOS.

MICROLECTEURS

MICRODRIVES

Les microlecteurs assurent le principal stockage permanent sur le QL. Chaque cartouche de microlecteur a une capacité d'au moins 100 Kbytes. Il est possible d'avoir un espace libre pour la mémoire par le QDOS en tant que tampons de microlecteurs quand c'est nécessaire pour réaliser un travail.

Chaque cartouche doit être formatée avant l'usage et peut contenir 255 secteurs de 512 bytes par secteur. Le QDOS garde un annuaire des fichiers contenus sur la cartouche. Chaque fichier de microlecteur est identifié en utilisant un nom standard de fichier ou de périphérique.

Une cartouche peut être protégée contre une réutilisation en enlevant la petite languette à droite.

Soins généraux

Chaque cartouche de microlecteur contient une boucle de 200 pouces (un pouce = 2 cm 54) de ruban magnétique qui progresse à 28 pouces par seconde. Le ruban se déroule en 7 secondes 1/2.

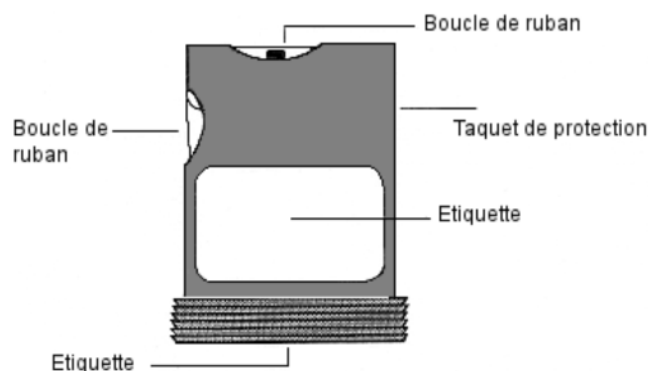
- NE JAMAIS** toucher le ruban avec les doigts
introduire quelque chose dans la cartouche.
- NE JAMAIS** allumer ou éteindre l'ordinateur avec des cartouches en place.
- TOUJOURS** remettre les cartouches dans leur chemise après usage.
- TOUJOURS** placer et enlever les cartouches lentement et avec soin.
- TOUJOURS** s'assurer que la cartouche est bien en place avant de faire fonctionner le microlecteur.
- NE JAMAIS** déplacer le QL avec les cartouches en place, même si elles ne fonctionnent pas.
- NE JAMAIS** toucher la cartouche quand le microlecteur fonctionne.
- EVITEZ** de mettre et d'enlever la cartouche sans faire fonctionner le microlecteur.

Boucles de ruban

Si une boucle de ruban apparaît à l'un ou l'autre endroit indiqué sur le dessin, le remettre doucement dans la cartouche. Utilisez pour cela un instrument non fibreux : le côté d'un crayon ou d'un stylo.

Ne touchez jamais le ruban avec les doigts pour quelque raison que ce soit.

Avant utilisation de cartouches vierges, formatez-les plusieurs fois pour que le ruban soit en bonne place.



Commande	Fonction
FORMAT	prépare à l'usage une nouvelle cartouche
DELETE	efface un fichier sur une cartouche
DIR	liste les fichiers présents sur une cartouche

SAVE	
SBYTES	sauve des données sur une cartouche
SEXEC	
<hr/>	
LOAD	
LBYTES	
EXEC	charge des données sur une cartouche
MERGE	
<hr/>	
OPEN_IN	
OPEN_NEW	
OPEN	ouvre et ferme des fichiers
CLOSE	
<hr/>	
PRINT	
INPUT	E/S fichier SuperBASIC
INKEY\$	
<hr/>	

Attention :

Si vous essayez d'écrire sur une cartouche qui est protégée contre l'écriture, alors le QL essaiera sans cesse d'écrire les données. Enlever la cartouche pour arrêter le QL.

MONITEUR

MONITOR

Un moniteur couleur peut être connecté au QL par la prise RGB à l'arrière de l'ordinateur. La connexion se fait avec une prise DIN 8 broches côté QL et une prise adaptée côté moniteur.

broche	signal	fonction
1	PAL	vidéo composite PAL
2	GND	masse
3	VIDEO	video composite monochrome (3)
4	SCYNC	synchronisation composite (1)
5	VSYNC	synchronisation verticale (2)
6	GREEN	vert (1)
7	RED	rouge (1)
8	BLUE	bleu (1)

- (1) 0-5V TTL compatible (actif valeur haute)
- (2) 0,5V TTL compatible (actif valeur basse)
- (3) 1V pk-pk 75 ohms (non inversée)

Un moniteur monochrome peut être connecté à l'aide d'un câble avec prise DIN 3 broches ou 8 broches côté Q.L. La connexion au moniteur varie suivant les appareils.

Le moniteur doit avoir une entrée vidéo composite (c'est le standard) 75 ohm IV crête à crête. Les prises DIN 3 broches peuvent se trouver dans les boutiques Hi-Fi.

Un moniteur couleur peut être connecté à l'aide d'un câble avec prise DIN 8 broches côté QL. La connexion au moniteur varie suivant les appareils (pas de standard) et doit être souvent adaptée.

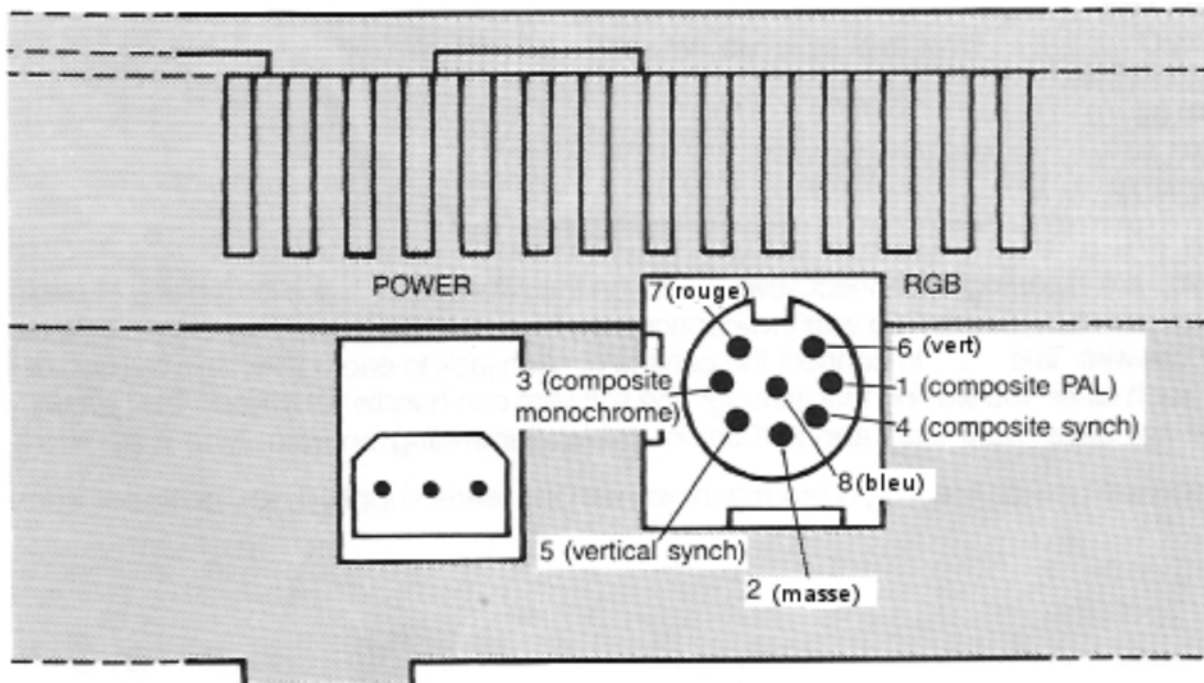


Schéma du connecteur

MOT CLE

KEYWORD

Les mots-clés SuperBASIC sont des identifiants qui sont définis dans le Guide de Référence - Lexique. Les mots-clés ont la même forme que les identifiants en SuperBASIC. La position du mot-clé n'a pas d'importance. Les mots-clés sont indiqués par un mélange de lettres majuscules et minuscules et toujours reproduits en entier. La partie MAJuscule indique le minimum qu'il faut taper pour que le SuperBASIC le reconnaisse.

La liste des mots-clés peut être étendue en ajoutant des procédures au système et en les chargeant dans le système QL. C'est une bonne idée d'identifier ainsi les procédures qui doivent être utilisées de cette manière, avec leur nom en majuscules, alors les noms de procédures seront toujours écrites en majuscules par le SuperBASIC et mettra en évidence leur fonction particulière à l'intérieur du programme et du système QL.

Des mots-clés existant déjà ne peuvent être utilisés comme des identifiants ordinaires dans un programme SuperBASIC.

OPERATEURS

OPERATORS

Opérateur	Type	Fonction
=	décimaux chaîne *	logique
==	numérique chaîne	presque égaux (équivalents) ** (comparaison : type 3)
+	numérique	addition
-	numérique	soustraction
/	numérique	division
*	numérique	multiplication
<	numérique chaîne	Plus petit que (comparaison : type 2)
>	numérique chaîne	Plus grand que (comparaison : type 2)
<=	numérique chaîne	Plus petit ou égal à (comparaison : type 2)
>=	numérique chaîne	Plus grand ou égal à (comparaison : type 2)
<>	numérique chaîne	différent de (comparaison : type 3)
&	chaîne	concaténation
&&	mode binaire	ET
	mode binaire	OU
^^	mode binaire	OU exclusif
~	mode binaire	NON négation
OR	logique	OU
AND	logique	ET
XOR	logique	OU exclusif
NOT	logique	NON négation
MOD	entier	Modulo
DIV	entier	Division entière
INSTR	chaîne	Comparaison type 1
^	décimaux	Elévation à la puissance
-	décimaux	Moins unitaire (changement de signe)
+	décimaux	Plus unitaire

* chaîne = chaîne de caractères

** égal à 10^{-7} près

Si une opération logique est vraie, alors une valeur non nulle est retournée.

Si une opération logique est fausse, alors une valeur nulle est retournée.

Priorités :

La priorité des opérateurs SuperBASIC est définie ci-dessous. Si l'ordre des opérations dans une expression ne peut être déduit de cette table, alors les opérations sont exécutées de gauche à droite. Les priorités du SuperBASIC peuvent être surmontées en introduisant des parenthèses appropriées.

priorités hautes + ou - unitaire
 concaténation de chaînes recherche de chaînes exponentiation multiplication
 ou division addition ou soustraction comparaison logique
 NON logique
 ET logique
 OU et OU exclusif logiques

ORDRES DU SUPERBASIC

STATEMENT

Un ordre du SuperBASIC est une instruction donnée au QL d'exécuter une opération spécifique, par exemple :

```
LET a = 2
```

qui donne la valeur 2 à la variable a.

Plusieurs ordres peuvent être écrits sur la même ligne en les séparant par le caractère (:), par exemple

```
LET a=a+2 : PRINT a
```

qui ajoute 2 à la valeur de la variable a et stocke le résultat dans a, puis le résultat est affiché.

Si une ligne n'est pas précédée par un numéro de ligne, alors il s'agit d'une commande directe qui sera exécutée immédiatement.

Si une ligne est précédée par un numéro de ligne, elle devient une partie d'un programme SuperBASIC.

Attention :

Certains ordres du SuperBASIC peuvent avoir un effet sur les ordres situés derrière sur la même ligne logique. Il est donc déconseillé d'utiliser certains ordres SuperBASIC dans une commande directe, exemple :

```
IF, FOR, REPEAT, REM, etc
```

PERIPHERIQUES

DEVICES

Un périphérique est un élément de l'installation du QL qui peut donner ou recevoir des données.

Etant donné que le système ne préjuge pas d'avance des périphériques I/O qu'il va utiliser, les périphériques I/O peuvent être facilement changés et les données réparties entre les périphériques. Par exemple un programme peut avoir besoin de sortir sur une imprimante pendant un certain temps de son exécution. Si l'imprimante n'est pas disponible, alors la sortie peut être aiguillée sur un fichier microdrive et stockée. Le fichier peut être imprimé plus tard. L'I/O (Entrée/Sortie) sur le QL doit être conçu comme pouvant être lu et imprimé par un fichier logique, qui est une forme indépendante du périphérique.

Toutes les opérations spécifiques des périphériques sont exécutées par un driver (pilote) de périphérique spécifique à chaque périphérique du QL. Le système sélectionnera automatiquement le driver adapté. Ils doivent être écrits dans le format standard de driver de QL. (Cf notion Extension périphérique)

Quand un périphérique est activé, un canal est ouvert et lié à ce périphérique. Pour ouvrir correctement un canal, l'information de base du périphérique doit être donnée pour initialiser le driver du périphérique. Cette information complémentaire est ajoutée au nom du périphérique (vg. le nom d'un fichier ou le nom driver de ce périphérique). L'ensemble du nom du périphérique et de l'information complémentaire s'appelle **nom de fichier**.

Le nom de fichier doit se conformer aux règles d'établissement de nom en SuperBASIC. Il est aussi possible de donner pour nom une chaîne de caractères.

La forme générale d'un nom de fichier est :

```
identifiant [_information])
```


Dans ce cas, le nom de fichier complet (comportant l'information complémentaire) se conforme aux règles de "nom" en SuperBASIC.

Chaque périphérique logique du système nécessite sa propre "information complémentaire", bien que des paramètres par défaut puissent servir dans tous les cas possibles.

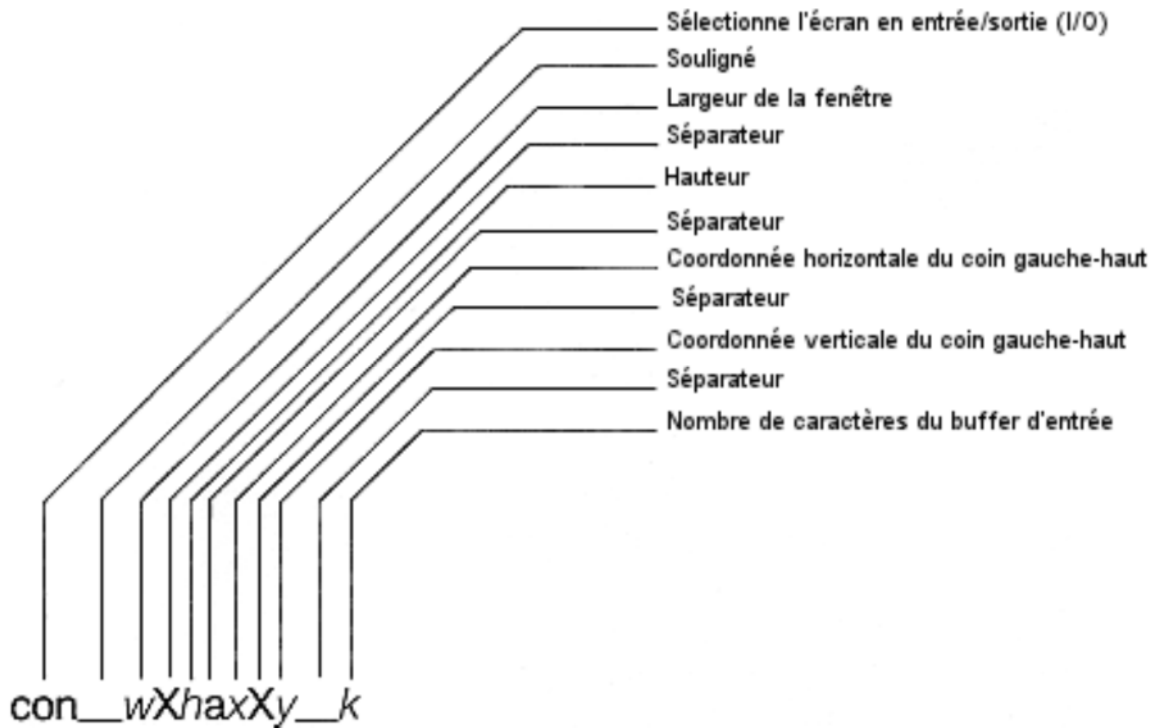
Définition

périphérique: = nom

quand la forme du nom de périphérique est soulignée.

Exemple

pour le périphérique de CONsole



CON_wXhaxXy_k Console I/O
 [wXh]
 [AxXy]
 [k]

Ecran clavier
 largeur et hauteur de fenêtre
 coordonnées x,y du début de fenêtre
 taille (en caractères du tampon de clavier)

valeurs par défaut : con_448x180a32x16_128

Exemples :

OPEN #4 , con_20x50a0x0_32
 OPEN #8 , con_20x50
 OPEN #7 , con_20x50a10x10

SER_nphz

Sortie : Série (RS-232-C)

n numéro de port (1 ou 2)

[*p*] parité

e - pair

o - impair

m - point

s - espace

[*h*] handshake

i - ignore

h - handshake

[*z*] protocole

r - donnée brute pas EOF

z - controle Z si EOF

c - comme z mais ENTREE est un séparateur d'enregistrement

valeurs par défaut : **ser1rh** (8 bit sans parité avec "handshake")

Valeurs par défaut `ser1r`
 `OPEN #3, ser1e`
 `OPEN #4, serc`
 `COPY mdv1test_file TO ser1c` (Voir aussi TR A dans mots-clefs)

NETd_s

Réseau I/O

[d] indique la direction *[s]* : le numéro de station
i – input 0 : envoi généralisé
o – output **own station** : sortie sur sa propre station pour écoute générale

valeurs par défaut : aucune

exemples : `OPEN #7, neti_32`
 `OPEN #4, neto_0`
 `COPY ser1 TO neto_21`

MDVn_name

Accès au microdrive de fichier

n - n° de Microdrive
name - nom du fichier

Pas de valeur par défaut

Exemples : `OPEN #9, mdv1_fichier_donnees`
 `OPEN #9, mdv1_program_test`
 `COPY mdv1_fichier_test TO scr_`

Mot-clé	Fonction
OPEN	Initialise un périphérique et active son utilisation
CLOSE	Désactive un périphérique
COPY	Copie de données entre différents périphériques
COPY_N	Copie des données entre différents périphériques – ne copie pas l'en-tête du fichier
EOF	Test de fin de fichier
WIDTH	Définit la largeur

PROGRAMME

PROGRAM

Un programme SuperBASIC consiste en une séquence d'instructions SuperBASIC où chaque instruction est précédée d'un numéro de ligne. Les numéros de lignes disponibles vont de 1 à 32767.

Syntaxe : **[numéro_de_ligne instruction *[: instruction]]**

Exemples :

```
i.      10 PRINT "Ce numéro de ligne est valable"
        STOP

ii.     10 REM un petit programme
        20 FOR couleur = 0 TO 7
        30   FOR contraste = 0 TO 7
        40     FOR trame = 0 TO 3
        50       PAPER couleur,contraste,trame
        60       CURSOR 0,70
        70       PRINT couleur !contraste !trame
        80       FOR pas = 0 TO 2
        90         SCROLL 2,1
        100        SCROLL -2,1
        110        END FOR pas
        120       END FOR trame
        130      END FOR contraste
        140     END FOR couleur
        STOP
```

Commande	Fonction
RUN	Démarre l'exécution d'un programme chargé
LRUN	Charge un programme à partir du périphérique spécifié et démarre son exécution
[CTRL] [SPACE]	Provoque l'arrêt d'un programme en cours d'exécution.

QDOS

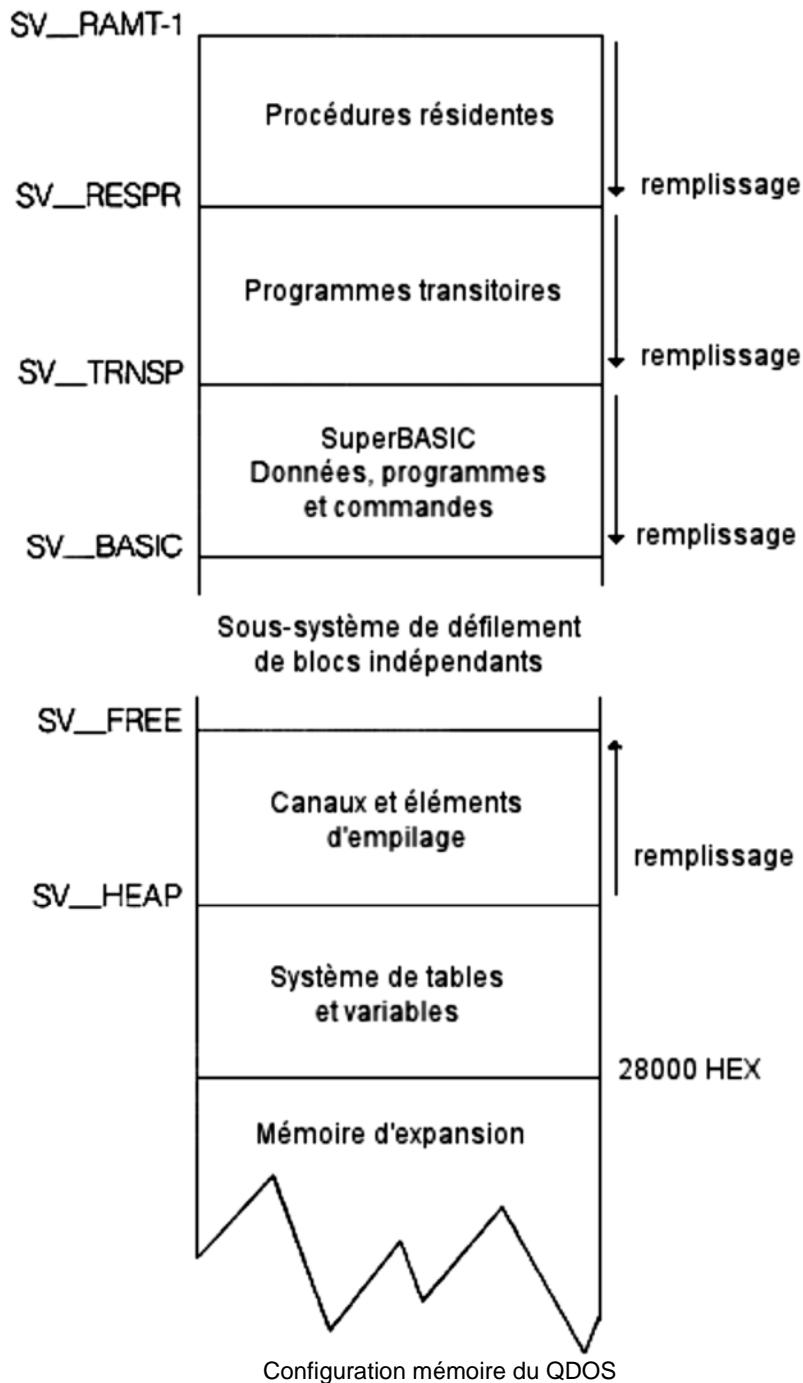
Le QDOS est le système d'exploitation du QL. Le QDOS gère :

- Les procédures et les allocations de ressource.
- Les entrées clavier et sorties d'écran (y compris les fenêtres)
- Les entrées-sorties micro-lecteur.
- Le réseau et les canaux de communication.
- La gestion de la mémoire.

Une description complète du QDOS sera fournie ultérieurement ; cependant une description sommaire du QDOS est donnée ici.

Configuration de la mémoire

La mémoire RAM a une organisation imposée par le système d'exploitation QDOS et est organisée comme suit :



Les appellations SV_RAMT, SV_RESPT, SV_TRNSP, SV_BASIC, SV_FREE, SV_HEAP sont utilisées pour représenter les adresses à l'intérieur du QL. Ces appellations ne sont pas reconnues par le SuperBASIC ou système d'exploitation QDOS. De plus les adresses représentées sont susceptibles de changer quand le système est actif.

sv_ramt Haut de la RAM

Ceci varie en fonction des cartes d'extension reliées à la mémoire.

sv_respt Procédures résidentes

Les procédures résidentes sont chargées en haut de la RAM. Une fois chargée, l'allocation de cette zone ne peut être modifiée que par une réinitialisation du système. Les procédures résidentes sont ajoutées à la liste des noms SuperBASIC et deviennent ainsi des extensions du système SuperBASIC.

sv_trnsp Programmes transitoires

Les programmes transitoires sont chargés immédiatement sous les procédures résidentes. Chaque programme doit être de taille suffisante, c'est à dire qu'il doit contenir de la place pour ses propres données et ses propres empilages. Il doit être en position indépendante et doit être chargé par un éditeur de lien spécialement écrit. Les programmes transitoires sont exécutés à partir du BASIC en utilisant la commande **EXEC** ou par le QDOS en le lançant comme un "job" (travail).

Les zones de programmes transitoires peuvent être utilisées pour stocker uniquement des données, mais celles-ci sont traitées par QDOS comme un "job" et, par conséquent, ne seront pas activées.

sv_basic Zone SuperBASIC

Cette zone contient tous les programmes chargés en SuperBASIC et les données correspondantes. Cette zone s'étend et se contracte en utilisant l'espace libre en fonction des besoins.

sv_free Espace libre

L'espace libre est utilisé par le sous-système des fichiers QDOS pour créer des blocs indépendants du microdrive (microlecteur), par exemple les copies des blocs du micro-drive qui doivent être stockées dans la RAM.

sv_heap Système d'empilage

Ceci est utilisé par le système pour stocker les données des canaux, les définitions, etc...

Il prévoit aussi une mémoire de travail pour le sous-système d'entrée-sortie. Les programmes transitoires peuvent allouer de l'espace de travail pour eux-mêmes, dans le "HEAP" par l'intermédiaire des commandes du QDOS.

Tables et variables - systèmes

Cette zone est juste au-dessous de la mémoire d'écran. Les tables "système" et le superviseur sont placés au-dessus des variables "système".

Appel au système

Les appels au système sont exécutés par le QDOS en mode superviseur. Quand il est dans ce mode, QDOS ne permet pas à un autre "job" d'utiliser le processeur. Les appels au système exécutés de cette manière sont dits "atomiques", c'est à dire qu'un appel au système va s'exécuter jusqu'à la fin afin de libérer le processeur. Certains appels "système" sont seulement partiellement "atomiques" dans le cas où ils libèrent le processeur si nécessaire seulement après avoir terminé leur fonction principale.

Sauf spécification particulière, les appels "système" d'E/S sont partiellement atomiques.

Le mécanisme standard pour faire un appel "système" consiste à récupérer un des vecteurs systèmes avec les paramètres appropriés dans les registres du processeur. L'action résultante dépend de l'appel lui-même et de l'état général du système au moment où l'appel est fait.

Entrée/Sortie

QDOS permet de travailler dans un environnement multitâche. Aussi peut-on accéder à un fichier par plusieurs tâches à la fois. Le sous-système de gestion de fichiers QDOS peut gérer des fichiers qui ont été ouverts comme exclusifs ou partagés.

On ne peut pas écrire dans un fichier partagé.

Les périphériques du QL sont gérés par le sous-système E/S série. Le sous-système de gestion de fichiers et le sous-système E/S série constituent ensemble le système E/S redirigeable. Comme son nom l'indique, n'importe quelle donnée de sortie de ce système peut être redirigée vers n'importe quel autre périphérique, également supporté par le système E/S redirigeable.

Les noms de périphériques requis pour QDOS sont les mêmes que les noms de périphériques requis par SuperBASIC. Ils sont expliqués dans le paragraphe "périphériques" des concepts QL. L'ensemble des périphériques standards gérés par QL peut être étendu.

Gestion multitâche

Les "jobs" (travaux) se voient allouer une partie du CPU avec leur priorité et concurremment avec les autres jobs du système. Les "jobs" tournent sous le contrôle du QDOS peuvent prendre l'un des trois états suivants :

- ACTIF** capable de s'exécuter et de partager les ressources du système. Un "job" dans cet état ne peut pas s'exécuter de façon continue mais se voit attribuer une partie du CPU en fonction de sa priorité.
- SUSPENDU** Ce "job" est capable de s'exécuter mais est en attente d'un autre job ou d'une E/S. Un job peut être complètement suspendu pour une période de temps déterminée.
- INACTIF** Ce "job" est incapable de s'exécuter. Sa priorité est 0 et ainsi il ne peut jamais obtenir une partie du CPU. QDOS réorganise le système automatiquement dans une forme en rapport avec la forme d'origine. Le système sera ainsi réorganisé après certains appels-systèmes.

Exemple

Ce programme génère un affichage à l'écran de l'horloge temps-réel, sous la forme d'un job indépendant.

Commencez par en lancer l'exécution en utilisant l'instruction RUN, après avoir disposé une cartouche formatée dans le microdrive 2.

Vous allez générer un programme en langage machine nommé "HORLOGE".

Attendez que le microlecteur de cartouches s'arrête puis réglez l'horloge en utilisant la commande SDATE.

Tapez alors :

```
EXEC mdv2_horloge
```

et un affichage continu de l'heure apparaîtra dans le coin supérieur droit de la fenêtre de commande.

```
100 c=RESPR(100)
110 FOR i = 0 TO 68 STEP 2
120   READ x:POKE_W i+c,x
130 END FOR i
140 SEXEC mdv2_horloge,c,100,256
```

```
1000 DATA 29439,29697,28683,20033,17402
1010 DATA 48,13944,200,20115,12040
1020 DATA 28691,20033,17402,74,-27698
1030 DATA 13944,236,20115,8279,-11314
1040 DATA 13944,208,20115,16961,16962
1050 DATA 30463,28688,20035,24794
1060 DATA 0,7,240,10,272,200
```

N.B.

La ligne 1060 gouverne la position et la couleur de la fenêtre de l'horloge voici, dans l'ordre, la signification des différents paramètres :

- couleur/largeur du bord,
- couleur papier/encre,
- largeur de la fenêtre,
- hauteur,
- origine en x,
- origine en y.

Il s'agit de paires d'octets, entrés sous la forme de mots en utilisant l'instruction POKE_W.

Nous suggérons pour l'origine en x et en y (les derniers paramètres) : 272,202 en mode moniteur et 240,216 en mode télévision.

Pour changer le papier il faut utiliser la formule : $256 * \text{papier} + \text{encre}$. Ainsi, $256 * 7 + 2 = 1794$ donnera un papier blanc et une encre rouge.

Périphériques

Les périphériques standards compris dans le système sont expliqués dans ce guide au paragraphe "périphériques". Des périphériques peuvent être ajoutés au système, prendre un nom, et alors être accessibles de la même façon que n'importe quel périphérique QL.

REPETITION

Les boucles du SuperBASIC sont contrôlées par deux constructions de base. Chaque construction doit être reconnue par SuperBASIC.

REPeat <i>identifiant</i>	FOR <i>identifiant = de... à</i>
<i>instructions</i>	<i>instructions</i>
END REPeat <i>identifiant</i>	END FOR <i>identifiant</i>

Ces deux constructions sont utilisées en conjonction avec deux autres constructions SuperBASIC.

NEXT <i>identifiant</i>	EXIT <i>identifiant</i>
--------------------------------	--------------------------------

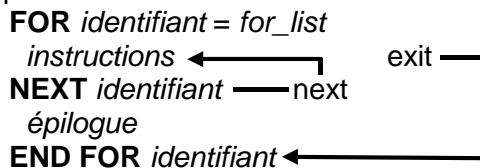
NEXT et **EXIT** peuvent être utilisés dans les deux constructions, en fait **EXIT** doit être utilisé dans la construction d'un **REPEAT**, sinon une boucle infinie en résultera.

L'exécution d'une instruction **NEXT** soit passera le contrôle à l'instruction suivant le **FOR** approprié (ou le **REPEAT**), soit si la limite de **FOR** a été atteinte à l'instruction suivant le **NEXT**.

L'exécution d'un **EXIT** passera le contrôle à l'instruction après le **END FOR** ou **END REPeat** choisis par l'identifiant après le mot-clé **EXIT**. Si un **EXIT** est utilisé dans une boucle, la boucle doit se terminer par **END FOR** ou **END REPeat**. **EXIT** peut être utilisé pour sortir à plusieurs niveaux de structures répétitives.

EXIT doit toujours être utilisé dans une boucle **REPeat** pour en sortir sous une condition donnée.

Une combinaison de **NEXT**, **EXIT** et **END** permet aux boucles **FOR** et **REPeat** d'avoir un épilogue de boucle. Un épilogue de boucle est une suite d'instructions SuperBASIC qui ne s'exécute que sous une condition particulière survenant dans la boucle.



L'épilogue de boucle est seulement exécuté si la boucle **FOR** se termine normalement. Si la boucle se termine par l'instruction **EXIT**, alors l'exécution continue au niveau du **END FOR** et l'épilogue n'est pas exécuté.

Il est possible d'avoir une construction identique dans une boucle **REPeat**.



Cette fois l'entrée dans l'épilogue de boucle est contrôlée par l'instruction **IF**. L'épilogue sera ou ne sera pas exécuté suivant la condition du **IF**.

Une instruction **SELection** peut aussi être utilisée pour contrôler l'entrée dans un épilogue.

RESEAUX

NETWORK

Le QL peut être connecté à 63 autres QL et/ou Spectrums. S'il y a plus de deux machines sur le réseau, alors chaque machine ou station doit se voir attribuer un numéro unique de station. Sur le QL ceci peut être fait en utilisant la commande **NET**.

L'information est transmise sur le réseau par blocs. Pour une communication normale entre 2 stations, la station réceptrice doit assurer une réception correcte des blocks. Si un block est altéré, alors la station réceptrice doit redemander une retransmission.

La station 0 est particulière, une sortie sur la station 0 est appelée "Broadcasting" (Envoi généralisé). N'importe quelle station qui est branchée sur la station 0 peut recevoir les données. Il n'y a pas alors de protocole de raccordement (handshaking). Si vous êtes en lecture de votre propre n° de station, vous recevez toutes les communications du réseau.

Commande	Fonction
NET	Attribue un n° de station réseau
OPEN	Ouvre un canal sur le réseau
CLOSE	Ferme un canal sur le réseau
PRINT	
INPUT	Permet des Entrées/Sorties sur le réseau
INKEY\$	
LOAD	
SAVE	
LBYTES	
SBYTES	
EXEC	Charge et sauve via le réseau
SEXEC	
LRUN	
MRUN	
MERGE	

SON

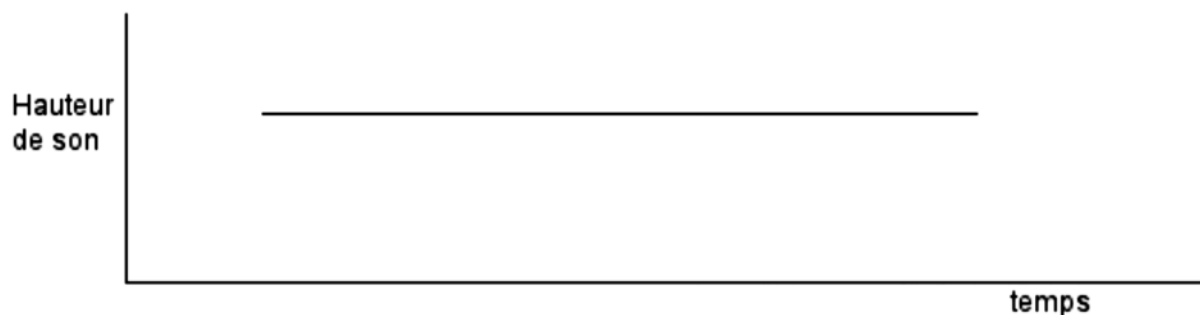
SOUND

Le son sur le QL est émis par le système IPC (8049) (le 2ème processeur) et est contrôlé en spécifiant :

- jusqu'à deux hauteurs de son
 - la vitesse à laquelle le son doit varier entre les deux hauteurs et l'enveloppe
 - des distributions et des dentelures (randomness, fuzziness).
- Les dentelures provoquent des bourdonnements tandis que les distorsions, suivant les autres paramètres, produiront un son mélodieux ou un bruit.

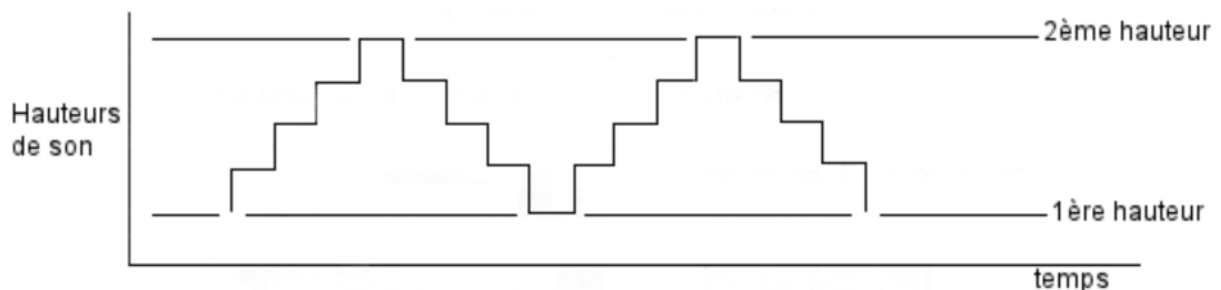
Niveau 1

On spécifie une durée et une hauteur uniques : la hauteur de son émise pendant le temps spécifié.



C'est la commande de son la plus simple, en dehors de la commande pour stopper le son sur le QL.

Niveau 2

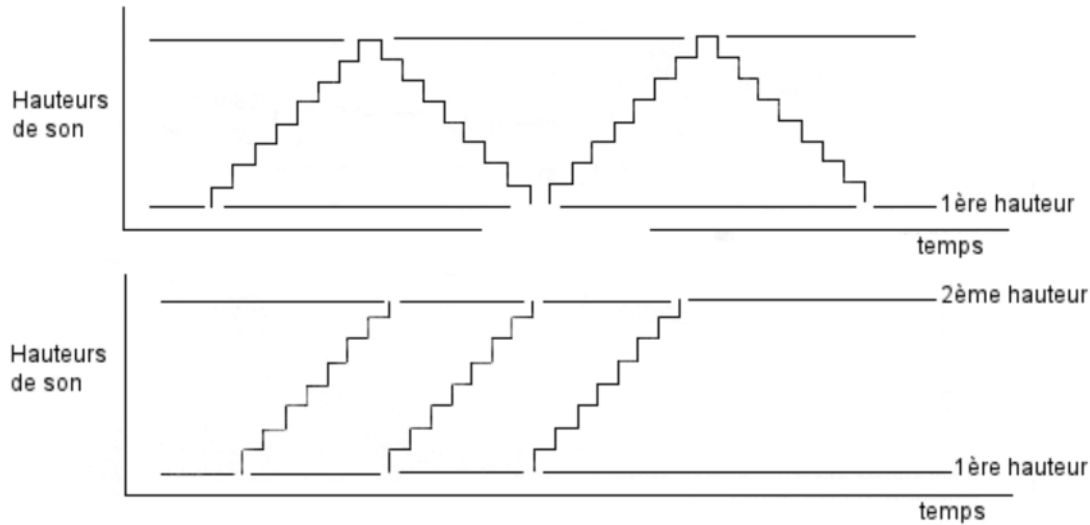


Une seconde hauteur et une pente peuvent être ajoutées à la commande. Le son rebondit alors entre les deux hauteurs à la vitesse spécifiée par la pente.

Le son produit à ce niveau peut varier entre un son semi-musical, un grognement, un gémissement, etc... Il est bon de faire des essais.

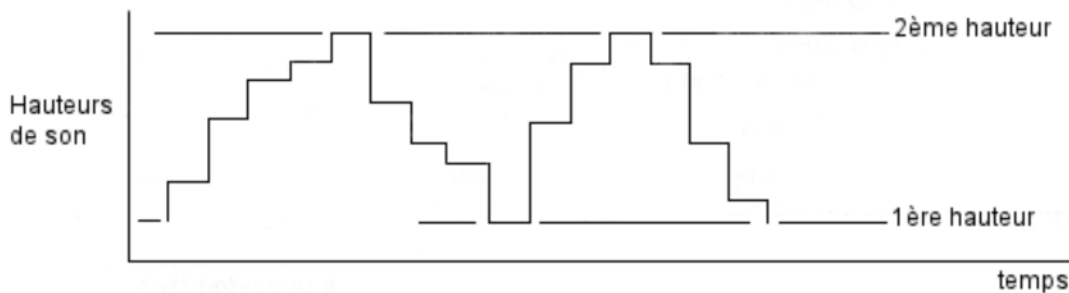
Niveau 3

Un paramètre peut être ajouté pour contrôler comment le son évolue quand il devient égal à une des hauteurs indiquées. Le son peut rebondir ou envelopper. Le nombre d'enveloppes peut être indiqué, y compris des enveloppes perpétuelles. Il est très important de faire des essais.



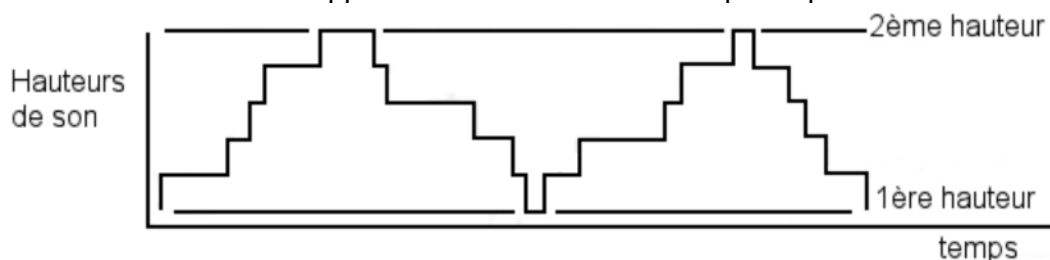
Niveau 4

On fait ajouter au son une distorsion. C'est une déviation d'un pas donné ou d'une pente donnée. Suivant la valeur de la distorsion par rapport aux hauteurs et aux pentes, cela générera une très large gamme de sons inattendus.



Niveau 5

On peut trouver plus de variétés en spécifiant les dentelures. Ceci ajoute un facteur aléatoire chaque fois que le son rebondit ou enveloppe. Les dentelures tendent à provoquer un bourdonnement.



En combinant tous les effets ci-dessus, on peut obtenir une large gamme de sons. Beaucoup sont inattendus. Le mieux est d'explorer ces sons par l'expérience. En spécifiant un intervalle de temps de 0 on fait répéter le son sans arrêt et une séquence de **BEEP** peut être utilisée jusqu'à ce que le son généré soit le son recherché.

Un mot de mise en garde : De petits changements dans la valeur d'un simple paramètre peut avoir des résultats alarmants sur le son produit. Pas de panique !

SYNTAXE - DEFINITIONS

SYNTAX - DEFINITIONS

La syntaxe du SuperBASIC est définie en utilisant un métalangage non rigoureux pour la notation. Quatre types de constructions sont utilisées :

	Sélection d'un élément
[]	Ce qui est entre crochet est optionnel
* *	ce qui est entre * * est répété
..	Etendue
{ }	Commentaire

Exemples :

A B	A ou B
[A]	A est optionnel
* A *	A est répété
A..Z	A, B, C... etc... Z
{ Ceci est un commentaire }	

Considérons un identifiant SuperBASIC

Une séquence de nombres, chiffres, soulignés, démarrent par une lettre et finit par un | optionnel ou un \$.

lettre := | **A..Z**
| **a..z**
La lettre est entre A et Z
ou entre a et z.

chiffre := | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
Le chiffre est entre 0 et 9 (inclus)

souligné := _

Identifiant = *lettre* **[lettre|chiffre|souligné]**

Doit démarrer avec une lettre ←
Séquence de lettres, chiffres, souligné

TABLEAUX

ARRAYS

Les tableaux doivent posséder utilisés. Quand un tableau dimensionné", la valeur de établie à partir de zéro, ou de chaîne de caractères.

Le tableau va de zéro à la La seule limite à la quantité de déterminée par la capacité de éléments d'un tableau sont positionnés de telle manière que le dernier index défini soit rapidement accessible.

	0	1	2	3	5
0	0,0	0,1	0,2	0,3	0,4
1	1,0	1,1	1,2	1,3	1,4
2	2,0	2,1	2,2	2,3	2,4

leur **DIM**ensions avant d'être possède ses dimensions ("est chacun de ses éléments est la position 0 s'il s'agit d'une valeur souhaitée.

nombres à définir est mémoire de l'ordinateur. Les

Exemple :

Le tableau défini par : DIM tableau(2,4)

sera enregistré ainsi :

Adresse (*) inférieure ,

← Adresse supérieure

L'élément **tableau(a,b,c,)** équivaut à l'élément **tableau (a) (b) (c)**

* Adresse : Position mémoire

Commande	Fonction
DIM	Dimensionne un tableau
DIMN	Indique les dimensions maximum d'un tableau

TABLEAUX DE CHAINES VARIABLES DE CHAINES

STRING ARRAYS STRING VARIABLES

Les tableaux de chaînes et les tableaux de données sont de même nature, bien qu'il y ait de légères différences dans le traitement en SuperBASIC. La dernière dimension d'un tableau de chaînes définit la longueur maxima de la chaîne dans le tableau.

Les variables de chaînes peuvent avoir n'importe quelle longueur (jusqu'à 32768 caractères).

On peut découper aussi bien les tableaux de chaînes que les variables de chaînes.

Les longueurs de chaînes de chaque côté d'une affectation ne sont pas forcément égales. Si les tailles sont différentes, alors la chaîne de droite est tronquée pour l'ajustement ou alors la longueur de la chaîne de gauche est réduite.

Si une affectation est faite vers une chaîne découpée, alors si nécessaire le trou défini par la tranche sera comblé par des espaces.

Il n'est pas nécessaire de définir la dimension finale d'un tableau de chaîne. Ne pas définir la dimension sélectionne l'ensemble de la chaîne tandis que la définition d'un simple élément sélectionnera un simple caractère et la définition d'une tranche définira une sous-chaîne.

Remarque

Contrairement à beaucoup de BASIC, SuperBASIC ne traite pas les tableaux de chaînes en longueur fixe. Si les données stockées dans un tableau de chaînes sont inférieures à la taille maximale du tableau, alors la longueur de la chaîne est réduite.

Attention

L'affectation de données à une tranche de tableau de chaînes ou à une variable de chaînes peut ne pas avoir l'effet désiré. Les affectations faites de cette façon ne mettent pas à jour la longueur de la chaîne et alors il est possible que le système ne reconnaisse pas l'affectation. La longueur d'un tableau de chaînes ou d'une variable de chaînes est seulement mise à jour quand une affectation est faite à l'ensemble de la chaîne.

TABLEAUX (DECOUPEGE DES)

SLICING

Dans certains cas il est possible de faire référence à plus d'un élément d'un tableau. Ceci revient à définir une série de sous-tableaux pour le SuperBASIC. Chaque tranche peut définir une séquence continue d'éléments pour une dimension particulière du tableau d'origine.

Dans ce contexte, le tableau inclut un tableau numérique, un tableau de chaîne ou une simple chaîne. Il est nécessaire de spécifier un index pour chaque dimension du tableau. Si une dimension est omise, alors des tranches sont ajoutées pour compléter l'ensemble des éléments.

Syntaxe

<i>index:=</i>	<i>numeric_exp</i>	(élément simple)
	<i>numeric_exp TO numeric_exp</i>	(partie comprise entre 2 éléments)
	<i>numeric_exp TO</i>	(de la position indiquée à la fin)
	<i>numeric_expression</i>	(du début à la position indiquée)

tableau_référence:= | *variable*
| *variable([index *, index]*)*

Une tranche de tableau peut être utilisée pour indiquer une source ou une destination de sous-tableau dans une instruction d'affectation.

Exemples :

```
i. PRINT tableau_donnée
ii. PRINT lettres$(1 TO 15)
```

Attention !

L'affectation d'une donnée ou d'une tranche de tableau en chaîne de caractères peut ne pas avoir l'effet désiré.

Les affectations faites de cette façon ne modifient pas la longueur de la chaîne et il est possible que le système ne fasse pas correctement l'affectation. La longueur d'un tableau de chaînes ou de variable de chaîne est seulement mise à jour quand l'affectation est faite sur l'ensemble de la chaîne.