sinclair

QL

Conceptos

La Guía de Referencia de Conceptos describe algunos de los conceptos que relacionan el SuperBASIC y el hardware del QL. Es mejor pensar en esta Guía de Conceptos como una fuente de información para las preguntas que surjan sobre el SuperBASIC o el propio QL cuando utilice el ordenador o consulte otras secciones de su manual. Es en estos casos donde esta Guía de Conceptos puede tener la respuesta. Los conceptos se han listado en orden alfabético utilizando el término más aparente para cada concepto. Si no encuentra usted el tema que desea, consulte el índice que le indicará a qué página debe referirse.

En los casos en los que los ejemplos se listan con números de línea, se trata de programas completos que pueden introducirse en el ordenador y ejecutarse. Los ejemplos listados sin números son simples comandos y no siempre pueden introducirse aislados. Los ejemplos que demuestran algunos tipos de mezclas de colores (stipples) no funcionan adecuadamente en los aparatos de televisión domésticos.

por Stephen Berry *(Sinclair Research Limited)*

arrays-matrices

Las matrices deben **DIM**ensionarse antes de su utilización. Cuando se dimensiona una matriz, cada uno de sus elementos se hace cero o si se trata de una matriz de cadena, su longitud se hace cero. La dimensión de una matriz va desde cero hasta el valor especificado. El único límite para el número de dimensiones de una matriz es la capacidad de memoria total del ordenador. Los datos de una matriz se almacenan de tal forma que el último índice definido es el que más rápidamente cambia.
La matriz definida por

ejemplo

DIM matriz(2,4)

se almacena como

dirección más baja

	0	1	2	3	4
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

dirección más alta

El elemento al que se refiere la **matriz (a,b,c)** es equivalente al referido por la **matriz (a) (b) (c)**.

Comando	Función
DIM	dimensiona una matriz
DIMN	averigua las dimensiones de una matriz especificada

BASIC

El SuperBASIC incluye la mayoría de las funciones, procedimientos y construcciones que se encuentran en otros dialectos del BASIC. Muchas de estas funciones son superfluas en SuperBASIC, pero se han incluido por razones de compatibilidad.

GOTO	use **IF, REPeat,** etc.
GOSUB	use **DEFine PROCedure**
ON GOTO	use **SELect**
ON GOSUB	use **SELect**

Algunos comandos no parecen estar presentes. Sin embargo, siempre pueden obtenerse utilizando alguna función más general. Por ejemplo, no existen las instrucciones **LPRINT** o **LIST** en SuperBASIC, pero las salidas pueden dirigirse a la impresora operando el canal adecuado y utilizando **PRINT** o **LIST.**

LPRINT	use **PRINT#**
LLIST	use **LIST#**
VAL	no es necesaria en SuperBASIC
STR$	no es necesaria en SuperBASIC
IN	no es aplicable al 68008
OUT	no es aplicable al 68008

comentario

Prácticamente todas las formas del **BASIC** necesitan las funciones **VAL(x$)** y **STR$(x)** para poder convertir la codificación interna del valor de una expresión de cadena, o desde la codificación interna del valor de una expresión numérica.

Estas funciones son redundantes en SuperBASIC porque dispone de una característica única que cubre estas necesidades: la "coerción" o cambio de tipos. Por esta razón no disponemos de estas funciones **VAL** y **STR$.**

break

Si en cualquier momento el ordenador falla y no responde, o si desea parar un programa de SuperBASIC puede hacer lo siguiente:

mantenga pulsada

CTRL

y luego pulse

ESPACIO

Para continuar un programa detenido por este método utilice el comando CONTINUE.

canales

Un *canal* es un medio mediante el cual los datos pueden sacarse o introducirse en un *dispositivo* QL. Antes de utilizar un canal, éste deberá activarse (o abrirse) con el comando **OPEN.** Algunos canales deben estar siempre activos, como los canales de omisión, y permiten las comunicaciones normales con el QL a través del teclado y de la pantalla. Cuando no se va a utilizar más un determinado canal, debe desactivarse (cerrarse) con el comando **CLOSE.**

Los diferentes canales se identifican por su número de canal consistente en una expresión numérica precedida por el signo #. Cuando se abre un canal, se une un *dispositivo* a un número de canal y el canal se inicializa. A partir de ese momento, el canal se identifica siempre únicamente por su número de canal. Veamos un ejemplo:

```
OPEN #5,SER1
```

Mediante esta instrucción, unimos la puerta en serie 1 al canal número 5. Para cerrar un canal sólo necesitamos especificar el número de canal. Por ejemplo:

```
CLOSE #5
```

Para abrir un canal es necesario que el *manejador de ese dispositivo* se encuentre activado. Normalmente el conductor de dispositivo puede activarse de varios modos, por ejemplo, una red local necesita un *número de estación.* Esta información adicional se añade al nombre del dispositivo y luego se pasa al comando **OPEN** en forma de parámetro. Véase el concepto *dispositivo* y *expansión periférica.*

Los datos pueden sacarse a una canal imprimiendo —**PRINT**ing— en ese canal. El mecanismo es semejante al que realiza la salida en la pantalla del QL. **PRINT** sin ningún parámetro envía la salida por el canal de omisión. Por ejemplo:

```
10 OPEN #5, mdv1_arch_prue
20 PRINT #5, "Este texto está en el archivo de prueba"
30 CLOSE #5
```

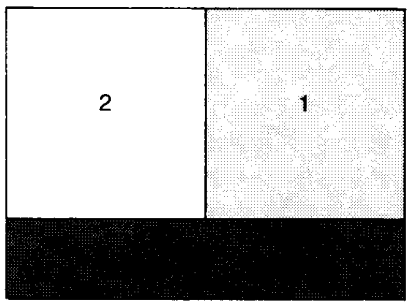
Con ello se obtendrá la salida "Este texto se encuentra en el archivo arch_prue". Es importante cerrar el archivo después de terminar los accesos a él. De esta forma nos aseguraremos de que todos los datos se han escrito.

Del mismo modo, los datos se pueden introducir en un archivo utilizando **INPUT.** La introducción se realiza desde un determinado canal, carácter por carácter utilizando **INKEY$.**

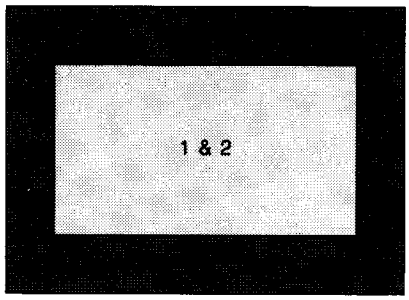
Un canal puede abrirse como canal de consola. La salida se dirige a una ventana específica de la pantalla del QL y la entrada se toma del teclado del QL. Cuando se abre un canal de consola, se especifica el tamaño y forma de la ventana inicial. Si se encuentran activos más de un canal de consola, se podrá tener más de una entrada a un mismo tiempo. El canal pedido puede seleccionarse pulsando CTRL C para ir pasando de un canal a otro de los que están en estado de espera. El cursor que se encuentre en la ventana del canal seleccionado estará intermitente.

El QL tiene tres canales de omisión que se abren automáticamente. Cada uno de esos canales se encuentra unido a una ventana determinada de la pantalla del QL.

canal 0 – canal de comandos y errores
canal 1 – canal de gráficos y salida
canal 2 – canal de listado de programas



Monitor



Television

Comando	Función
OPEN	abre un canal de E/S
CLOSE	cierra un canal abierto previamente
PRINT	da salida a una canal
INPUT	da entrada desde un canal
INKEY$	introduce un carácter desde un canal

conjunto de caracteres y teclas

Los controles del cursor no están necesariamente incluidos en el sistema operativo, sin embargo, si esas funciones se incluyen mediante aplicaciones de software, deberán utilizar las teclas específicas, y estas teclas específicas no deben utilizarse normalmente para otros fines.

Decimal	Hex	Teclas	Visualización/Función
0	00	CTRL ESC	Valor Nulo
1	01	CTRL A	
2	02	CTRL B	
3	03	CTRL C	Cambiar el canal de entrada
4	04	CTRL D	
5	05	CTRL E	
6	06	CTRL F	
7	07	CTRL G	
8	08	CTRL H	
9	09	TAB (CTRL I)	Campo siguiente
10	0A	↵ (CTRL J)	Nueva línea/entrada de comandos
11	0B	CTRL K	
12	0C	CTRL L	
13	0D	CTRL M	Enter
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 (CTRL SHIFT [)	Abortar al nivel de comandos urgente
28	1C	CTRL ⇧ –	
29	1D	CTRL ⇧]	
30	1E	CTRL ⇧ =	
31	1F	CTRL ⇧ ESC	
32	20	Espacio	Espacio
33	21	⇧ .	!
34	22	⇧ '	"
35	23	⇧ 3	#
36	24	⇧ 4	$
37	25	⇧ 5	%
38	26	⇧ 7	&
39	27	CTRL '	'
40	28	⇧ 9	(
41	29	⇧ 0	)
42	2A	⇧ 8	*
43	2B	⇧ =	+
44	2C	,	,
45	2D	–	–
46	2E	.	.
47	2F	⇧ 6	/
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	⇧ ;	:
59	3B	;	;
60	3C	<	<
61	3D	=	=
62	3E	⇧ <	>
63	3F	⇧ ,	?
64	40	CTRL 8	@

Decimal	Hex	Teclas	Visualización/Función
65	41	⇧ A	A
66	42	⇧ B	B
67	43	⇧ C	C
68	44	⇧ D	D
69	45	⇧ E	E
70	46	⇧ F	F
71	47	⇧ G	G
72	48	⇧ H	H
73	49	⇧ I	I
74	4A	⇧ J	J
75	4B	⇧ K	K
76	4C	⇧ L	L
77	4D	⇧ M	M
78	4E	⇧ N	N
79	4F	⇧ O	O
80	50	⇧ P	P
81	51	⇧ Q	Q
82	52	⇧ R	R
83	53	⇧ S	S
84	54	⇧ T	T
85	55	⇧ U	U
86	56	⇧ V	V
87	57	⇧ W	W
88	58	⇧ X	X
89	59	⇧ Y	Y
90	5A	⇧ Z	Z
91	5B	[	[
92	5C	CTRL 9	\
93	5D	] `	]
94	5E	⇧ backquote	^
95	5F	⇧ –	_
96	60	CTRL –	£
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 0	|
125	7D	CTRL [	}
126	7E	CTRL]	~
127	7F	⇧ ESC	©
128	80	CTRL ⇧ R	ä
129	81	CTRL ⇧ 1	â
130	82	CTRL ⇧ 2	à
131	83	CTRL ⇧ 3	é
132	84	CTRL ⇧ 4	ö
133	85	CTRL ⇧ 5	õ
134	86	CTRL ⇧ 6	ô
135	87	⇧]	ü
136	88	CTRL [	ç
137	89	Ñ	
138	8A	CTRL ⇧ 7	æ
139	8B	CTRL ⇧ 8	œ
140	8C	CTRL ,	á
141	8D	CTRL ⇧ 9	à
142	8E	CTRL .	â
143	8F	CTRL ⇧ S	ë

Decimal	Hex	Teclas	Visualización/Función
144	90	CTRL ⇧ Q	è
145	91	CTRL ⇧ L	ê
146	92	CTRL ⇧ M	ï
147	93	CTRL ⇧ N	í
148	94	CTRL ⇧ O	ì
149	95	CTRL ⇧ P	î
150	96	CTRL ⇧ T	ó
151	97	CTRL ⇧ I	ò
152	98	CTRL ⇧ ñ	ô
153	99	' U	ú
154	9A	yU	ù
155	9B	CTRL Ñ	ü
156	9C	CTRL ⇧ ,	ß
157	9D	CTRL <	¢
158	9E	CTRL ⇧ .	¥
159	9F	CTRL	'
160	A0	CTRL ⇧ <	Ä
161	A1	CTRL ⇧ A	Ã
162	A2	CTRL ⇧ B	Å
163	A3	CTRL ⇧ C	É
164	A4	CTRL ⇧ D	Ö
165	A5	CTRL ⇧ E	Õ
166	A6	CTRL ⇧ F	Ø
167	A7	CTRL ⇧ G	Ü
168	A8	CTRL ⇧ H	Ç
169	A9	SHIFT ñ	Ñ
170	AA	CTRL ⇧ J	Æ
171	AB	CTRL ⇧ K	Œ
172	AC	CTRL 1	α
173	AD	CTRL 2	δ
174	AE	CTRL 3	ϑ
175	AF	CTRL 4	λ
176	BO	CTRL 5	μ
177	B1	CTRL 6	π
178	B2	CTRL 7	φ
179	B3	⇧ 1	¡
180	B4	⇧ 2	¿
181	B5	CTRL ⇧ U	$
182	B6	CTRL ⇧ V	§
183	B7	CTRL ⇧ W	Ò
184	B8	CTRL ⇧ X	«
185	B9	CTRL ⇧ Y	»
186	BA	CTRL ⇧ Z	°
187	BB	CTRL ⇧ Ø	÷
188	BC	CTRL ⇧ ;	←
189	BD	CTRL ;	→
190	BE	CTRL ⇧ '	↑
191	BF	CTRL ⇧ `	↓
192	C0	IZDA.	Cursor izda. un carácter
193	C1	ALT IZDA.	
194	C2	CTRL IZDA	Borrar un carácter a la izda.
195	C3	CTRL ALT IZDA	
196	C4	⇧ IZDA	
197	C5	⇧ ALT IZDA	
198	C6	⇧ CTRL IZDA	
199	C7	⇧ CTRL ALT IZDA	
200	C8	DCHA	Cursor dcha. un carácter
201	C9	ALT DCHA	
202	CA	CTRL DCHA	Borrar un carácter a la dcha.
203	CB	CTRL ALT DCHA	
204	CC	⇧ DCHA	
205	CD	⇧ ALT DCHA	
206	CE	⇧ CTRL DCHA	
207	CF	⇧ CTRL ALT DCHA	
208	D0	ARRIBA	
209	D1	ALT ARRIBA	
210	D2	CTRL ARRIBA	
211	D3	CTRL ALT ARRIBA	
212	D4	⇧ ARRIBA	
213	D5	CTRL ALT ARRIBA	
214	D6	⇧ CTRL ARRIBA	
215	D7	⇧ CTRL ALT ARRIBA	
216	D8	ABAJO	
217	D9	ALT ABAJO	
218	DA	CTRL ABAJO	
219	DB	CTRL ALT ABAJO	
220	DC	⇧ ABAJO	
221	DD	⇧ ALT ABAJO	
222	DE	⇧ CTRL ABAJO	
223	DF	⇧ CTRL ALT ABAJO	

Decimal	Hex	Teclas	Visualización/Función
224	E0	MAYUSCULAS	
225	E1	ALT MAYUSCULAS	
226	E2	CTRL MAYUSCULAS	
227	E3	ALT CTRL MAYUSCULAS	
228	E4	⇧ MAYUSCULAS	
229	E5	⇧ ALT MAYUSCULAS	
230	E6	⇧ CTRL MAYUSCULAS	
231	E7	⇧ CTRL A MAYUSCULAS	
232	E8	F1	
233	E9	CTRL F1	
234	EA	⇧ F1	
235	EB	CTRL ⇧ F1	
236	EC	F2	
237	ED	CTRL F2	
238	EE	⇧ F2	
239	EF	CTRL ⇧ F2	
240	F0	F3	
241	F1	CTRL F3	
242	F2	⇧ F3	
243	F3	CTRL ⇧ F3	
244	F4	F4	
245	F5	CTRL F4	
246	F6	⇧ F4	
247	F7	CTRL ⇧ F4	
248	F8	F5	
249	F9	CTRL F5	
250	FA	⇧ F5	
251	FB	CTRL ⇧ F5	
252	FC	⇧ SPACE	
253	FD	⇧ TAB	
254	FE	⇧ ↵ ↕	
255	FF	Véase más abajo	

Los Códigos hasta el 20 hexadecimal son caracteres de control o caracteres no imprimibles. Las teclas alternativas se muestran entre paréntesis detrás de la tecla principal.

Observe que CTRL-C se utiliza por el Qdos, y no puede detectarse sin realizar cambios en las variables del sistema.

Observe que los códigos CO-DF son comandos de control del cursor.

La tecla ALT, pulsada junto con cualquier combinación diferente de las teclas del cursor o CAPSLOCK genera el código FF seguido por un byte que indica el código de la tecla que se hubiera obtenido si ALT no se hubiera pulsado. CTRL.

Observe que CAPSLOCK y CTRL-F5 se utilizan por el Qdos y no pueden detectarse sin utilizar para ello un software especial.

clock

El QL contiene un reloj con la hora real que funciona mientras el ordenador esté conectado.

El formato utilizado para la fecha y la hora es el formato ISO estándar

1983 ENE 01 12:09:10

Los datos individuales del año, mes día y hora pueden obtenerse troceando la cadena devuelta por el comando **DATE$**. El reloj funciona a partir de 1961 ENE 01 00:00:00

comentario

Para ver la descripción del fomato, véase BS5249: Part 1: 1976 y en su versión modificada, el Apéndice D2.1 Tabla 5 Serie 5 y el Apéndice E2 Tabla 6 Series 1 y 2.

Comando	Función
SDATE	Pone el reloj en funcionamiento
ADATE	Ajusta el reloj
DATE	Devuelve la fecha en forma de número
DATE$	Devuelve la fecha en forma de cadena
DAY$	Devuelve el día de la semana

coerción-cambio forzado de tipos

El lenguaje SuperBASIC convierte si es necesario el tipo de datos no utilizables en otro tipo que permita que la operación especificada se pueda realizar.

Los *operadores* utilizados determinan la conversión necesaria. Por ejemplo, si una operación necesita un parámetro de *cadena* y se introduce un parámetro numérico, el SuperBASIC convertirá primero dicho parámetro a parámetro tipo cadena. No siempre es posible la conversión de los datos a la forma necesaria, y si éstos no se pueden convertir, se enviará el correspondiente mensaje de error.

El tipo de un parámetro de *función* o de *procedimiento* puede también convertirse para que sea el correcto. Por ejemplo, el comando de SuperBASIC **LOAD** necesita un parámetro tipo *nombre* pero puede aceptar un parámetro tipo *cadena* que el propio procedimiento convertirá en el tipo adecuado. Esta modalidad de Coerción (cambio forzado de tipos) depende siempre del modo en que se implementa la función o procedimiento.

En el QL existe un orden natural para los tipos de datos. Véase la figura. Las *cadenas* son los tipos más generales ya que pueden representar nombres y números en coma flotante y *enteros*. La *coma flotante* no es tan general como las cadenas pero es más general que los enteros ya que los datos con coma flotante pueden representar datos enteros (casi exactamente). La Figura que se encuentra más abajo, muestra un diagrama del orden: los datos se pueden convertir siempre en dirección hacia arriba dentro de un diagrama, pero no siempre es posible moverlos en sentido descendente por el diagrama.

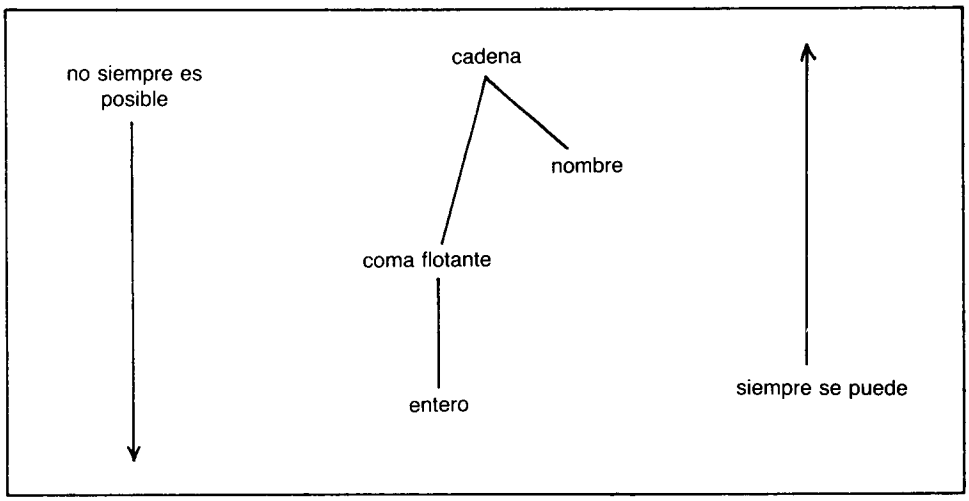


Fig. 1. Coerción

ejemplo

`a = b + c`	(no es necesario realizar la conversión antes de realizar la adición. La conversión no es necesaria antes de asignar el resultado a a.)
`a% = b + c`	(la conversión no es necesaria antes de realizar la suma, pero el resultado se convierte en entero antes de la asignación).
`a$ = b$ + c$`	(b$ y c$ se convierten en coma flotante, si es posible antes de sumarlos. El resultado se convierte en cadena antes de su asignación).

LOAD "mdv1_datos" (la cadena "mdv1_datos" se convierte en tipo nombre por el procedimiento load antes de su utilización.)

comentario En SuperBASIC se pueden escribir instrucciones que en muchos otros lenguajes generarían errores. Por lo general se pueden mezclar tipos de datos de forma muy flexible.

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

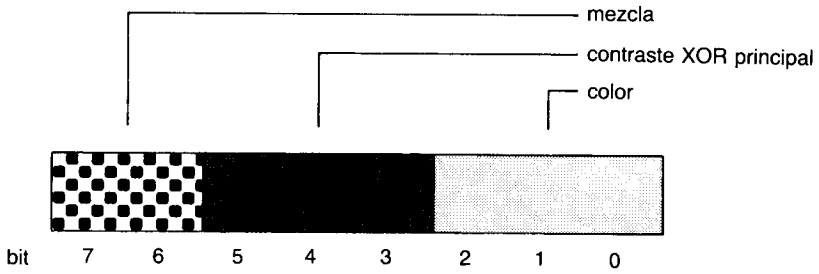
color

En el QL los colores pueden ser de dos tipos: **colores simplés** o **stripples**, es decir una mezcla de dos colores según un patrón predeterminado. De este modo, en el QL las especificaciones del color pueden indicar hasta tres elementos. Estos elementos son: un color, un color de contraste y un patrón de mezcla.

simples

color:= composición_color

Con un único argumento se indican las tres partes de las especificaciones del color. El color principal se incluye en los tres bits menos significativos del byte del color. Los tres bits contiguos contienen el 0 exclusivo (o XOR) del color principal y del color de contraste. Los dos bits más significativos indican el patrón de granulado (mezcla).



Si sólo se especifican los tres bits menos significativos (del color deseado) no se estará pidiendo mezcla alguna *(stipple)* y aparecerá un color simple en la visualización.

doble

color:= fondo, contraste

El *color* es la resultante de la mezcla de los dos colores especificados. Se supone como patrón de mezcla *(stipple)* omisión un patrón tipo ajedrez (mezcla 3).

triple

El *color de fondo,* y el *contraste,* así como el *patrón de mezcla* se definen cada uno de ellos separadamente.

colores

Los códigos para la selección de colores dependen del modo de pantalla que se encuentre vigente en ese momento.

código	patrón (bit)	composición	color	
			8 colores	4 colores
0	0 0 0		negro	negro
1	0 0 1	azul	azul	negro
2	0 1 0	rojo	rojo	rojo
3	0 1 1	rojo + azul	magenta	rojo
4	1 0 0	verde	verde	verde
5	1 0 1	verde + azul	cyano	verde
6	1 1 0	verde + rojo	amarillo	blanco
7	1 1 1	verde + rojo + azul	blanco	blanco

Composición de los colores y códigos

stipples-granulación

Las mezclas se realizan con dos colores, uno de fondo y otro de contraste formando un patrón fino de mezcla. Estas mezclas se pueden utilizar en el QL del mismo modo que si fueran colores simples normales, aunque no se reproducen correctamente en los aparatos domésticos normales de televisión.

El QL dispone de cuatro patrones de mezcla diferentes:

Stipple-mezcla 3 es el patrón-mezcla de omisión.

ejemplo

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

advertencia En los aparatos de televisión domésticos (UHF) no se reproducen las mezclas (stipples) correctamente.

comunicaciones RS-232-C

El QL dispone de dos puertas serie (llamadas SER1 y SER2) para las conexiones del equipo que utilicen comunicaciones en serie según el tipo EIA estándar RS-232-C u otro estándar compatible.

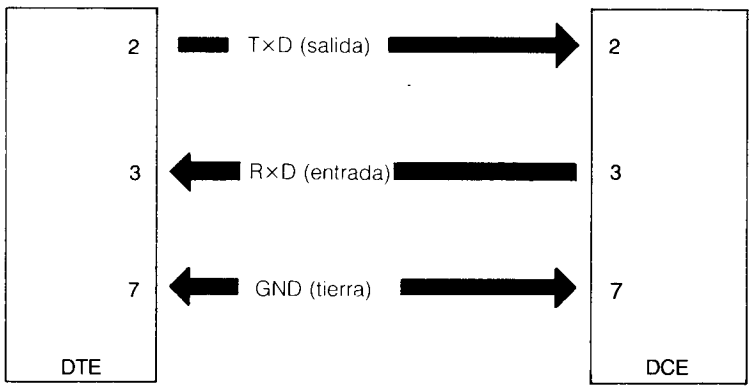
El RS-232-C "standard" se diseñó originalmente para permitir que los ordenadores pudieran enviar y recibir datos a través de las líneas del teléfono, utilizando modems para ello. Sin embargo, en la actualidad se utiliza con frecuencia para conectar ordenadores directamente unos con otros así como con otros elementos de equipos periféricos, como pueden ser impresoras, plotters, etc.

Como el RS-232-C "standard" se manifiesta en multitud de formas diferentes en las distintas partes del equipo, conectar dos partes de este supuesto "standard" RS-232-C, incluso para un experto resulta un trabajo muy complejo, sobre todo cuando es la primera vez que se realizan estas conexiones. Esta sección intentará cubrir la mayoría de los problemas básicos que puede encontrar el usuario.

El RS-232-C "standard" une dos tipos de equipo:

1. Equipos Terminales de Datos (DTE)
2. Equipos de Transmisión de Datos (DCE)

Lo normal es que tanto el terminal (normalmente DTE) como el modem (generalmente DCE) tengan ambos el mismo tipo de conector.



El diagrama anterior ilustra cómo transmite los datos el DTE en la patilla (pin) 2 (que sigue llamándose transmisión de datos). Del mismo modo, el DTE recibe los datos en la patilla (pin) 3 mientras que el DCE debe transmitir datos en su pin 3 (que continúa llamándose receptor de datos). Todo esto es muy confuso y puede llevar a problemas en ciertos casos en los que algún dispositivo puede configurarse como DCE o DTE.

Desafortunadamente, algunos usuarios deciden que sus ordenadores se configuren como dispositivos DCE mientras que otros configuran ordenadores equivalentes como dispositivos DTE. Esto obviamente lleva a dificultades en la configuración de las puertas serie en cada equipo.

El SER1 del QL está configurado como DCE mientras que el SER2 se ha configurado como DTE y por tanto se puede conectar al menos una de las puertas serie a un dispositivo dado simplemente utilizando cualquier puerta que se encuentre conectada del modo "correcto". A continuación se indica el diagrama de conexión (pin-out) de las puertas serie. Hay disponible un cable de conexión del QL con un conector tipo "D" estándar de 25 vías.

SER1		
pin	nombre	función
1	GND	tierra
2	T×D	entrada
3	R×D	salida
4	DTR	disponibilidad de recibir
5	CTS	disponibilidad de emitir
6		+ 12 V

T×D Transmite Datos
R× Recibe Datos

SER2		
pin	nombre	función
1	GND	señal de tierra
2	T×D	salida
3	R×D	entrada
4	DTR	salida inmediata
5	CTS	entrada inmediata
6		+ 12 v

DTR Terminal de Datos Preparado
Data Terminal Ready
CTS Listo Para enviar Clear To Send

Una vez que se haya conectado el equipo a la puerta "correcta", la relación de baudio (velocidad de transmisión de los datos) debe establecerse de modo que sea la misma en el QL y en el equipo que se conecte. El QL puede operar con las siguientes relaciones:

75
300
600
1200
2400
4800
9600
19200 baudios (sólo en transmisiones)

La velocidad de transmisión (baudios) en el QL se establece mediante el comando **BAUD** y se fija para los dos canales. Las velocidades de transmisión (baudios) no pueden establecerse independientemente.

La **paridad** que se establezca para el QL debe concordar también con la del equipo periférico. Esta paridad se utiliza para detectar errores simples en la transmisión, y puede ser par, impar, mark, espacio o sin paridad, es decir, los 8 bits de cada byte se utilizan con los datos.

Los bits de stop marcan el final de la transmisión de un byte o carácter. El QL siempre recibe sus datos con uno, uno y medio o dos bits de stop, y siempre transmite los datos con al menos dos bits de stop. Observe que si se activó el QL a una velocidad de transmisión de 9.600 baudios, no recibirá con un bit de stop necesitará al menos 1 1/2 bits.

Puede ser necesario conectar las líneas de concordancia **(handshake)** entre el QL y el equipo que se encuentre conectado con él. Esto permite al QL y a sus periféricos controlar su velocidad de transmisión. Todo esto puede ser necesario si uno de ellos no puede alcanzar la velocidad a la que se están transmitiendo los datos. El QL utiliza dos líneas de concordancia (handshaking).

CTS Clear to Send. Listo para Enviar
DTR Data Terminal Ready. Terminal de Datos Preparado

Si el DTE no puede alcanzar la relación de transmisión de los datos, puede negar la línea DTR que informa al DEC que debe parar la transmisión de los datos. Obviamente, cuando el DTE ha alcanzado esta relación, informa al DCE a través de la línea DTR que comience de nuevo la transmisión. Del mismo modo, el DCE puede parar el envío de datos del DTE negando la línea CTS. Si se requieren otras señales adicionales de control, pueden cablearse utilizando la alimentación de 12 V disponible en ambas puertas serie.

Aunque pueden llevarse a cabo comunicaciones sin ningún tipo de concordan-

cia — handshaking — el QL no recibirá correctamente en cualquier circunstancia si no se utilizan CTS en la puerta SER1 y DTR en la puerta SER2.

Las comunicaciones en el QL son en "full duplex", es decir, se puede recibir y transmitir a la vez.

La paridad y la concordancia — handshaking — se seleccionan cuando se abre el canal serie.

comando	función
BAUD	establece la velocidad de transmisión
OPEN	abre los canales en serie
CLOSE	cierra los canales serie

* Véase el concepto de *dispositivo* para una especificación completa.

TIPOS DE DATOS VARIABLES

enteros

Enteros son números de la gama de −32767 a +32768. Las variables se suponen enteras cuando el identificador de las variables lleva un signo de porcentaje % como sufijo. En el lenguaje SuperBASIC no existen constantes enteras, todas las constantes se almacenan como números de coma flotante.

sintasis: *identificador%*

ejemplo:
i. **contador%**
ii. **tamaño_límite %**
iii. **esta_variable_entera%**

coma flotante

Los números con coma flotante de la gama de $\pm(10^{-615}$ to $10^{+615})$ con 8 dígitos significativos. En el lenguaje SuperBASIC el tipo de datos de omisión es la coma flotante. Todas las constantes se guardan en esta forma y se pueden introducir utilizando notación exponencial.

sintaxis: *identificador* | *constante*

ejemplo:
i. **ventas_diarias**
ii. **76.2356**
iii. **354E15**

cadena

Una cadena es una secuencia de caracteres válidos en SuperBASIC cuya longitud puede llegar hasta 32.768 caracteres. Las *variables* se suponen tipo cadena si su nombre lleva el signo del dólar $ como sufijo. Las cadenas de datos se representan encerrando la adecuada selección de caracteres de SuperBASIC entre comillas simples o dobles.

sintaxis: *identificador$*

ejemplo:
i. **variables_de_cadena$**
ii. **«esto es una cadena de datos»**

nombre

El tipo nombre tiene la misma forma que un *identificador* estándar del SuperBASIC y se utilizan por el sistema para nombrar los *archivos* de *Microdrive*, etcétera.

sintaxis: *identificador*

ejemplo:
i. **mdv1_arch_datos**
ii. **ser1e**

dispositivos

Un **dispositivo** es una parte del equipo del QL al cual se pueden enviar datos y del cual se pueden obtener datos.

Puesto que el sistema no hace suposiciones sobre el dispositivo Entrada/Salida que se utilizará, este dispositivo de E/S puede cambiarse con facilidad y los datos se pueden diverger entre los dispositivos. Veamos un ejemplo. Un programa puede desear dar una salida a la impresora en algún momento de su ejecución. Si la impresora no está disponible puede diverger la salida a un archivo de *Microdrive* y almacenar los datos. Este archivo puede imprimirse en otro momento. Las E/S en el QL pueden considerarse como susceptibles de escribirse en ellas así como de leerse desde un **archivo lógico** que es una forma de dispositivo estándar.

Todas las operaciones para un dispositivo específico las realiza un **manejador individual** que se ha escrito especialmente para cada dispositivo del QL. El sistema buscará e incluirá automáticamente los manejadores adecuados; y éstos deben escribirse en el formato de manejador de dispositivos estándar del QL. Véase el concepto *expansión periférica.*

En el momento en el que se activa un dispositivo, se abre un *canal* y se une al dispositivo. Para abrir correctamente un canal, debe introducirse una información para inicializar el manejador del dispositivo. Esta información adicional se añade al nombre del dispositivo.

El nombre de archivo debe cumplir las normas del lenguaje SuperBASIC respecto al *tipo* ya que también se puede construir el nombre de archivo como una *expresión de cadena* de SuperBASIC.

Resumiendo, la forma general de un nombre de archivo es la siguiente:

identificador [*información*]

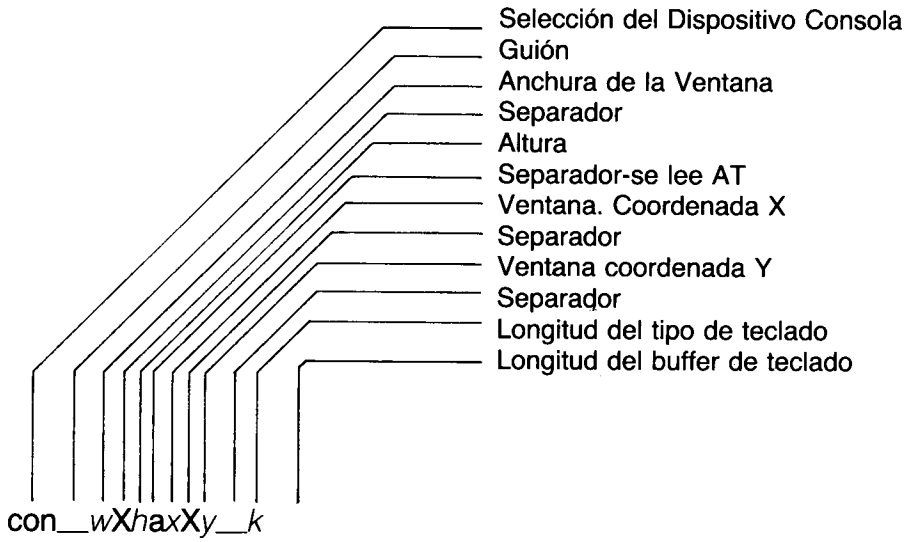
donde el nombre de archivo completo (incluyendo la información adicional) cumple las reglas de un nombre de SuperBASIC.

Cada dispositivo lógico del sistema necesita su «información adicional» particular, aunque se supondrán los parámetros de omisión en los casos en los que sea posible.

dispositivo: = nombre **define**

donde la forma del nombre del dispositivo se subraya.

para el dispositivo CONsole-Consola **ejemplo**



CON_*wXhaxXy*_*k* Consola E/S

[*wXh*] — ventana, anchura altura
[A*x*X*y*] — ventana, coordenadas *X Y*
[k] — longitud en bytes del buffer de teclado

valores de omisión: con_448×180a32×16_128

ejemplo.
```
OPEN#4, con_20×50a0×0_32
OPEN#8, con_20×50
OPEN#7, con_20×50a10×10
```

[xX*y*] — ancho de ventana, altura
[A*x*X*y*] — ventana, coordenadas *X Y*

valor de omisión: scr_448×180a32×16

ejemplo.
```
OPEN#4, scr_10×10a20×50
OPEN#5, scr_10×10
```

SER*nphz* Serie(RS232-C)
n número de puerta (1 ó 2)

[*p*] paridad
- e-par
- o-impar
- s-espacio
- m-mark

[h] (handshaking) (concordancia)
- i-ignora
- h-handshake

[*z*] protocolo
- r - fila de datos no EOF
- z - el control z es EOF
- c - igual que z pero convierte el carácter ASCII 10 (en el Qdos carácter de comienzo de nueva línea en el carácter ASCII 13 <CR>

omisión: **ser1rh** (8 bit sin paridad con concordancia-handshake)

ejemplo:
```
OPEN#3, ser1e
OPEN#4, ser1c
COPY mdv1_arch_test TO ser1c
```

NETd_s **Red Local Serie. I/O**
[*d*] indica la dirección,
- i-entrada;
- O-salida

s el número de la estación
- O - difusión;
- **Estación Propia** para escucha general únicamente

Omisión: no hay omisión

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

MDVn_*name* MDVn-nombre Acceso a Archivo de Microdrive,
n número de Microdrive,
nombre, nombre del archivo de Microdrive.

Omisión: no hay omisión

Ejemplo:
```
OPEN #9, mdv1_arch_datos
OPEN #9, mdv1_prog_test
COPY mdv1_arch_prueba TO scr_
```

Mando	Función
OPEN	inicializa un dispositivo y lo activa para su uso
CLOSE	desactiva un dispositivo
COPY	copia datos entre dispositivos
COPY-N	copia datos entre dispositivos pero no copia la información de la cabecera del archivo
EOF	comprueba si se ha llegado al final de un archivo
WIDTH	establece la anchura

comandos directos

El SuperBASIC diferencia entre una instrucción tecleada y precedida por un número de línea, y una instrucción tecleada sin número de línea. Si la instrucción no lleva números de línea se trata de un **comando directo** y por tanto se procesará inmediatamente por el **intérprete de comandos** del SuperBASIC. Por ejemplo, **RUN** se teclea en la línea de comandos y se procesa. Su efecto es que el programa comienza a ejecutarse. Si las instrucciones se introducen con números de línea se comprobará la sintaxis de la línea y se informará al usuario de los posibles errores cometidos. Las líneas que son correctas se introducen en el programa de SuperBASIC y se almacenan. Estas instrucciones constituyen un *programa* de SuperBASIC y sólo se ejecutarán cuando el programa se inicialice mediante un comando **RUN** o **GOTO.**

Observe que todas las instrucciones de SuperBASIC tienen sentido cuando se introducen como comandos directos. Por ejemplo, **END FOR, END DEFine,** etcétera.

manejo de los errores

El SuperBASIC informa de los errores en una forma estándar:

En la línea_*número*_*de*_*línea texto del error*

donde el número de línea es el número de la línea donde se detectó el error. El texto del error se lista a continuación.

(1) **No finalizado**
La operación se terminó prematuramente

(2) **Tarea inválida**
Error que parte de Qdos relativo a las llamadas al sistema (llamadas que controlan la función multitarea o las E/S).

(3) **Sin Memoria**
El Qdos y/o el SuperBASIC tienen memoria insuficiente.

(4) **Fuera de rango**
Normalmente se produce por intentar escribir fuera de una ventana o porque un índice de una matriz es incorrecto.

(5) **Buffer llena**
Una operación de E/S que va llenando de caracteres una memoria intermedia y llena dicha memoria antes de encontrar el carácter de fin de registro.

(6) **Canal no abierto**
Se intentó leer, escribir o cerrar un canal que no se había abierto.

(7) **No encontrado**
Un archivo, sistema o dispositivo medio, etc no se encuentra.
El lenguaje SuperBASIC no puede encontrar un identificador. Puede deberse a que las estructuras se anidaron de forma incorrecta.

(8) **Ya existe**
El archivo del sistema encontró un archivo ya existente con el mismo nombre que el archivo nuevo que se iba a abrir.

(9) **En uso**
El sistema de archivo ha encontrado que el dispositivo o archivo se está utilizando en ese momento con carácter exclusivo.

(10) **Fin de fichero**
Se detectó el final del archivo durante la entrada.

(11) **Drive lleno**
El dispositivo Microdrive se ha llenado.

(12) **Nombre incorrecto**
El archivo del sistema ha reconocido el nombre pero se detecta un error en la sintaxis o en el valor del parámetro.

(13) **Error de transmisión**
Error de paridad RS232-C.

(14) **Fallo en inicialización**
La operación de formato falló posiblemente debido a que el medio es erróneo.

(15) **Parámetro incorrecto**
Existe un error en la lista de parámetros del sistema o en el procedimiento de SuperBASIC o en la llamada a la función.
Se intentó leer datos de un dispositivo sólo de escritura.

(16) **Medio incorrecto**
El medio es posiblemente erróneo.

(17) **Expresión errónea**
Se detectó un error cuando se evaluaba una expresión.

(18) **Desbordamiento**
Overflow aritmético, división por cero raíz cuadrada de un número negativo.

(19) **No implementado**

(20) **Solo lectura**
Se intentó escribir datos en un archivo sólo lectura (protegido).

(21) **Línea incorrecta**
Aparece un error de sintaxis en SuperBASIC
Este error puede también ocurrir después de otro error en una función de una llamada a un procedimiento. Evita que se escriban líneas que modifiquen el programa. Los comandos NEW y CLEAR modifican esta condición.

(22) **PROC/FN inicializados**
Es un mensaje sólo para informar, no un mensaje de error. Informa que se ha parado el programa y a continuación se ha cambiado forzando al SuperBASIC a reinicializar su estado interno a otro nivel de programa. Por tanto habrá perdido toda la información correspondiente al procedimiento que estaba en uso anteriormente.

Recuperación de errores

Después de la aparición de un error, el programa puede reinicializarse en la línea siguiente tecleando

```
CONTINUE
```

Si la condición que causó el error se puede corregir sin cambiar el programa, éste puede continuar en la línea siguiente tecleando

```
RETRY
```

expresiones

Las expresiones en SuperBASIC pueden ser *cadenas,* expresiones *numéricas* y expresiones *lógicas,* o bien una mezcla. En los casos en los que es posible, los tipos de los datos que no son adecuados se convierten automáticamente a la forma adecuada por el sistema.

definición

monop:= | +
| –
| NOT

expresión:= | [*monop*] *expresión operador expresión*
| (*expresión*)
| *átomo*

átomo:= | *variable*
| *constante*
| *función* [(*expresión**[,*expresión*]*)]
| *elemento_de_una_matriz*
| *delimitador (slicer)*

variable:= | *identificador*
| *identificador %*
| *identificador $*

función:= | *identificador*
| *identificador %*
| *identificador $*

constante:= | *dígito**[*dígito*]*
| *[*dígito*]* . *[*dígito*]*
| *[*dígito*]* [.] *[*dígito*]* E *[*dígito*]*

El valor final devuelto al evaluar la expresión puede ser entero dando una **expresión_entera,** de cadena dando una **expresión_de_cadena** o de coma flotante, dando una **EXPRESION FLOTANTE**. Con frecuencia, las expresiones de coma flotante y enteras son equivalentes. En este caso, se utiliza el término **expresión_numérica.**

Se pueden incluir los operadores lógicos en la expresión especificada. Si la expresión es verdadera, el sistema devolverá un 1 como valor de la operación. Si es falsa, el valor devuelto será un cero. Aunque los operadores lógicos pueden utilizarse en cualquier expresión, normalmente se suelen utilizar en expresiones que forman parte de las instrucciones **IF.**

ejemplo:

i. `prueba_datos + 23.3 + 5`
ii. `"abcdefghijklmnopqrstuvwxyz"(2 TO 4)`
iii. `32.1 * (color = 1)`
iv. `cuenta = +límite`

tipos de archivo archivos

En el QL, todas las entradas/salidas se realizan a o desde *archivos lógicos.* Existen varios tipos de archivos:

data

Programas, archivos de texto, en SuperBASIC. Pueden almacenarse utilizando los comandos **PRINT, SAVE** y cargarse utilizando **INPUT, INKEY, LOAD,** etc.

exec

Programa ejecutable que se carga temporalmente. Se almacena utilizando **SEXEC,** y se carga utilizando **EXEC, EXEC_W,** etc.

code

Fila de datos de memoria, imágenes de pantalla, etc. Se almacena utilizando **SBYTES** y se carga mediante **LBYTES.**

funciones y procedimientos

En el lenguaje SuperBASIC, las *funciones* y *procedimientos* se definen mediante las instrucciones **DEFine, FUNction** y **DEFine PROCedure.** Para activar a una función (o llamarla) es necesario teclear su nombre en el punto adecuado de la expresión de SuperBASIC. La función debe estar incluida en una expresión porque nos devolverá un valor y ese valor debe utilizarse. Los procedimientos se activan (se llaman) tecleando su nombre como primer elemento de una instrucción de SuperBASIC.

Los datos pueden introducirse en una función o procedimiento añadiendo una lista de *parámetros actuales* detrás del nombre de la función o del procedimiento. Esta lista se compara con otra lista similar que se añadió detrás del nombre de la función o del procedimiento, en el momento de su definición. Esta segunda lista se conoce con el nombre de *parámetros formales* de la función o del procedimiento. Los parámetros formales deben ser variables en SuperBASIC. Los parámetros actuales deben ser una *matriz,* una parte de esta *matriz* o una *expresión* de SuperBASIC en la que la variable única o la constante tiene la forma más simple posible.

Como los parámetros actuales son expresiones actuales, deben tener un tipo actual asociado con ellos. Los parámetros formales sólo se utilizan para indicar la forma en la que los parámetros actuales deben procesarse y, por tanto, no tienen ningún tipo asociado a ellos. Los elementos de cada lista de parámetros se relacionan de dos en dos en orden cuando se llama a la función o al procedimiento, y los parámetros formales se hacen equivalentes a los parámetros actuales. Hay tres modos diferentes de utilizar parámetros:

Si el parámetro actual es una variable única, cuando los datos se asignen al parámetro formal en la función o procedimiento, estos datos también se asignarán al parámetro actual correspondiente.

Si el parámetro actual es una expresión, cuando se asignen los datos al parámetro formal correspondiente, no tendrán efecto fuera del procedimiento. Es importante recordar que una variable puede hacerse expresión sin más que encerrarla entre paréntesis.

Si el parámetro actual es una variable pero no se estableció previamente, los datos asignados al parámetro formal correspondiente establecerán esta variable específica como parámetro actual.

Las variables pueden definirse como locales a una función o procedimiento mediante la instrucción **LOCAL.** Estas variables locales no tienen efecto en variables con el mismo nombre que se encuentren fuera de la función o procedimiento en el que se han definido y, por tanto, dan una gran libertad en la elección de nombres adecuados para las variables, evitando que se "deterioren" las variables externas. Las variables locales pueden introducirse en cualquier función o procedimiento a no ser que la función o el procedimiento llamado contenga otra declaración posterior del mismo nombre de varible como local.

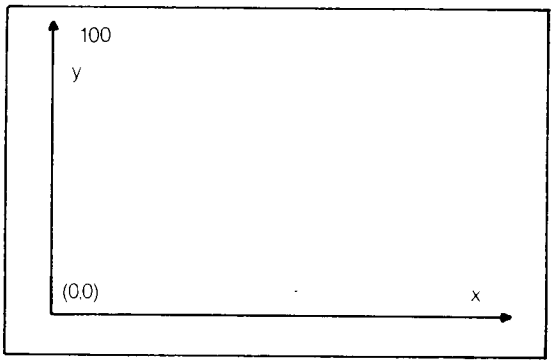
En el lenguaje SuperBASIC las funciones y procedimientos se pueden utilizar recursivamente. Esto quiere decir que una función o un procedimiento se puede llamar a sí mismo bien directa o indirectamente.

Comando	Función
DEFine FuNction	**DEFine una función**
DEFine PROCedure	define un procedimiento abandona la función o procedimiento
RETurn	(devuelve datos de la función)
LOCal	define como locales ciertos datos de una función o procedimiento

Es importante darse cuenta que la pantalla del QL tiene pixels que no son cuadrados, y que al cambiar el modo cambiará la forma de los pixels. Así pues, si los procedimientos gráficos se basaran únicamente en pixels, en cada uno de los dos modos dibujarían formas diferentes. Por ejemplo, en un modo dibujarían en círculo mientras en el otro modo el dibujo sería una elipse.

Los procedimientos gráficos aseguran que, sin importar el modo que esté en uso, se producirán las figuras adecuadas. Para indicar los tamaños de las figuras no basta simplemente una cuenta de pixels. En lugar de hacer esto, en los procedimientos gráficos, para especificar los tamaños y posiciones de las figuras, se utilizan escalas y coordenadas elegidas arbitrariamente.

Los procedimientos gráficos utilizan un **sistema gráfico de coordenadas,** es decir unas coordenadas relacionadas con el origen gráfico, que es un punto situado en el ángulo inferior izquierdo de la ventana especificada o de la ventana de omisión. Este origen NO es el mismo que el *origen de los pixels,* utilizado en la definición de la posición de las *ventanas, bloques,* etc. El origen gráfico permite la utilización de un sistema de coordenadas cartesiano estándar. Véase la figura. Después de cada operación gráfica, la posición del cursor gráfico se pone al día, y las operaciones siguientes serán bien relativas a la posición del cursor, o bien absolutas, es decir relativas al origen gráfico.



Sistema Gráfico de Coordenadas

El **factor de escala** es tal que la máxima distancia en la dirección vertical en la ventana especificada o en la ventana elegida por defecto, toma (por defecto) una longitud de cien. Puede, sin embargo, cambiarse con el comando SCALE. La escala en la dirección X es igual a la escala en la dirección Y, sin embargo, la longitud de la línea que puede dibujarse en la dirección X depende de la forma de la ventana. Si se incrementa el factor de escala, se incrementará el tamaño de la figura que se puede dibujar sin exceder el tamaño de la ventana. Si la salida gráfica se conecta a otra ventana de tamaño diferente, el tamaño correspondiente de la salida se ajusta para que quepa en la nueva ventana. Si la figura sobrepasa la ventana asociada a su salida, se cortará adecuadamente.

Es muy útil considerar la ventana como una ventana (valga la redundancia) dentro de un espacio gráfico más amplio en el que se puede dibujar las figuras. El comando **SCALE** permite establecer el origen de los gráficos, de forma que permite que la ventana pueda moverse por el espacio gráfico.

Los procedimientos gráficos tienen salida a la ventana asociada con el **CANAL** especificado o con el **CANAL** de omisión. Esta salida se realizará en la tinta-**INK** de ese canal.

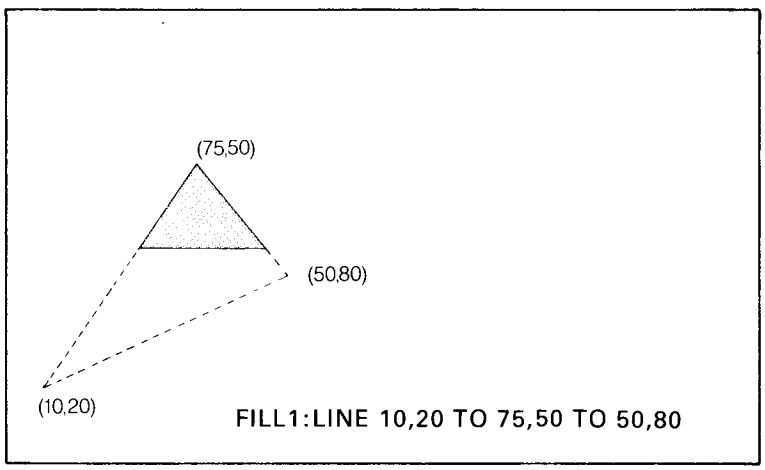
Comando	Función	
CIRCLE	dibuja un círculo o una elipse	absolutas
LINE	dibuja una línea	
ARC	dibuja un arco de círculo	
POINT	traza un punto	
CIRCLE-R	dibuja una elipse o un círculo	relativas
LINE-R	dibuja una línea	
ARC-R	dibuja un arco de círculo	
POINT-R	traza un punto	
SCALE	establece la escala y mueve el origen	
FILL	llena de color una forma	

llenado de color en los gráficos-fill

Las figuras dibujadas con los procedimientos gráficos y gráficos de tortuga pueden rellenarse opcionalmente con un color o una mezcla de colores específica. Si se selecciona el comando **FILL**, cuando se trace la figura, se rellenará de color.

El algoritmo de **FILL** almacena una lista de puntos a trazar en lugar de trazarlos. Una vez que la figura se cierra, existirán dos puntos sobre una misma línea horizontal. Esos dos puntos se unen por una línea del color de tinta vigente en ese momento y el proceso se repite. El comando **FILL** debe seleccionarse de nuevo antes de dibujar otra figura para estar seguros de que la memoria intermedia utilizada para almacenar la lista de puntos se ha puesto a cero.

Veamos el siguiente diagrama, que ilustra el comando **FILL**.



advertencia

Debe tenerse en cuenta una importante restricción de **FILL**. Este comando no debe usarse en figuras que son cóncavas. Estas formas cóncavas deben dividirse en otras formas menores que no lo sean y cada una de ellas se llenará independientemente.

identificadores

Un *identificador* en lenguaje SuperBASIC es una secuencia de letras, números y signos de subrayado.

letras: | A... Z
| a... z

números: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |

identificador:= letra *[|*letra* | *número* | _ |]*

i. a
ii. limite_1
iii. nuevo_límite
iv. cuenta

Los identificadores deben comenzar con una letra e ir seguidos por una secuencia de letras, números y subrayados. Pueden tener una longitud de hasta 255 caracteres. Las letras mayúsculas y minúsculas son equivalentes.

Los identificadores se utilizan en SuperBASIC para identificar a las *variables, procedimientos, funciones, bucles de repetición,* etc.

advertencia

Los identificadores no tienen otro significado que no sea el de identificar construcciones en SuperBASIC. El SuperBASIC no puede deducir el uso que usted intenta hacer de un identificador partiendo de su nombre.

joystick

El joystick con puertas **CTL1** y **CTL2** permite enchufar al QL dos joysticks.

Los joysticks se organizan de forma que al moverse de una forma específica generen las pulsaciones de unas teclas determinadas. Todos los programas que utilicen joysticks deben ser capaces de adaptarse a esto. El teclado del QL puede leerse directamente utilizando el comando **KEYROW**.

	CTL1	CTL2
modo	tecla	tecla
hacia arriba	cursor arriba	F4
hacia abajo	cursor abajo	F2
a la izquierda	cursor izquierda	F1
a la derecha	cursor derecha	F3
disparos	espacios	F5

comentario

Las puertas joysticks pueden utilizarse también para añadir al QL otros dispositivos de control más generales.

palabras clave

En SuperBASIC, las palabras clave son *identificadores,* definidos en la *Guía de Referencia de Palabras Clave* de SuperBASIC. Las palabras clave tienen la misma forma que los *identificadores* estándar del SuperBASIC. El tipo de letras (mayúsculas y minúsculas) de la palabra clave no tiene importancia. Estas palabras clave se repiten mezclando mayúsculas y minúsculas, y se han reproducido siempre completas. En este libro, la parte de la palabra clave que aparece en mayúsculas indica el mínimo que es preciso teclear para que el SuperBASIC reconozca esa determinada palabra clave. Por el contrario, los procedimientos ordinarios deben definirse escribiendo sus nombres en minúsculas.

El conjunto de palabras clave se amplía añadiendo procedimientos al sistema y cargándolos en el sistema de QL. Es útil definir siempre aquellos procedimientos de la forma en que se van a utilizar. Si se introducen sus nombres en mayúsculas, el SuperBASIC siempre los visualizará en mayúsculas y de este modo ayudará indicándonos su función especial dentro del programa y del sistema QL.

advertencia

Las palabras clave ya existentes no pueden utilizarse como identificadores ordinarios en los programas de SuperBASIC.

Palabras clave

ABS	DEFine PROCedure	LEN	RANDOMISE
ACOS, ASIN	END DEFine	LET	RND
ACOT, ATAN	DEG	LIST	RECOL
ADATE	DELETE	LOAD	REMark
ARC, ARC_R	DIM	LOCal	RENUM
AT	DIMN	LN, LOG10	REPeat,
AUTO	DIR	LRUN	END REPeat
BAUD	DIV	MERGE	RESPR
BEEP	DLINE	MOD	RETurn
BEEPING	EDIT	MODE	RETRY
BLOCK	ELLIPSE,	MOVE	RUN
BORDER	ELLIPSE_R	MRUN	SAVE
CALL	EOF	NET	SIN
CHR$	EXEC, EXEC_W	NEW	SCALE
CIRCLE	EXIT	NEXT	SCROLL
CIRCLE_R	EXP	ON GO TO	SDATE
CLEAR	FILL	ON GO SUB	SELect
CLOSE	FILL$	OPEN, OPEN_IN	END SELect
CLS	FLASH	OPEN_NEW	SEXEC
CODE	FOR	OVER	SQRT
CONTINUE	END FOR	PAN	STOP
RETRY	FORMAT	PAPER	STRIP
COPY, COPY_N	GO SUB	PAUSE	TAN
COS	GO TO	PEEK, PEEK_W	TO
COT	IF, THEN, ELSE	PEEK_L	TURN
CSIZE	END IF	PENUP	TURN TO
CURSOR	INK	PENDOWN	UNDER
DATA, READ,	INKEY$	PI	VER$
RESTORE	INPUT	POINT, POINT_R	WIDTH
DATE$, DATE	INSTR	POKE, POKE_W	WINDOW
DAY$	INT	POKE_L	
DEFine FuNction,	KEYROW	PRINT	
END DEFine	LBYTES	RAD	

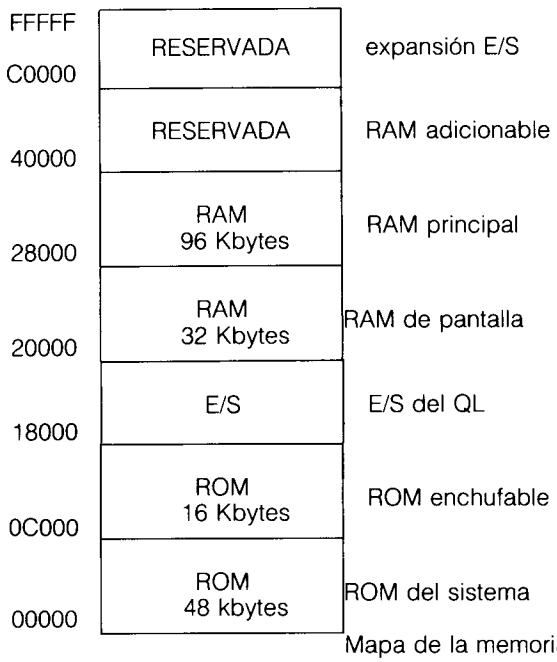
funciones matemáticas

El SuperBASIC dispone de las funciones matemáticas y trigonométricas estándar.

Función	Nombre
COS	coseno
SIN	seno
TAN	tangente
ATAN	arco tangente
ACOT	arco cotangente
ACOS	arco coseno
ASIN	arco seno
COT	cotangente
EXP	exponencial
LN	logaritmo natural
LOG10	logaritmo en base 10
INT	entero
ABS	valor absoluto
RAD	conversión a radianes
DEG	conversión a grados
PI	devuelve el valor de π
RND	genera un número aleatorio
RANDOMISE	regenera el generador de números aleatorios

MAPAS DE MEMORIA

El QL contiene un microprocesador Motorola 68008 que puede direccionar un Megabyte de memoria, es decir, desde 00000 a hex FFFFF Hex. Sinclair Research ha definido las direcciones de este espacio de memoria del siguiente modo:



Mapa de la memoria Física

La memoria RAM de pantalla está organizada del siguiente modo: en series de palabras de dieciséis bits comenzando en la dirección Hex 20000 y avanzando en el orden en que se activa la pantalla, es decir de derecha a izquierda en cada línea y de arriba a abajo en la figura. Los bits de cada palabra se organizan de forma que el pixel situado más a la izquierda es más significativo que el de su derecha. (El patrón de pixels de la pantalla tiene el mismo aspecto que el patrón binario.) Sin embargo, la organización de la información sobre el color en los dos modos de pantalla es diferente:

byte alto AO = 0	byte bajo AO = 1	modo
GGGGGGGG	RRRRRRRR	modo 512 (Alta Resol.)
GFGFGFGF	RBRBRBRB	modo 256 (Baja Resol.)

G-Verde B-Azul R-Rojo F-Flash

Mapa de los Colores

Al establecer el bit-"interruptor" de Flash, se pondrá el estado de Flash, y se congelará el color de fondo con el valor del flash dado por R, G y B para ese pixel. El Flash se reinicializa siempre al comienzo de cada línea visualizada.

En alta resolución, cuando se especifiquen los colores rojo y verde juntos al hardware los interpretará como blanco.

advertencia

La utilización de las áreas reservadas del mapa de la memoria puede causar incompatibilidades con otros productos que Sinclair desarrolle en el futuro. Las salidas espúreas a direcciones consideradas como direcciones de E/S para los periféricos pueden provocar comportamientos impredecibles. Por tanto es recomendable NO escribir o utilizar esas áreas para ningún otro propósito. La utilización de las áreas usadas como bufferes de Microdrive pueden estropear los datos del Microdrive causando pérdidas de información. El almacenamiento de datos en las áreas que se utilizan como tablas del sistema puede hacer que éste falle. El resultado es la pérdida de los datos y los programas.

Todas las Entradas/Salidas deben realizarse introduciendo los comandos adecuados de SuperBASIC y las facilidades del *sistema operativo Qdos*.

Microdrives

Los Microdrives constituyen el principal sistema de almacenamiento permanente en el QL. Cada cartucho de Microdrive tiene una capacidad de al menos 100 Kbytes. Cuando es necesario, para aumentar la potencia del Sistema, el Qdos puede asignar algún espacio libre de memoria como bufferes de Microdrive.

Cada cartucho de Microdrive debe **formatearse** antes de su uso, y puede tener hasta 255 sectores de 512 bytes por sector. El Qdos mantiene un directorio de los archivos almacenados en el cartucho. Cada archivo de Microdrive se identifica utilizando un nombre de *archivo* o Dispositivo estándar en SuperBASIC.

Los cartuchos pueden protegerse contra la escritura arrancando la pequeña pestaña del lado derecho del cartucho.

Cuando reciba un cartucho de Microdrive nuevo, y por tanto en blanco, formatéelo varias veces para poner la cinta en condiciones.

cuidados generales

Físicamente, cada cartucho de Microdrive contiene una cinta de 200 pulgadas de cinta de vídeo de alta calidad que se mueve a una velocidad de 28 pulgadas por segundo. La cinta completa totalmente el circuito cada siete segundos y medio.

NUNCA toque la cinta con los dedos ni inserte nada en el cartucho.

NUNCA desconecte o conecte el ordenador con los cartuchos introducidos en su lugar

SIEMPRE almacene los cartuchos en sus cubiertas, cuando no se están utilizando.

SIEMPRE inserte o saque los cartuchos del Microdrive suavemente y despacio.

SIEMPRE asegúrese de que el cartucho está introducido adecuadamente antes de arrancar el Microdrive.

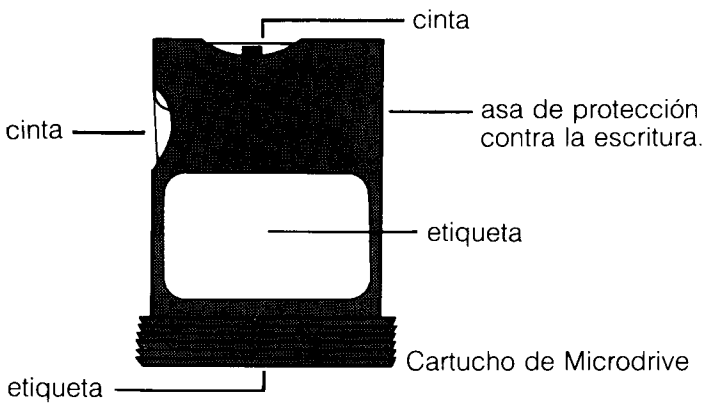
NUNCA mueva de lugar el QL con los cartuchos instalados, incluso si no está funcionando

NUNCA toque el cartucho cuando el drive está funcionando.

NO inserte y saque repetidamente el cartucho sin hacer funcionar al Microdrive.

cintas (bucles)

Si la cinta aparece en cualquiera de las dos posiciones que muestra el dibujo de la figura 1, libérela introduciéndola en el cartucho con cuidado. Utilice un instrumento no fibroso, como puede ser el lado de una pluma o un lápiz. Nunca toque la cinta con los dedos por la razón que sea.



Comando	Función
FORMAT	prepara un cartucho nuevo para su uso
DELETE	borra un archivo de un cartucho
DIR	lista los archivos de un cartucho
SAVE SBYTE SEXEC	almacena los datos desde un cartucho
LOAD LBYTES EXEC MERGE	carga los datos desde un cartucho
OPEN_IN OPEN_NEW OPEN CLOSE	abre y cierra archivos
PRINT INPUT INKEY$	E/S en SuperBASIC

advertencia

Si intenta escribir en un cartucho protegido contra la escritura su QL intentará cada vez escribir los datos. Elimine el cartucho para parar el QL. También puede ocurrir que le envíe un mensaje de error: "Medio erróneo".

monitor

A su ordenador QL puede unirle un monitor en color. La unión se realiza en la parte posterior de su ordenador, y a través de un conector RGB. La conexión se realiza a través de un conector adecuado en el otro extremo o a través de un cable de tres vías DIN si el monitor es monocromo. Las conexiones se indican en la tabla que aparece más abajo, en la que los colores de la última columna se refieren al color de los diferentes cables que componen el conector. En cuanto a las patillas del conector, su uso puede verse también en el diagrama que se muestra más abajo.

Patilla	Función	Señal		Color del cable
1	PAL	Pal compuesta	(4)	naranja
2	GND	tierra		verde
3	VIDEO	compuesto monocromo de vídeo	(3)	marrón
4	CSYNC	sincr. compuesta	(2)	azul
5	VSYNC	sincr. vertical	(1)	amarillo
6	VERDE	verde	(1)	rojo
7	ROJO	rojo	(1)	blanco
8	AZUL	azul	(1)	púrpura

Los monitores monocromos pueden conectarse utilizando para ello un cable de antena de 3 vías o un conector de 8 vías DIN que se conectará al QL. Sólo es necesrio conectar las patillas 2 (tierra) y (3) vídeo a través del cable. El extremo conectado al monitor variará de acuerdo con cada monitor, pero normalmente el interruptor será tipo phono. El monitor deberá disponer de una entrada no invertida de 74 ohmios 1V pk-pk de vídeo compuesto (que es la estándar en la industria). Estos dos conectores, tanto el de 3 vías DIN como el de phono se pueden comprar con facilidad en las tiendas de sonido.

Los monitores RGB (color) se pueden conectar utilizando un cable de 8 vías DIN que se unirá al QL en uno de sus extremos. La conexión con el monitor variará con cada monitor (en la industria no hay conectores estándar) y, por tanto, normalmente este conector se incluye con el monitor. El cable con su conector de 8 vías DIN puede obtenerse de Sinclair Research Limited.

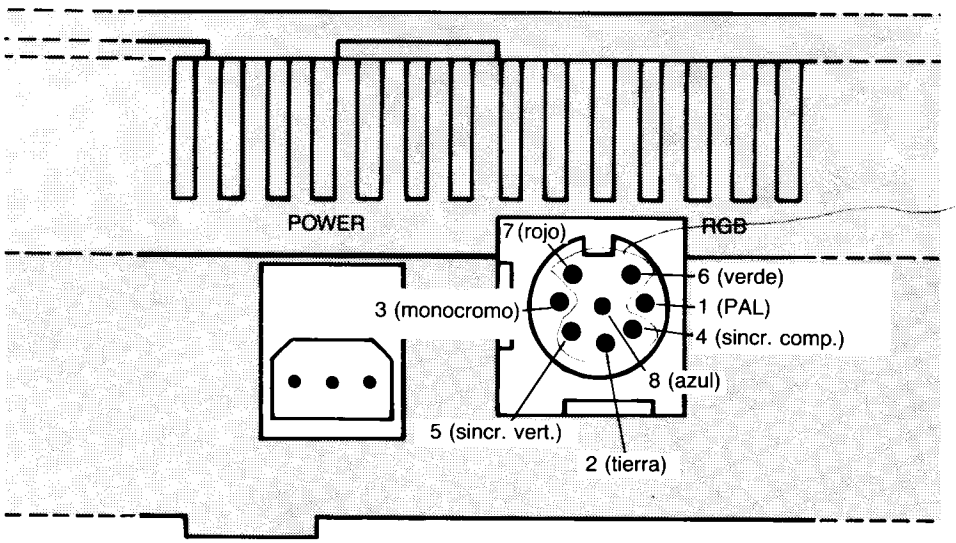


Diagrama del conector para Monitor. Se encuentra en la parte posterior de su QL. En él pueden verse las diferentes patillas con sus funciones.

RED LOCAL

El QL puede conectarse hasta con 63 y/o QLs o Spectrums. Si la red local está formada por dos o más ordenadores, cada ordenador (o estación) debe tener asignado un único número de estación. En el QL puede hacerse utilizando el comando **NET.**

La información se transmite por la red local en bloques. Para una comunicación normal entre dos estaciones, la estación receptora debe dar cuenta de la correcta recepción del bloque. Si algún bloque no recibe correctamente, la estación receptora pedirá una nueva transmisión.

La estación de la red local 0 es una estación especial. La salida a la estación 0 **(neto_0)** se llama difusión. Cualquier estación que esté a la escucha de la estación 0 **(neti_0)** podrá recibir datos. Observe que la forma normal de verificar que un mensaje ha sido recibido es inadecuada para la difusión, de forma que los mensajes difundidos no deben tener una longitud mayor que un bloque (255 bytes).

En cualquier momento, atendiendo a su propia estación de la red podrá acceder a la comunicación con toda la red. (Ejemplo **NET3:LOAD neti_3.**)

Comando	Función
NET	asigna un número a una estación de la red
OPEN CLOSE	abre un canal de la red cierra un canal de la red
PRINT INPUT INKEY$	E/S de la red
LOAD SAVE LBYTES SBYTES EXEC SEXEC LRUN MRUN MERGE	carga y almacena a través de la red

comentario

Si piensa conectar varios QL en una red local, o utilizar un cable muy largo, debe utilizar un cable especial coaxial de baja capacitancia de aproximadamente 3 amp. Realice las conexiones de la parte central y la exterior del cable con cuidado. Aunque el software puede soportar 63 estaciones, el hardware no podrá manejar más de 100 m de cable (dependiendo del tipo empleado).

Si sólo desea conectar unas cuantas máquinas, no tendrá problemas, no se preocupe.

operadores

Operador	Tipo	Función
=	flotante cadena	lógica comparación tipo 2
= =	numérico cadena	aprox. igual comparación tipo 3 **
+	numérico	adición
−	numérico	sustracción
/	numérico	división
*	numérico	multiplicación
<	numérico cadena	menor que (comparación tipo 2)
>	numérico cadena	mayor que (comparación tipo 2)
< =	numérico cadena	menor o igual (comparación tipo 2)
> =	numérico cadena	mayor o igual que (comparación tipo 2)
< >	numérico cadena	diferente de (comparación tipo 3)
&	cadena	concatenación
&&	bitwise	AND
||	bitwise	OR
^ ^	bitwise	XOR
~ ~	bitwise	NOT
OR	lógico	OR
AND	lógico	AND
XOR	lógico	XOR
NOT	lógico	NOT
MOD	entero	módulo
DIV	entero	divide
INSTR	cadena	comparac. cadena tipo 1
^	flotante	elevación a potencias
(^)	entera	elev. a potencias
−	flotante	minus unario
+	flotante	más unario

**aprox. igual = igual en 1/10

Si la operación lógica especificada es verdadera, el sistema devolverá un valor diferente de cero, si la operación es falsa el valor devuelto será cero.

prioridades

El orden de prioridades de los operadores de SuperBASIC está definido en la tabla anterior. Si el orden de valoración no puede deducirse de esta tabla, las operaciones se realizarán de izquierda a derecha. La prioridad establecida en el SuperBASIC puede eliminarse encerrando entre paréntesis las diferentes partes de la expresión.

mayor menos y más unarios
concatenación de cadenas
búsqueda de cadenas
exponenciación
multiplicación y división (módulos y enteros)
suma y sustracción
comparaciones lógicas
NOT
AND
menor OR y XOR

expansiones periféricas

El conector de expansión permite conectar al QL periféricos adicionales. Las conexiones disponibles en el conector son las siguientes:

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	IPLOL
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

El conector del QL es de 64 vías (macho) DIN-41612 de borde indirecto.

La L que se añade al nombre una señal indica que la señal es activa por debajo.

Señal	Función
A0.A19	Líneas de direccionamiento del 68008
RDWL	Lectura/Escritura
ASL	Señal Strobe de direcciones
DSL	Señal Strobe de los datos
BGL	Bus Grant
DSMCL	Señal Strobe de los datos-Chip maestro
CLCKCPU	Reloj de la CPU
E	Reloj de Periféricos 6800
RED	Rojo
BLUE	Azul
GREEN	Verde
CSYNCL	Sincronismo compuesto
VSYNCH	Sincronismo vertical
ROMOEH	Liberación de salida a ROM
FCO	Estado del Procesador
FC1	Estado del Procesador
FC2	Estado del Procesador
RESETCPUL	Reinicialización de la CPU

Señales de salida a Perifericos de QL

Señal	Función
DTACKL	Disponibilidad (Bus arbitration)
BRL	Llamada (Bus request)
VPAL	Dirección válida de Periférico
IPLOL	Interrupción prioridad 0
IPL1L	Interrupción prioridad 1
BERRL	Error en el Bus
EXTINTL	Interrupción externa
DBGL	Grabación en el Bus de datos

Señales de entrada a Periféricos de QL

Señal	Función
D0... D7	Líneas de datos

Señales Bi-Direccionales a Periféricos de QL

Señal	Función
SP0.SP3	Elige periférico de 0 a 3
VIN	9V DC (nominal) 500 mA máximo
VM12	−12V
VP12	+12V
GND	tierra

Otras señales

No consideramos que la descripción que sigue de los mecanismos periféricos de expansión del QL sea suficiente para implementar un dispositivo de expansión que usted pueda necesitar, pero sí creemos que debe leerse para tener un conocimiento básico del mecanismo de expansión.

Al QL se le pueden añadir dispositivos periféricos (uno o varios hasta un máximo de 16 dispositivos). Si el periférico es único, puede conectarse directamente en la Ranura de Expansión del QL, pero si los dispositivos son varios, deben conectarse en el Módulo de expansión del QL que a su vez se conecta en la Ranura de expansión del QL a través de una tarjeta buffer.

En este contexto, el término dispositivo incluye también memoria de expansión. Aunque las zonas del mapa de memoria del QL asignadas a la memoria de expansión son diferentes que las de los dispositivos de expansión, el mecanismo es básicamente el mismo. En un momento dado, sólo se puede conectar al QL una única expansión de memoria periférica, el espacio de direcciones asignado a la expansión periférica en el mapa de memoria física del QL permite 16 Kbytes por periférico. Este área debe contener la zona de memoria E/S necesaria para el manejador y el código del manejador mismo.

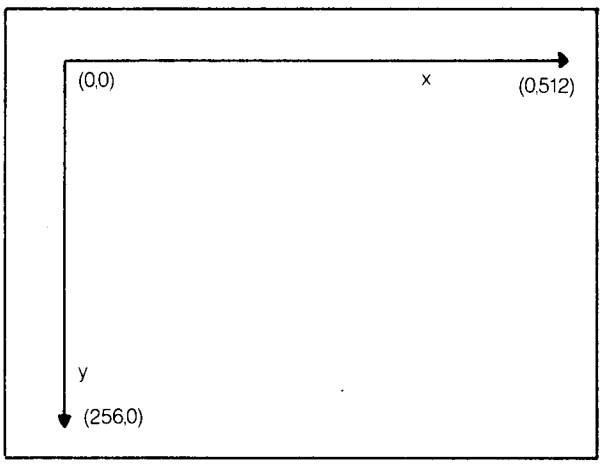
El Qdos incluye facilidades para el manejo en "cola" y E/S serie únicas que pueden ser útiles cuando se escriben manejadores de dispositivos.

La posición de cada periférico dentro de la memoria total del QL viene determinada por las líneas de señales de selección: SP0, SP1, SP2 y SP3. Las líneas de selección generan una señal que corresponde a la posición de la ranura en el módulo de expansión del QL, y, así para un dispositivo que se va a seleccionar, las entradas de dirección que se obtienen desde las líneas de direcciones: A14, A15, A16, A17 deben ser las mismas que las señales que provienen de las tres líneas de selección respectivamente.

pixels sistema de coordenadas

El **sistema de coordenadas de pixels** se utiliza para definir las posiciones y tamaños de las *ventanas, bloques* y las posición del cursor en la pantalla del QL. Este sistema de coordenadas tiene su origen en el ángulo superior izquierdo de la ventana de omisión (pantalla), y siempre supone que las posiciones se especifican con la pantalla en el modo 512 (modo de alta resolución). El sistema utiliza el pixel más próximo disponible para ese modo en particular, haciendo las coordenadas independientes del modo de pantalla en uso en ese momento.

Algunos comandos siempre se relacionan con el origen de la ventana de omisión. Un ejemplo puede ser el comando **WINDOW**. Otros, siempre están relacionados con el origen de la ventana que se está utilizando en ese momento, como por ejemplo el comando **BLOCK**.



Sistema de Coordenadas de Pixels

programa

Un programa de SuperBASIC consiste en una secuencia de instrucciones de SuperBASIC en la que cada instrucción va precedida por un *número de línea*. Estos números de línea están en una gama que va desde 1 a 32768.

Comando	Función
RUN	inicia un programa ya cargado
LRUN	carga un programa desde un dispositivo y luego lo inicia
CTRL ESPACIO	fuerza la parada del programa

sintaxis: **número de línea= *[dígito] * [gama 1.32768]*
[número-línea instrucción * [:instrucción]*]

ejemplo: i.
```
10 PRINT "Este número de línea es válido"
RUN
```
ii.
```
 10 REM Pequeño programa
 20 FOR tono = 0 TO 7
 30   FORcontraste = 0 TO 7
 40     FOR mezcla = 0 TO 3
 50       PAPER tono, contraste, mezcla
 60       CURSOR 0,70
 70       FOR n = 0 TO 2
 80         SCROLL 2,1
 90         SCROLL 2,1
100       END FOR n
110     END FOR mezcla
120 END FOR contraste
130 END FOR tono
RUN
```

Qdos

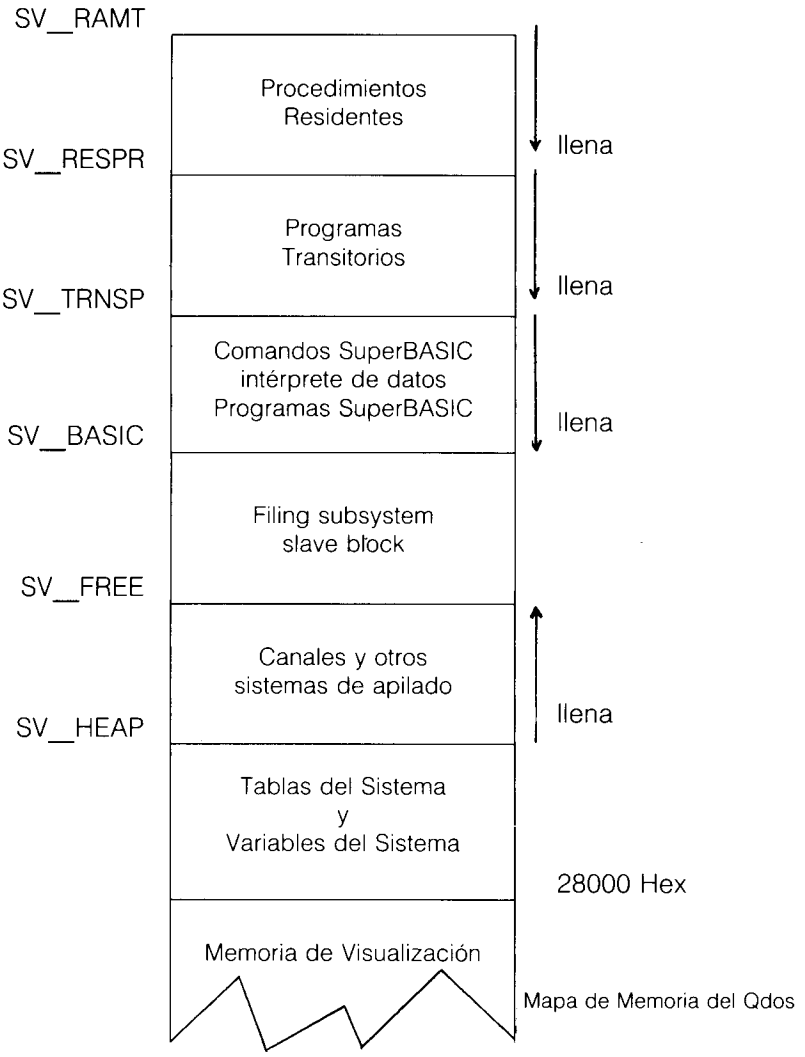
El **Qdos** es el Sistema Operativo del QL. Controla:

Organización de Tareas y asignación de recursos
E/S de Pantalla (incluidas las ventanas)
E/S Microdrive
Red Local y canales de comunicación serie
Entradas por teclado
Gobierno de la memoria

La descripción completa del Qdos se sale del propósito de esta guía pero incluimos una breve descripción.

mapa de memoria

El sistema RAM tiene una organización que viene impuesta por el sistema operativo Qdos. A continuación describimos su definición:



Los términos SV_RAMT, SV_RESPR, SV_TRNSP, SV-BASIC, SV_FREE, SV_HEAP se utilizan para representar ciertas direcciones interiores al QL. Estos términos no son reconocidos por el sistema operativo SuperBASIC o Qdos. Más aún, las direcciones representadas pueden cambiar con el funcionamiento del sistema.

sv_ramt **RAM TOP**-Dirección más alta de la memoria
Varía según las tarjetas de la expansión de memoria conectada al sistema.

sv_respr **Procedimientos residentes**
Los procedimientos residentes se cargan en la parte superior de la memoria RAM. Se puede obtener espacio en el área de

procedimientos residentes utilizando la función RESPR, pero el espacio utilizado sólo puede modificarse volviendo a inicializar el sistema. Los procedimientos residentes, escritos en código máquina se añaden a la lista de nombres de SuperBASIC, y de este modo se convierten en extensiones al sistema SuperBASIC.

sv_trnsp

Programas Transeúntes
Los programas transeúntes se cargan inmediatamente debajo de los procedimientos residentes. Cada programa debe contenerse en sí mismo; es decir, debe contener espacio para sus propios datos y su propia pila. Debe colocarse independiente o debe cargarse mediante un cargador especialmente diseñado como cargador de unión (linking loader). Los programas transeúntes se ejecutan desde el basic utilizando el comando **EXEC** o desde el Qdos activándolo como una tarea.

El área de un programa transeúnte puede utilizarse solamente para almacenar los datos, pero éstos se tratarán por el Qdos como tarea, y por tanto no se deben activar.

sv_basic

Area de SuperBASIC
Esta área contiene todos los programas de SuperBASIC cargados y los datos relacionados con ellos. El área se expande y se contrae utilizando el espacio libre que se necesario.

sv_free

Espacio libre
El espacio libre se utiliza por el subsistema de archivos del Qdos para crear Bloques Slave de Microdrive, es decir, copias de bloques de Microdrive que pueden guardarse en la RAM.

sv_heap

Pila del sistema
Se utiliza para que el sistema almacene datos, definiciones de canal, etc. También suministra almacenamiento de trabajo para subsistema E/S. Los programas transeúntes pueden obtener espacio de trabajo para ellos en la pila a través de las llamadas del sistema Qdos.

Tablas del Sistema/Variables del Sistema
Esta área está situada directamente encima de la memoria de pantalla. Las Tablas del Sistema y la pila del supervisor se colocan encima de las variables del sistema.

llamadas al sistema

Las llamadas al Sistema se procesan por el Qdos en **modo supervisor.** Cuando el Qdos se encuentra en modo supervisor, no permite que el procesador realice otra tarea diferente. Las llamadas al Sistema que se procesan de este modo se llaman **atómicas,** es decir, la llamada al sistema procesará todo antes de volver a conectar el procesador. Algunas llamadas al Sistema son **atómicas parciales,** es decir, una vez han completado su función primaria conectarán el procesador *si es necesario.* Si no se pide específicamente, todas las llamadas al sistema de E/S son atómicas parcialmente.

El mecanismo estándar para hacer una llamada al sistema es realizar un trap a uno de los vectores del sistema Qdos con los parámetros adecuados en los registros del procesador. La acción que toma el Qdos después de una llamada al sistema depende de cada llamada particular y del estado general del sistema en el momento en el que se realizó la llamada.

entrada/salida

El Qdos soporta un entorno multiárea y, por tanto, se puede acceder a un archivo por más de un proceso al mismo tiempo. El subsistema de llenado del Qdos puede manejar archivos que se han abierto como archivos **exclusivos** o

como archivos **compartidos.** No se puede escribir sobre un archivo compartido. Los dispositivos QL se procesan por el **subsistema E/S serie.** El subsistema de llenado y el subsistema E/S serie juntos forman el **subsistema E/S redireccionables.** Como bien indica su nombre, cualquier dato sacado mediante ese sistema puede redireccionarse a cualquier otro dispositivo que esté soportado también por el sistema de E/S redireccionables.

Los nombres de dispositivo que necesita el Qdos son los mismos que los nombres de dispositivos que necesita el SuperBASIC y se discuten en la sección *dispositivos,* del Capítulo *conceptos.* El conjunto de dispositivos estándar que se incluyen en el QL puede ampliarse.

dispositivos

Los dispositivos estándar incluidos en el sistema se discuten en esta guía en la sección **dispositivos.** Para añadir nuevos dispositivos al sistema debe dárseles un nombre (por ejemplo SER1, NET) y a continuación se podrá acceder a ellos del mismo modo que a cualquier otro dispositivo QL.

multiárea

Las diferentes tareas se pueden compartir en la CPU en línea, con una cierta prioridad y competición con otras tareas del sistema. Las tareas que funcionan bajo el control del Qdos pueden estar en uno de los tres estados siguientes:

activa: Capaz de trabajar y compartir algunos recursos del sistema. Una tarea en este estado no puede ejecutarse con continuidad, pero compartirá la CPU de acuerdo con su prioridad.

suspendida: La tarea se puede ejecutar, pero está en espera de otra tarea o E/S. Una determinada tarea puede suspenderse indefinidamente por un período de tiempo especificado.

inactiva: La tarea no puede ejecutarse, su prioridad es 0 y por tanto nunca podrá compartir la CPU.

El Qdos reorganizará los tiempos del sistema automáticamente a una velocidad que es función de la velocidad de partición (frame rate). El Sistema se reorganizará también después de un cierto número de llamadas al sistema.

El Qdos reorganizará el sistema automáticamente a una velocidad relacionada con la frecuencia de 50 Hz. El sistema también se reorganiza de nuevo después de ciertas llamadas al sistema.

ejemplo: Este programa genera una presentación en pantalla del reloj de tiempo real y funciona como tarea independiente.

En primer lugar, ejecute este programa con un cartucho formateado en el Microdrive 2. Se generará un código de máquina llamado "reloj". Espere que el Microdrive se pare. A continuación ponga en marcha el reloj usando la instrucción **SDATE.**

En ese momento, teclee:

```
EXEC mdv2_reloj
```

y aparecerá en el ángulo superior derecho de la ventana de comandos una indicación permanente de la hora.

```
100  c=RESPR(100)
110  FOR i=0 TO 68 STEP 2
120    READ x :PODE_W i+c,x
130  END FOR i
140  SEXEC mdv2_reloj,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
```

Nota. La línea 1060 controla la posición y el color de la ventana del reloj; los datos que aparecen son los siguientes:

> color y anchura del borde, color de fondo **(PAPER)** y de texto **(INK)**, anchura de la ventana, altura, origen-x, origen-y.

Se introducen parejas de bytes (como palabras) mediante el **POKE_W.**

El origen-x y origen-y, los últimos datos introducidos, serán 272 y 202 en modo Monitor, o 240 y 216 en Modo Televisor.

Introduzca la palabra con los datos del color del fondo y de texto como 256 * fondo + texto; por ejemplo, fondo blanco y texto rojo serán 256 * 7 + 2 = 1794.

Repetición

En el lenguaje SuperBASIC, la repetición se controla con dos estructuras básicas de programación. Cada estructura debe llevar un identificador:

REPEAT *identificador*
instrucciones
END REPEAT *identificador*

FOR *identificador = rango*
instrucciones
END FOR *identificador*

Estas dos estructuras se usan en conjunción con otras dos instrucciones de SuperBASIC:

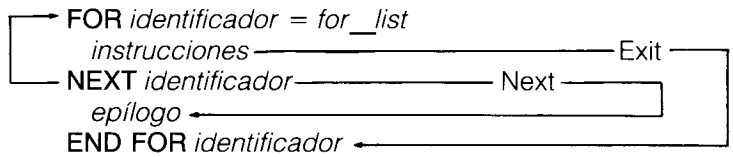
NEXT *identificador*

EXIT *identificador*

En el proceso de instrucción **NEXT**, el control pasará bien a la instrucción que sigue a la instrucción apropiada **FOR** o **REPeat**, o, si se agotó la gama de **FOR**, el control pasará a la instrucción siguiente a una instrucción **NEXT**.

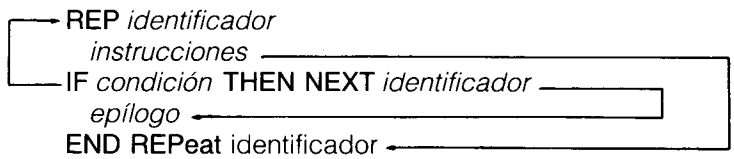
Cuando se procesa una instrucción **EXIT**, el control pasará a la instrucción de después de **END FOR** o **END REPeat** seleccionada por la instrucción **EXIT** . Si se utiliza una instrucción **EXIT** en un bucle, éste debe terminar con una instrucción **END FOR** o **END REPeat**. **EXIT** puede utilizarse para salir de varios niveles de estructuras repeat anidadas. Esta instrucción **EXIT** debe utilizarse en bucles **REPeat** para terminar el bucle con alguna condición.

Podemos añadir un **epílogo del bucle** a los bucles **FOR** y **REPeat** mediante una combinación de instrucciones **NEXT EXIT** y **END**. Un epílogo de bucle consiste en una serie de instrucciones que se ejecutan con alguna condición especial que se indica en el interior del bucle.



Los epílogos de los bucles sólo se procesan si el bucle **FOR** termina normalmente. Si el bucle termina a través de una instrucción **EXIT**, el proceso continuará en **END FOR** y el epílogo no se procesará.

En los bucles **REPEAT** se pueden tener estructuras similares:



En esta ocasión, la entrada al bucle está controlada por una instrucción **IF**. El epílogo se procesará o no dependiendo de la condición de la instrucción *IF*. También puede utilizarse una instrucción **SELect** para controlar la entrada al epílogo.

ROM
ROM-ranura (slot) para cartuchos

Permite cargar el software en el sistema del QL a través del cartucho ROM del QL Sinclair. El cartucho ROM puede contener software para cambiar directamente el comportamiento del sistema SuperBASIC. El cartucho puede contener:

i. Software para sustituir el sistema SuperBASIc. Por ejemplo:

   assemblers
   compiladores
   debuggers (depuradores)
   aplicaciones de software
   etc.

ii. Software para la expansión del sistema SuperBASIC. Por ejemplo:

   procedimientos especiales
   etc

En el QL no pueden utilizarse Cartuchos ZX ROM.

conector de salida (pin out)

—	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

Señal	Función
AO..A15	Líneas de direcciones
DO..D8	Líneas de datos
ROMOE	Disponibilidad de Salida ROM
VDD	5V
GND	Tierra

advertencia

Nunca conecte o desconecte un cartucho ROM mientras el QL esté funcionando.

Pantalla

modo 512

La pantalla tiene 512 pixels en la dirección horizontal y 256 pixels en la dirección vertical. En este modo sólo se pueden visualizar los colores simples:

negro
rojo
verde

El modo de baja resolución también tiene un flash de hardware.

modo 256

La pantalla tiene 256 pixels en la dirección horizontal y 256 pixels en la dirección vertical. En este modo se puede disponer de toda la gama de colores simples:

negro
azul
rojo
morado
verde
ciano
amarillo
blanco

advertencia

Los aparatos de televisión domésticos no puede visualizar la pantalla del QL completa. Ciertas zonas de la pantalla, en su parte superior y también la zona izquierda, no se reproducirán. La ventana inicial, de omisión, tiene en cuenta este problema y reducirá el tamaño efectivo del dibujo. Para volver a tener el tamaño normal debe utilizarse el comando **WINDOW.**

Comando	Función
MODE	establece el modo de pantalla

cortes-slicing

En ciertas circunstancias podemos referenciar más de un elemento en una matriz, es decir cortar, delimitar esa matriz. Esta parte de la matriz puede definirse en SuperBASIC como una **submatriz** o una serie de submatrices. Cada trozo puede definir una secuencia continua de elementos que pertenecen a una dimensión específica de la matriz original. El término matriz, en este contexto puede incluir muchas cosas, como matrices numéricas, matrices de cadena, o simplemente cadenas.

No es necesario especificar un índice para el número total de dimensiones de una determinada matriz. Si se omite una dimensión, se añaden fragmentos adecuados (slices) para seleccionar la gama completa de elementos de esa dimensión específica, es decir el fragmento (0 TO). El lenguaje SuperBASIC sólo puede añadir submatrices (slices) al final de la lista de los índices de la matriz.

sintaxis:

índice:= *expre_numér* {elemento único}
expre_numer **TO** *expre_numer* {gama de elementos}
expre_numer **TO** {gama hasta el final}
TO *expre_numer* {gama desde el comienzo}

refer_matriz:= *variable*
variable ([*índice**[,*índice*]*)

Los cortes (slices) de matrices se pueden utilizar para especificar una submatriz fuente o destino, en una instrucción de asignación.

ejemplo:

```
i.    PRINT datos_matriz
ii.   PRINT letra$(1 TO 15)
iii.  PRINT matriz_dos (3) (2 TO 4)
```

Los cortes en las cadenas se realizan del mismo modo que los cortes en matrices numéricas o de cadena.

Por tanto:

a$(n)	selecciona el carácter enésimo
a$(n TO m)	selecciona los caracteres desde el enésimo al emésimo
a$(n TO)	selecciona desde el carácter enésimo hasta el final
a$(1 TO m)	selecciona desde el comienzo hasta el carácter emésimo incluido
a$	selecciona el contenido completo de a$

Algunas formas del lenguaje **BASIC** incluyen funciones llamadas **LEFT$, MID$** o **RIGHT$.** En SuperBASIC estas funciones no son necesarias. Sus equivalentes se incluyen más abajo:

SuperBASIC	Otros BASIC
a$(n)	MID$(a$,n,1)
a$(n TO m)	MID$(a$,n,m+ 1− n)
a$(1 TO n)	LEFT$(a$,n)
a$(nTO)	RIGHT$(a$,LEN(a$)+1−n)

advertencia

Si asigna datos a un trozo de matriz o de cadena o a una variable de cadena puede no obtener los resultados que usted espera. Las asignaciones que se realizan de este modo no actualizan la longitud de la cadena, y, por tanto, puede ocurrir que el sistema no procese correctamente esta asignación. La longitud de una matriz de cadena o de una variable de cadena sólo se actualiza cuando la asignación se realiza en la cadena completa.

inicialización

Inmediatamente después de conectar (o reinicializar) el QL probará la memoria RAM, y por tanto se visualizará un dibujo extraño en la pantalla. Una vez pasada esta prueba por la memoria RAM la pantalla se limpiará y aparecerá la pantalla en el copyright

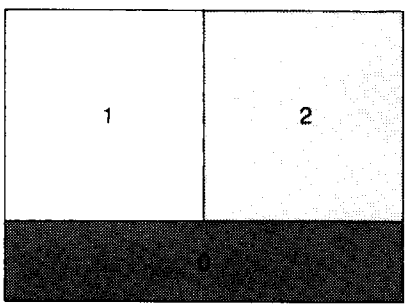
El QL se inicializa después de su conexión, visualiza el mensaje de copyright y determina si se va a utilizar con una televisión o con un monitor. De la respuesta depende que se establezcan modos iniciales de pantalla diferentes, y ventanas cuyos tamaños también dependen de esta respuesta.

Pulse **F1** si va a utilizar un monitor y **F2** si va a utilizar un aparato de televisión doméstico.

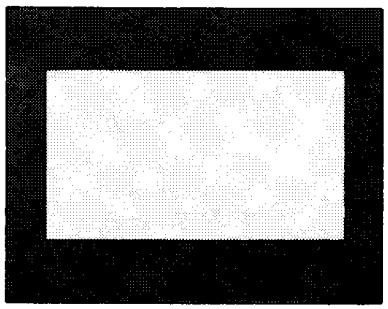
El QL tiene la capacidad de "cargarse él mismo" a partir de programas contenidos en un cartucho ROM, o bien en el Microdrive 1. Si la ranura del cartucho ROM contiene un programa autorrecargable, la inicialización continúa bajo el control del programa que se encuentra en el cartucho ROM, si no encuentra nada adecuado, el QL comprobará un archivo de nombre **BOOT**-cargador-, lo carga y lo ejecuta.

pantalla de omisión

El QL tiene tres canales de omisión, unidos a tres ventanas de omisión



Monitor


Televisión

El canal 0 se utiliza para listar los comandos y enviar los mensajes de error. El canal 1 para salida de los programas y gráficos, y el canal 2 para los listados de los programas. El canal de omisión puede modificarse utilizando el especificador de canal opcional en el comando adecuado.

advertencia

Es importante NO conectar el QL con algún cartucho de Microdrive alojado en su lugar normal. Si es necesario reinicializar el Microdrive, el cartucho se debe insertar en un momento entre la conexión del aparato y la selección entre las dos opciones, **F1** y **F2**.

sonido

El sonido del QL está generado por el sistema IPC (8049) segundo procesador, y se controla con las siguientes especificaciones:

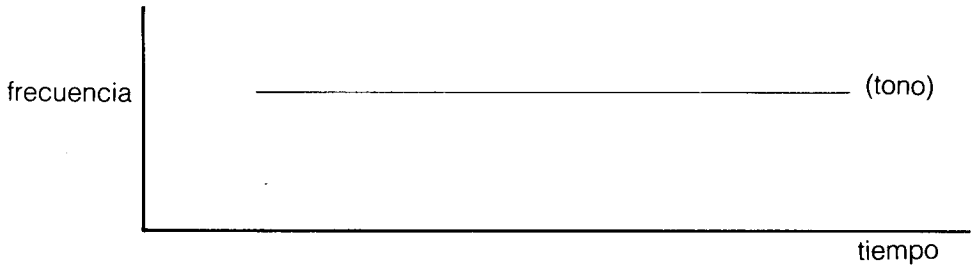
hasta un total de dos frecuencias
velocidad a la que el sonido puede variar entre esas dos frecuencias, la inclinación
comportamiento del sonido después de que haya alcanzado una de las frecuencias especificadas, la envolvente
si debe incluirse alguna componente aleatoria en el sonido, ejemplo desviaciones de la inclinación
si debe incluirse algún "emborronamiento" (fuzzines), es decir alguna desviación en cada ciclo de sonido.

Los resultados de dicho "emborronamiento" suelen ser zumbidos, mientras que los aleatorios que dependen de los otros parámetros dan como resultado "sonidos melódicos" o ruidos.

La complejidad del sonido puede irse realizando etapa por etapa, gradualmente, para ir consiguiendo cada vez sonidos más complejos. De hecho, este es el mejor sistema para controlar y manejar el sonido en el QL.

PRIMER NIVEL

Especifique una duración y una frecuencia única. Esta frecuencia se emitirá durante el tiempo especificado.

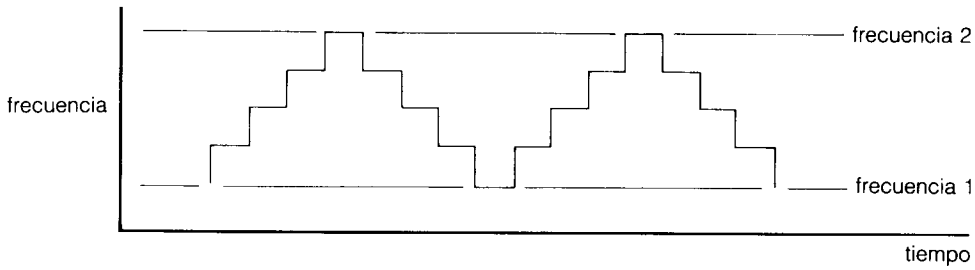


El anterior, es el comando más sencillo para el sonido, en el QL (sin tener en cuenta el comando para parar dicho sonido).

SEGUNDO NIVEL

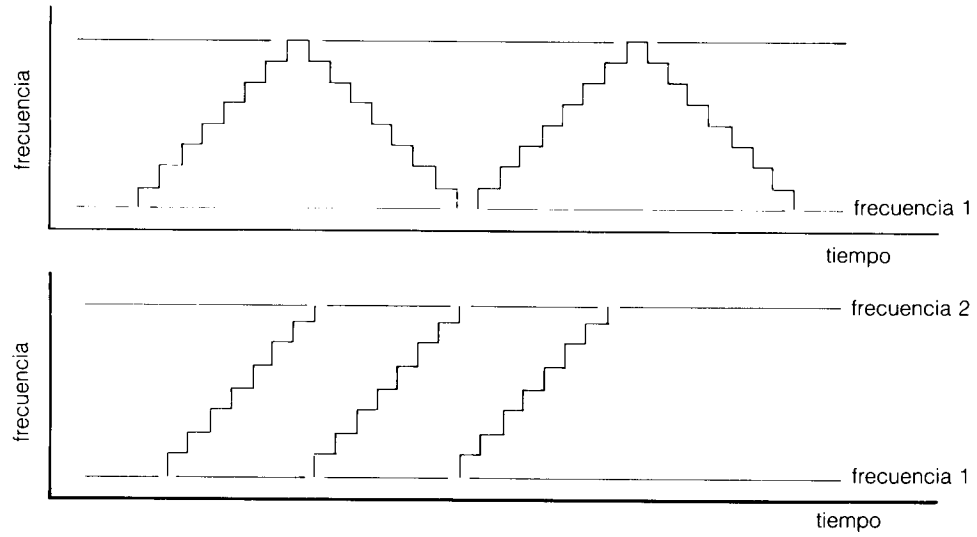
Puede añadirse al comando una segunda frecuencia (tono) y un gradiente. El sonido variará entre estos dos tonos (frecuencias) a la velocidad especificada por el gradiente.

Los sonidos que se obtienen a este nivel pueden variar entre pequeños bocinazos semi-musicales, gruñidos, lamentos, etc. Es mejor experimentar.



TERCER NIVEL

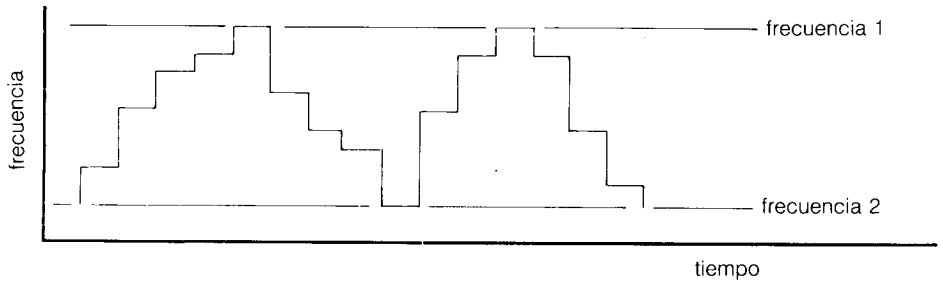
Podemos añadir un tercer parámetro que controle el comportamiento del sonido cuando alcanza a una de las frecuencias especificadas. Podemos hacer que el sonido "oscile" o "envuelva". Puede especificarse el número de "envolventes", incluyendo una envolvente continua. Es más importante aún que antes experimentar.



CUARTO NIVEL

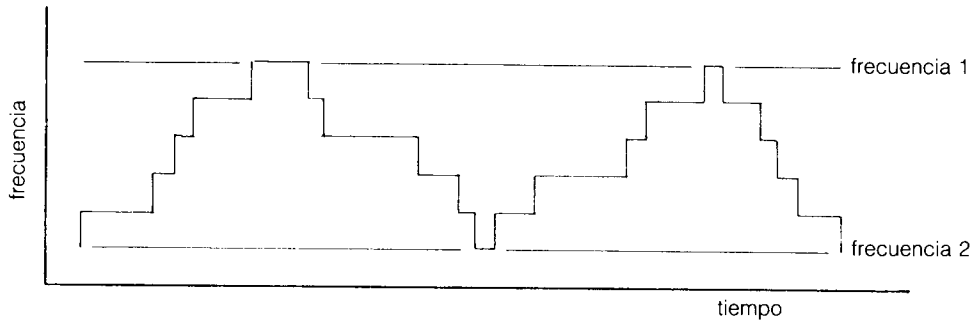
Puede añadirse al sonido alguna componente aleatoria. Esto es, una desviación del escalón o gradiente especificado.

Podemos generar una gama muy amplia así como inesperada de sonidos que depende de la proporción de componente aleatoria añadida en relación con las frecuencias (tonos) y el gradiente.



QUINTO NIVEL

Puede añadirse más variación especificando además un "emborronamiento" (fuzziness). Con ello añadimos un factor aleatorio a la frecuencia cada vez que el sonido oscila o se envuelve. Este "emborronamiento" tiende a hacer que el sonido parezca una sirena.



Combinando todos los efectos anteriormente explicados podemos obtener una gama de sonidos realmente muy amplia. Muchos de ellos serán realmente inesperados. La mejor manera de conocer el sonido de su QL es experimentar con él. Si especifica un intervalo de 0, el sonido puede repetirse para siempre, y se podrán utilizar una secuencia de comandos **BEEP** hasta que el sonido generado sea el que se desea. Debemos, sin embargo, advertir que con un pequeño cambio en un único parámetro los resultados del sonido generado pueden ser alarmantes, por tanto, ¡evite el pánico!

instrucciones

Una instrucción de SuperBASIC es una orden para su QL para que éste realice una operación específica. Por ejemplo:

```
LET a = 2
```

asignará el valor 2 a la variable a.

Puede escribirse en una línea, una o más sentencias, separadas por dos puntos:

```
LET a = a + 2 : Print a
```

Añadirá 2 al valor inicial de la variable a, y lo almacenará en su variable. El resultado se imprimirá.

Si una línea no va precedida por un *número de línea,* esta línea es un *comando directo,* y el SuperBASIC la procesará inmediatamente. Si la instrucción va precedida por un número de línea, esta instrucción se convierte en una parte de un *programa* de SuperBASIC, y se añade al área de programa del SuperBASIC para su posterior ejecución.

Ciertas instrucciones de SuperBASIC pueden tener efecto sobre otras instrucciones que estén en el resto de la línea lógica en la que se encuentra. Por ejemplo **IF, FOR, REPeat, REM,** etc. Por tanto, ciertas instrucciones de SuperBASIC no tienen sentido como comandos directos.

matrices de cadena variables de cadena

Las matrices de cadena y las matrices numéricas son esencialmente iguales. Sin embargo, existen unas pocas diferencias en su tratamiento por el SuperBASIC. La última dimensión de una matriz es la que define la longitud máxima de las cadenas interiores a la matriz. Las variables de cadena pueden tener una longitud cualquiera. Ambos elementos, matrices de cadena y variables de cadena, se pueden "cortar".

Los dos trozos de cadena a ambos lados de la asignación de cadena no tienen por qué ser iguales. Si estos dos lados no son iguales, se puede truncar el lado derecho para que concuerde con la longitud del izquierdo o bien se reduce el izquierdo para conseguir la concordancia. Si se realiza una asignación a una cadena cortada es necesario llenar el "hueco" creado por el corte con espacios.

No es necesario especificar la dimensión final de una matriz de cadena. Al no especificar la dimensión seleccionaremos la cadena completa. Si especificamos un único elemento estaremos tomando un único carácter, y si especificamos un trozo (slice) habremos definido una subcadena.

comentario

A diferencia de muchos BASICs, el SuperBASIC no trata las matrices de cadena como cadenas de longitud fija. Si los datos almacenados en una matriz de cadena son inferiores en número al tamaño máximo de la matriz de cadena, la longitud de la cadena se reducirá.

advertencia

Puede no obtener el efecto deseado al introducir los datos en un trozo de matriz de cadena o variable de cadena. Si se introducen de este modo los datos, no se actualiza la longitud de la cadena, y puede ocurrir que el sistema no reconozca esta asignación de datos. Esta longitud de una matriz de cadena o de una variable de cadena sólo se actualiza cuando se realiza la introducción a la cadena completa.

Comando	Función
FILL$	genera una cadena
LEN$	obtiene la longitud de una cadena

comparación de cadenas

order (punto decimal/punto)
dígitos o números en orden numérico
AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz
espacio ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [|] _ \ { | } ~ ©

Otros caracteres no imprimibles.

La relación existente entre dos cadenas puede ser:

igual: Todos los caracteres o números son equivalentes o iguales.

inferior: La primera parte de la cadena que es diferente del carácter correspondiente en la segunda cadena, es anterior a éste en el orden anteriormente establecido.

mayor: La primera parte de la primera cadena que es diferente del carácter correspondiente de la segunda cadena, es posterior a éste, según el orden antes establecido.

Observe que "." puede tratarse como punto decimal en las comparaciones de clasificación de números (como las que realiza el SuperBASIC). Debe tenerse en cuenta también que las comparaciones de cadenas que contengan caracteres no imprimibles pueden dar resultados falsos.

tipos de comparaciones

tipo 0 dependiente de mayúsculas-comparación carácter por carácter

tipo 1 independiente de mayúsculas-comparación carácter por carácter

tipo 2 independiente de mayúsculas-los números se clasifican en orden numérico

tipo 3 dependiente de mayúsculas-los números se clasifican en orden numérico

uso

El tipo 0 no se utiliza normalmente por el Sistema del SuperBASIC

El tipo 1 se utiliza para comparaciones de Archivos y variables

El tipo 2 se utiliza en: <, <=, =, 0, >=, > y <> del SuperBASIC

El tipo 3 se utiliza por el SuperBASIC para las equivalencias: ==

sintaxis de las definiciones

La sintaxis del SuperBASIC se define utilizando un metalenguaje no riguroso tipo notación. Se pueden utilizar cuatro tipos de construcción:

| | Seleccione uno de
[ ] Los elementos entre paréntesis son opciones
* * Los elementos interiores a ellos se repiten

. . Gama
{ } Comentario

ejemplo:

| A | B |	es A o B
[A]	A es opcional
* A*	Se repite la A
A.. Z	A, B, C etc.
{comentario}	

Consideremos un identificador de SuperBASIC.

Una secuencia de números, dígitos, guiones, comenzando en una letra y terminando con un signo opcional % o $.

letra:= **A... Z**
a... z

{una letra es uno de los siguientes símbolos: ABCDEFGHIJKLMNOPQRSTUVWXYZ}
o: abcdefghijklmnopqrstuvwxyz

dígitos:= | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |

{0 o 1 o 2 o 3 o 4 o 5 o 6 o 7 o 8 o 9}

guión:= _

{un guión es _}

identificador = *letra* * [*letra* | *dígito* | *guión*] * | % | $ |

debe comenzar por una letra

secuencia de letras, dígitos o guiones
repite algo que es opcional

gráficos de "tortuga"

El SuperBASIC dispone de un conjunto de comandos para obtener gráficos de "tortuga".

Comando	Función
PENUP	Para el dibujo
PENDOWN	Comienza a dibujar
MOVE	Mueve la "tortuga"
TURN	Gira la "tortuga"
TURNTO	Gira a un punto de comienzo especificado

Este conjunto de comandos es mínimo y normalmente se usa dentro de otro procedimiento para expansión de los comandos. Por ejemplo:

```
10 DEFine PROCedure delante (distancia)
20   MOVE distancia
30 END DEFine
40 DEFine PROCedure atrás (distancia)
50   MOVE - distancia
60 END DEFine
70 DEFine PROCedure Izdo. (ángulo)
80   TURN ángulo
90 END DEFine
100 DEFine PROCedura escribe (ángulo)
110   TURN - ángulo
120 END DEFine
```

Hemos definido así algunos de los más conocidos comandos de gráficos de tortuga.

Inicialmente, la pluma de la "tortuga" se encuentra hacia arriba y apuntando a 0 grados, que es hacia el lado derecho de la ventana.

El comando **FILL** también funciona para el relleno de color en figuras trazadas con estos gráficos de tortuga. Los gráficos normales y los de "tortuga" pueden mezclarse, aunque la dirección de la "tortuga" no se ve afectada por los comandos gráficos ordinarios.

ventanas

Las ventanas son zonas de la pantalla que se comportan en muchos aspectos como si cada ventana individual fuera una pantalla por sí misma, es decir, esta ventana sufrirá un desplazamiento cuando se encuentre llena de texto (scroll), y se vaciará (limpiará) con el comando **CLS**, etc.

Las ventanas pueden especificarse y unirse a un canal, una vez que éste se haya *abierto*. La ventana vigente en ese momento puede sufrir cambios en su forma mediante el comando **WINDOW**, y también se le puede añadir un borde mediante el comando **BORDER**. Se puede dirigir una salida a una ventana imprimiento (*print*) el canal adecuado. Se puede obtener una entrada de una determinada ventana introduciéndola (INPUT) por el *canal* adecuado. Si existen varios canales listos para la introducción de entradas, la entrada que se va a utilizar se puede ir seleccionando entre todos los canales que están preparados pulsando

CTRL C

Y el cursor estará intermitente en la ventana seleccionada.

Las ventanas pueden utilizarse para salidas de gráficos, y no de gráficos al mismo tiempo. La salida no gráfica está relacionada con la posición del cursor que puede colocarse en cualquier lugar de la ventana utilizando para ello el comando **CURSOR**, y en cualquier línea y columna mediante el comando **AT**. La salida gráfica está relacionada con el cursor gráfico, que puede colocarse y manipularse con los procedimietos *gráficos*.

partes

Ciertos comandos (**CLS**, **PAN**; etc.) aceptan parámetros opcionales para definir una parte de la ventana vigente en ese momento como ventana de operación. Vamos a definir a continuación dicho parámetro.

Parte	Descripción
0	La pantalla completa
1	Por encima del cursor excluyendo la línea en que éste se encuentra
2	La parte inferior de la pantalla excluyendo la línea del cursor
3	La línea del cursor completa
4	La línea a la derecha del cursor incluyéndolo

Comando	Función
WINDOW	redefine una ventana
BORDER	añade un borde a una ventana
PAPER	define el color de fondo (paper) de una determinada ventana
INK	define el color del texto de una determinada ventana
STRIP	define el patrón de mezcla de colores de una determinada ventana
PAN	desliza el contenido de las ventanas (horizontales)
SCROLL	desplaza el contenido de la ventana (vertical)
AT	establece la posición de impresión
CLS	limpia (vacía) la ventana
CSIZE	establece el tamaño de los caracteres
FLASH	flash en los caracteres
RECOL	vuelve a colorear una ventana