

CONTROL DEL CRT

Pero, ¿qué es el CRT? Pues, sencillamente, las iniciales de «Cathode Ray Tube», que traducido a nuestro idioma quiere decir, literalmente, «Tubo de Rayos Catódicos», o en términos menos técnicos, la pantalla «física» del monitor de su Amstrad, que también es un tubo de rayos catódicos como el de un televisor cualquiera.



De lo anterior se deduce que el artículo que presentamos está dedicado a cómo controlar la pantalla. Y efectivamente, así es, pero desde un punto de vista distinto al habitual del BASIC y, sin embargo, sin utilizar código máquina ni PEEK's ni POKE's, a menos que, por razones de velocidad de ejecución, se decida emplear otros medios más rápidos que sólo son posibles desde código máquina, y para los que la base teórica que expondremos seguirá siendo igualmente válida.

Con las órdenes existentes en BASIC podemos hacer un montón de cosas «EN la pantalla», pero también hay un comando que nos permitirá hacer otras tantas «CON la pantalla». Y dicho comando no es, ni más ni menos, que el comando «OUT».

EN LA PANTALLA Y CON LA PANTALLA

El comando OUT generalmente pasa desapercibido para el programador normal, sobre todo porque en los manuales no se explica cuál puede ser su utilidad práctica, limitándose a dar en dos líneas una escueta descripción de la acción que realiza y nada más. Transcribimos para aquéllos que no tengan el manual a mano:

«OUT < número de portal > , < expresión entera >

Envía el valor del parámetro entero al portal

de SALIDA correspondiente a la dirección mencionada.»

Aun cuando tan sucinta definición es totalmente exacta, lo menos que puede ocurrir es que nos deje absolutamente indiferentes si no se complementa con un ejemplo o con una explicación de lo que es un portal de salida.

Intentaremos dar un poco de luz sobre el tema a lo largo del artículo, pero antes, y para desperezar los dedos, teclearemos en modo directo las siguientes instrucciones, operando en **MODE 1** y procurando no equivocarse:

```
OUT &BC00, 1 [ENTER] OUT &BD00, 20 [ENTER]
```

¿Qué es lo que ha ocurrido? Aparentemente, se han insertado líneas en blanco entre las que acaba de escribir, pero intente escribir algo, de más de 20 caracteres, o moverse con el cursor más allá de la columna 20 y verá cómo el cursor salta a la primera columna de la línea siguiente. ¡EUREKA! Pensará que hemos descubierto el comando WINDOW, sólo que más complicado. PUES NO. Puede comprobarlo tecleando MODE 1, con lo que volveríamos a la pantalla normal, pero que en esta ocasión no sirve de nada. Para ver lo que ha sucedido teclee:

```
BORDER 15 [ENTER]
```

Esta orden pone el borde de la pantalla de color naranja y, en estas circunstancias, además, la mitad derecha de la pantalla también aparecerá de este color, lo que quiere decir que la pantalla útil se ha reducido a la mitad. Para ser más exactos, los 40 caracteres de una línea normal ahora se distribuyen en dos líneas de 20 caracteres, es decir, la pantalla de $25 \times 40 = 1.000$ caracteres ahora se ha transformado en 50 líneas de 20 caracteres. Imagínese la pantalla como una masa de letras que comprimimos horizontalmente: las letras al no poder salirse de la pantalla, se desplazarán hacia abajo, reestructurando la visualización, pero sin perderse ni una, solamente que no las vemos. Compruébelo desplazando el cursor por debajo de la última línea visible, pulse ESCAPE y siga bajando el cursor hasta que empiece el scroll hacia arriba, con



lo que aparecerá el «*Break*» que no veíamos antes.

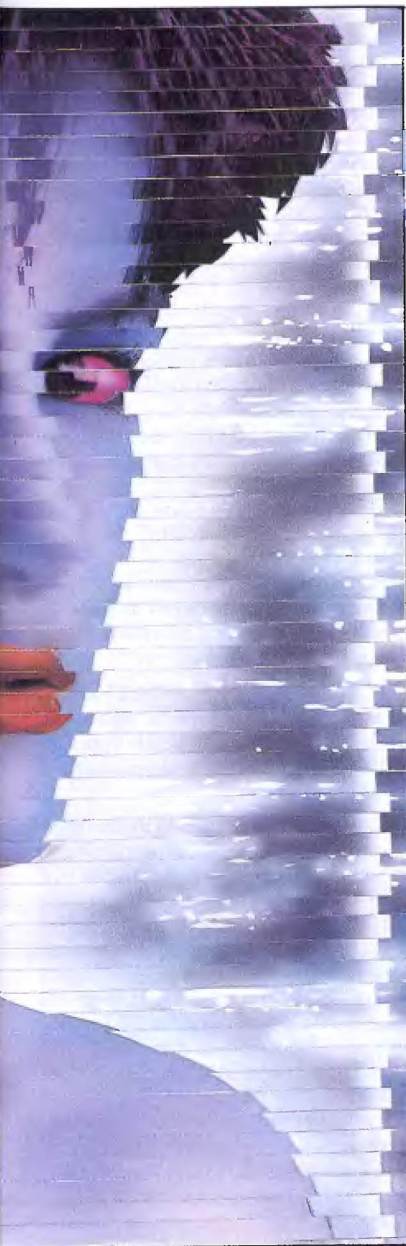
Para volver a la pantalla normal escriba:

```
OUT &BC00,1: OUT &BD00,40 [ENTER]
```

o resetee el ordenador.

Ahora se comprenderá la diferencia entre hacer cosas «EN la pantalla» y «CON la pantalla», puesto que la orden OUT afectará a toda la pantalla y no a zonas de la misma.

Antes de que se le ocurra empezar a experimentar por su cuenta haciendo OUT's a diestro y siniestro, estamos en la obligación moral de advertir que un OUT de cualquier valor en cualquier portal de salida, sin un ligero



conocimiento de lo que se hace, puede producir un «hermoso» bloqueo del sistema, siendo necesario desconectar el ordenador, y si no prueba con:

OUT &BC00,0: OUT &BD00,1

UN POCO DE TEORIA

Todos los ordenadores están compuestos físicamente por una serie de elementos que nos permiten trabajar con él. El elemento más importante es el microprocesador o Unidad de

Control de Proceso (CPU), que en los **Amstrads** es el renombrado Z80A, y que no es más que una *pastillita* de silicio de unos cuantos milímetros cuadrados conteniendo cerca de 8.000 transistores, interconectados de tal manera que posibilitan hacer cosas tales como jugar a marcianitos o llevar la contabilidad de una empresa.

Pero decimos que «posibilitan» porque, a pesar de la potencia del microprocesador, éste por sí solo no es incapaz de hacer absolutamente nada si no se encuentra conectado con otros dispositivos como un teclado, una pantalla o la memoria; mediante los cuales los podemos dar órdenes o recibir los resultados de una serie de operaciones o almacenar datos, respectivamente.

El Z80A es el cerebro que controla, directa o indirectamente, todas las acciones que realiza el ordenador, y entre ellas se encuentran: leer el teclado, manejar la pantalla, generar sonidos, controlar el cassette, enviar datos a la impresora, leer y escribir en la memoria, etc.

Nos centraremos en el área del control de la pantalla. El Z80 maneja la pantalla indirectamente a través de un chip o circuito integrado llamado CRTC, cuyas siglas corresponden a «**Cathode Ray Tube Controller**» (Controlador del Tubo de Rayos Catódicos).

El CRTC realiza las funciones de mantenimiento de la visualización en pantalla, controlando las dimensiones y la posición de la zona de exposición de información en la superficie del Tubo, y generando las direcciones de memoria donde están los datos de pantalla, entre otras cosas.

Todo esto se consigue utilizando el contenido de los 18 registros de que dispone el CRTC. Podemos entender un registro como un dispositivo cuya misión es retener una información para posteriormente ser tratada por el CRTC. Cada uno de estos registros puede contener un número de un byte de longitud, esto es, con un valor entre cero y 255.

Modificando el contenido de los registros del CRTC modificaremos las características de la pantalla, según hemos podido comprobar. ¿Y cómo modificar un registro? Nada más sencillo, con la orden **OUT**.

Como ya hemos dicho, el CRTC es un circuito conectado al microprocesador, y éste se comunica con aquél a través de unos canales que se encuentran en los llamados **PORTALES** de SALIDA. Un **PORTAL** no es más que una dirección de memoria donde se depositan datos para ser enviados al dispositivo correspondiente a ese número de portal. Explicándolo de otra forma, hacer un **OUT** de un Número a un Portal es como enviar una carta con un mensaje a una persona. El **OUT** es la acción de enviar la carta; el Número sería el mensaje, y el Portal sería la dirección de esa persona.

Necesitamos, pues, conocer la dirección del portal de salida correspondiente al Controlador del Tubo de Rayos Catódicos para poder enviarle los datos que queremos.

PROGRAMACION

El CRTC tiene dos portales o direcciones, cada uno de los cuales tiene una misión específica. El primer portal se utiliza para indicarle al CRTC el número del registro que queremos seleccionar. Y el segundo portal es por donde le enviaremos el valor al registro previamente seleccionado. Las direcciones son: **&BC00**: para seleccionar número de registro. **&BD00**: para modificar contenido del registro.

Aun cuando hemos dicho que el CRTC tiene 18 registros, el sistema sólo permite acceder a los primeros 16, por lo que el número que tendremos que enviar al primer portal del CRTC, el **&BC00**, será un número entero entre cero y 15, inclusive, por aquello de que los ordenadores empiezan a contar desde cero. Lo haremos del siguiente modo:

OUT &BC00, < num. reg > siendo num. reg > un valor entre 0 y 15.

para modificar el contenido del registro seleccionado:

OUT &BD00, < dato > siendo < dato > un valor entre 0 y 255.

En la tabla 1 damos los nombres de cada uno de los registros, así como los valores estándar y los valores mínimos y máximos en cuya gama no se bloquee el ordenador. Insistimos en que cualquier valor fuera de esta gama nos obligará a tener que desconectar nuestro **Amstrad**.

También existen portales de **ENTRADA** al microprocesador, donde recoger datos mediante la orden del **BASIC IN**, pero no los trataremos por salirse del tema que abordamos. Únicamente decir al respecto que un uso inapropiado de las entradas puede ocasionar **daños físicos** a los circuitos del **ordenador**.

Comentar todos y cada uno de los efectos que se producen al modificar los registros del CRTC con los distintos valores posibles es prácticamente imposible. Por otra parte, enviando **OUT**'s en modo «directo» y de forma desordenada nos encontraremos con que es una forma un tanto pesada de investigar, y otras veces habremos cambiado tantos registros que ya no sabremos cómo volver a la pantalla normal.

Por todo ello, creemos que la mejor solución está en facilitar las cosas con un programa que nos permita manejar los registros a nuestro antojo, pudiendo recuperar la pantalla normal cuando queramos y ver qué registros hemos modificado hasta ese momento, para, de esta manera, sacar nuestras propias conclusiones de cómo inciden los diferentes parámetros enviados al CRTC, sobre todo, pensando en el dicho de que una imagen vale más que mil palabras. Todo esto nos lo permitirá el programa «**CONTROL DEL CRT**» que luego comentaremos.

TABLA 1. REGISTROS DEL CRTC

N. Reg	Estand.	Min.	Max.	Nombre del Registro
0	63	46	100	Total
1	40	0	64	Horizontal Visualizado
2	46	0	63	Posición de sincronismo horizontal
3	142	(*)	(*)	Ancho del sincronismo
4	38	0	255	Total Vertical
5	0	0	255	Ajuste del Total Vertical
6	25	0	50	Vertical Visualizado
7	30	0	255	Posición de sincronismo vertical
8	30	0	255	Modo solapado y sesgado
9	7	0	255	Máximo número de líneas de barrido
10	0	0	255	Comienzo del cursor sintetizado
11	0	0	255	Fin del cursor sintetizado
12	48	0	255	Dirección comienzo (byte alto) rastreo memoria de pantalla
13	0	0	255	Idem (byte bajo)
14	192	0	255	Registro del cursor (byte alto)
15	0	0	255	Idem (byte bajo)

(*)=Dependiente del contexto de los demás registros.

Volviendo sobre el famoso dicho, pruebe a teclear el siguiente programa que nos muestra una forma más vistosa de presentar las pantallas en nuestros programas:

```
10 BORDER 15: MODE 0
20 FOR a=40 TO 0 STEP-1: GOSUB 60:
NEXT: '--- Recoge Pantalla
30 GOSUB 70: '---Pinta Pantalla
40 FOR a=0 TO 40: GOSUB 60: NEXT: '---
Extiende Pantalla
50 CALL &BB06: GOTO 20: '---Espera pulsación y vuelve a empezar
60 CALL &BD19: OUT &BC00,1: OUT
&BD00,a: RETURN: 'Ejecuta OUT's
70 t=TIME
80 CLG RND*13
90 WHILE TIME < t+300
100 DRAW 640*RND, 400*RND, 13*RND:
TAG: PRINT CHR$(RND*26+64):
110 WEND
120 RETURN
```

A los que posean el juego **BEACH HEAD** les resultará familiar esta forma de recoger la pantalla. Está basada en variar el contenido del registro 1 del CRTC entre los valores 40 y 0, mediante un bucle FOR... NEXT STEP -1, para recoger la pantalla, y al revés, entre 0 y 40, para extenderla de nuevo.

Antes de cambiar el registro con los OUT's correspondientes, hacemos una llamada al FIRMWARE, en &BD19, para sincronizar el cambio que se va a realizar con el haz de barrido de pantalla, consiguiéndose una mayor suavidad en el movimiento. Es lo mismo que hacer un FRAME en el CPC 6128. Para ver la necesidad del CALL &BD19, pruebe a quitarlo.

Otro de los ejemplos que se me ocurren como consecuencia de indagar con los registros del CRTC utilizando el programa «CONTROL CRT» es el siguiente:

```
10 FOR d=1 TO 3: READ men$
20 men$=SPACE$( (40-LEN (men$))/2)+
+men$: '---Centra mensaje
30 MODE 1
40 LOCATE 1,20
50 FOR a=1 TO 40
60 letra$=MID$(men$,a,1)
70 GOSUB 140: '---Ejecuta OUT's
80 PRINT letra$:
90 NEXT
100 FOR r=0 TO 1500: NEXT: '---Retardo
110 LOCATE 1,25: PRINT STRING$(20,10):
'---Sube mensaje
120 NEXT
130 RUN
140 CALL &BD19: OUT &BC00,13: OUT
&BD00,a: RETURN
150 DATA MICROHOBBY AMSTRAD ESPECIAL
160 DATA Presenta:
170 DATA *CONTROL DEL CRT*
```

Esta vez se modifica el registro 13, que contiene el byte bajo de la dirección de comienzo de rastreo de la memoria de pantalla, por lo que, al ir desplazando dicha dirección con el comando OUT, se desplaza a su vez la pantalla, produciendo un suave **SCROLL** horizontal de la pantalla.

Esta última técnica, en combinación con unos cuantos conocimientos de código máquina para obtener la rapidez necesaria, nos permitiría construir un programa que utilizara la pantalla de su **Amstrad** como un tablón de anuncios electrónico a todo color, como ya he podido ver en algunos sitios.

Antes de hacerlo correr es conveniente salvarlo en cinta/disco por si hubiera algún OUT perdido que nos lo echara todo a perder.

Primero tenemos una pantalla de presentación en la que se explica la función y el manejo del programa, para, después de ver una pequeña demostración del uso del CRTC, aparecer la pantalla de trabajo con un informe del estado actual de los registros, el valor estándar entre corchetes y el nombre de los mismos.

Abajo, a la derecha, y recuadrado, tenemos los dos Modos de funcionamiento, apareciendo en vídeo inverso el modo activo.

En modo «Ejecutar» podemos modificar los registros sin que se observe su efecto, hasta pulsar las teclas CTRL+E.

En modo «Inmediato», nada más modificar un registro, se ejecuta la acción inmediatamente.

Se puede cambiar de modo en cualquier momento pulsando las teclas SHIFT.

En cuanto seleccionamos un registro, su nombre aparece en vídeo inverso y el programa espera la pulsación de una de las teclas posibles que se visualizan en la última línea y que son:

«V»: Visualiza el rango permitido y espera la entrada del dato a enviar al registro.

«Cursor Arriba/Cursor Abajo»: Incrementa/Decrementa el contenido del registro en una unidad.

Cualquier otra tecla, así como un valor fuera de rango, anula la acción, produciendo una señal acústica de error.

En la pantalla se informa con un asterisco del último registro modificado, y, en vídeo inverso aparecerán los valores que no sean estándar, al objeto de una más fácil localización.

Si se llega a perder el control de la pantalla, pulsando la tecla «N» volvemos a la pantalla normal, apareciendo el mensaje: PULSE UNA TECLA PARA VOLVER AL ESTADO ANTERIOR.

Si en ese momento estamos en modo Inmediato y pulsamos cualquier tecla, excepto SHIFT, volveremos al caso anterior; pero si pulsamos SHIFT, cambiamos al modo Ejecutar, por lo que, al pulsar otra tecla, se mantiene la pantalla normal, pudiendo restaurar los registros al valor que queremos hasta que pulsemos las teclas CTRL+E para observar el efecto que se produce.

Con la tecla «X» veremos cuatro de las infinitas posibilidades que nos presenta el manejo del CRTC.

Por fin, pulsando «T» termina el programa restaurando los valores normales del CRTC.

Finalizar aclarando que el registro r3, Ancho del Sincronismo, es muy inestable, por lo que el programa no permite acceder al mismo. No obstante, el que quiera aventurarse sólo tiene que modificar la línea 710 quitando la condición «OR tec = 4». ¡Ah!, y que alterar los registros que se refieren al Cursor no produce efecto alguno.


```

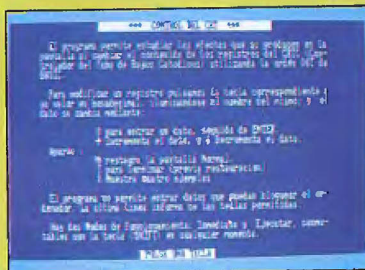
10 '*-----*
20 '* CONTROL CRT *
30 '* por J.F.Bayo *
40 '* Mayo 1986 *
50 '*-----*
60 :
70 '>:: PRESENTACION ::
80 :
90 MODE 2:WINDOW 8,80,1,25
100 i%=CHR$(24)
110 PRINT,i% SPC(7) "*** CONTROL
DEL CRT ***" SPC(7)i%:PRINT
120 PRINT" El programa permite est
udiar los efectos que se producen e
n la"
130 PRINT"pantalla al cambiar el co
ntenido de los registros del CRTC (
Con-"
140 PRINT"trolador del Tubo de Rayo
s Catodicos) utilizando la orden U
T de"
150 PRINT"BASIC.":PRINT
160 PRINT" Para modificar un regis
tro pulsamos la tecla correspondien
te a"
170 PRINT"su valor en hexadecimal,
iluminandose el nombre del mismo; y
el"
180 PRINT"dato se cambia mediante:"
:PRINT
190 PRINT,"V para entrar un dato, s
eguido de ENTER."
200 PRINT, CHR$(240)+ " Incrementa e
l dato, y "+CHR$(241)+ " Decrementa
el dato."
210 PRINT" Aparte : "
220 PRINT,"N restaura la pantalla N
ormal."
230 PRINT,"T para Terminar (previa
restauracion)."
240 PRINT,"X Muestra cuatro ejemplo
s":PRINT
250 PRINT" El programa no permite
entrar datos que puedan bloquear el
or-"
260 PRINT"denador. La ultima linea
informa de las teclas permitidas.":
PRINT
270 PRINT" Hay dos Modos de funcio
namiento: Inmediato y Ejecutar, co
nmu-"
280 PRINT"tables con la tecla <SHIFT>
en cualquier momento.":PRINT
290 PRINT TAB(23) i%+ " PULSE UNA TE
CLA "+i%
300 :
310 '>:: INICIALIZACION ::

```

```

320 :
330 DEFINT a-u
340 DIM stat(3,15),nom$(15)
350 FOR a=0 TO 15
360 READ stat(0,a),stat(1,a),stat(2
,a),nom$(a)
370 stat(3,a)=stat(2,a)
380 f%=MID$(STR$(stat(3,a)),2)
390 f%=SPACE$(3-LEN(f%))+f%
400 nom$(a)="["+f%+" ]"+nom$(a)
410 NEXT
420 ENT -1,1,10,1:ENT -2,1,-10,1
430 ENV 1,7,1,1,7,-1,1:ENV 2,1,10,1
,10,1,1,15,5,1,15,-1,15
440 masc1%=i%+"&"+i$:masc2%=i%+" ##
# "+i%

```



```

450 m$(0)="Ejecutar " :m$(1)="Inmedi
ato"
460 CALL &B806: BORDER 12
470 REM MIN,MAX,STANDAR,NOMBRE DEL
REGISTRO
480 DATA 46,100,63,Total horizontal
490 DATA 0,64,40,Horizontal Visuali
zado
500 DATA 0,63,46,Posicion sincronis
mo horizontal
510 DATA 0,255,142,Ancho del sincro
nismo <No Accesible>
520 DATA 0,255,38,Total Vertical
530 DATA 0,255,0,Ajuste total Verti
cal
540 DATA 0,50,25,Vertical visualiza
do
550 DATA 0,255,30,Posicion sincroni
simo vertical
560 DATA 0,255,0,Modo solapado o se
sgado
570 DATA 0,255,7,Numero lineas de b
arrido
580 DATA 0,255,0,Comienzo del curso
r
590 DATA 0,255,0,Fin del cursor
600 DATA 0,255,48,Direccion comienz

```

```

o rastreo (byte alto)
610 DATA 0,255,0,Idem (byte bajo)
620 DATA 0,255,192,Registro del cur
sor (byte alto)
630 DATA 0,255,0,Idem (byte bajo)
640 :
650 GOSUB 1480:*** DEMOSTRACION **
*
660 :
670 '>:: CONTROL ENTRADAS ::
680 :
690 GOSUB 840:'inkey$
700 tec=INSTR("0123456789ABCDEFNTX"
+CHR$(5),q%)
710 IF tec=0 OR tec=4 THEN SOUND 7,
600,-2,0,1,2:GOTO 690
720 IF tec=17 THEN GOSUB 1200:GOTO
690
730 IF tec=18 THEN 1200
740 IF tec=19 THEN GOSUB 1340:GOSUB
1300:GOTO 690
750 IF tec=20 THEN GOSUB 1270:GOTO
690
760 reg=tec-1
770 LOCATE 20,reg+3:PRINT USING mac
c1%:nom$(reg)
780 LOCATE 20,23:PRINT"<V> <Flechas
cursor> Cualquier otra anula la ac
cion"
790 GOSUB 840
800 IF q%="v" OR q%="V" THEN LOCATE
1,23:PRINT CHR$(10)+CHR$(11):PRINT
"ENTRAR DATO ("stat(0,reg)..."stat
(1,reg)") (ENTER para volver):";LI
NE INPUT "",v$:v=VAL(v%):IF STR$(v)
=" "+v% THEN IF v<=stat(1,reg) AND
v>=stat(0,reg) THEN stat(3,reg)=v:G
OTO 890
810 IF q%=CHR$(240) AND stat(3,reg)
<stat(1,reg) THEN stat(3,reg)=stat(
3,reg)+1:GOTO 890
820 IF q%=CHR$(241) AND stat(3,reg)
>stat(0,reg) THEN stat(3,reg)=stat(
3,reg)-1:GOTO 890
830 SOUND 7,600,-2,0,1,1:LOCATE 20,
reg+3:PRINT nom$(reg):GOSUB 1300:G
OTO 690
840 IF INKEY(21)<>-1 THEN n=n XOR 1
:LOCATE 48,20+n:PRINT USING masc1%:

```



```

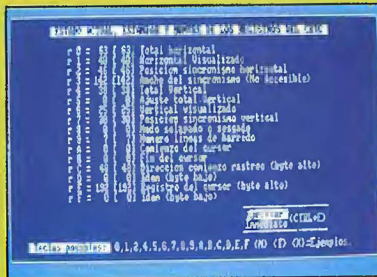
m$(m):LOCATE 48,21-m:PRINT m$(1-m):
GOSUB 1830
850 q$=UPPER$(INKEY$):IF q$="" THEN
840 ELSE RETURN
860 :
870 ':::: OUT :::::
880 :
890 IF m=0 THEN 910
900 OUT &BC00,reg:OUT &BD00,stat(3,
reg)
910 IF TEST(140,84)<>1 THEN GOSUB 1
130
920 LOCATE 20,reg+3:PRINT nom$(reg)
930 LOCATE 8,reg+3:PRINT "*"
940 LOCATE 15,reg+3:IF stat(3,reg)=
stat(2,reg) THEN PRINT USING " ###
";stat(3,reg) ELSE PRINT USING masc
2$;stat(3,reg)
950 IF reg<ureg THEN LOCATE 8,ureg
+3:PRINT " "
960 ureg=reg
970 GOSUB 1300:GOTO 490
980 :
990 ':::: ESTADO DE LOS REGISTROS ::
:
1000 :
1010 ORIGIN 0,0:DRAW 0,399:DRAW 639
,399:DRAW 639,0:DRAW 0,0:MOVE 1,0:D
RAWR 0,399:MOVE 638,0:DRAWR 0,399
1020 FOR a=0 TO 20:INK 0,12:INK 0,6
:INK 0,24:NEXT:INK 0,1
1030 PRINT TAB(6) i$+" ESTADO ACTUA
L, ESTANDAR Y NOMBRE DE LOS REGISTR
OS DEL CRTC "+i$
1040 PRINT
1050 FOR a=0 TO 15
1060 PRINT TAB(10)"r "+CHR$(a+48-(a
>9)*7)+"=";:PRINT USING " ### ";st
at(3,a):LOCATE 20,a+3:PRINT nom$(a)
1070 BORDER 26-a:INK 1,9+a:NEXT
1080 LOCATE 48,20:PRINT USING masc1
$j;m$(0)
1090 LOCATE 48,21:PRINT m$(1)
1100 PLOT 484,84:DRAWR 144,0:DRAWR
0,-40:DRAWR -144,0:DRAWR 0,40
1110 PLOT 482,70:TAG:PRINT<CTRL+E>
";TAGOFF
1120 GOTO 1160
1130 LOCATE 5,20:PRINT "* indica el
ultimo registro modificado"
1140 PRINT TAB(5)"En video inverso
los datos no Estandar"
1150 PLOT 60,84:DRAWR 312,0:DRAWR 0
,-40:DRAWR -312,0:DRAWR 0,40
1160 GOSUB 1300:RETURN
1170 :
1180 ':::: PANTALLA NORMAL :::

```

```

1190 :
1200 FOR a=0 TO 15:OUT &BC00,a:OUT
&BD00,stat(2,a):NEXT
1210 IF tec=19 THEN RETURN
1220 IF tec=18 THEN LOCATE 1,1:END
1230 LOCATE 1,23:PRINT CHR$(18)
1240 LOCATE 10,23:PRINT i$+" PULSE
UNA TECLA PARA VOLVER AL ESTADO AN
TERIOR "+i$
1250 GOSUB 840
1260 GOSUB 1300:IF m=0 THEN RETURN

```



```

1270 FOR a=0 TO 15:OUT &BC00,a:OUT
&BD00,stat(3,a):NEXT
1280 RETURN
1290 :
1300 LOCATE 1,23:PRINT i$+" Teclas
posibles: "+i$+" 0,1,2,4,5,6,7,8,9,
A,B,C,D,E,F <N> <T> <X>=Ejemplos."+
CHR$(18):RETURN
1310 :
1320 ':::: EJEMPLOS :::::
1330 :
1340 GOSUB 1200
1350 LOCATE 1,23:PRINT"ELEGIR EJEMP
LO <1,2,3,4> <ENTER para volver>"+C
HR$(18)
1360 GOSUB 840
1370 ej=INSTR("1234",q$):IF ej=0 TH
EN RETURN
1380 ON ej GOSUB 1540,1620,1700,179
0:RETURN
1390 :
1400 '==== DEMOSTRACION ====
1410 :
1420 FOR a=25 TO 0 STEP-1:GOSUB 147
0:NEXT
1430 MODE 2:WINDOW 5,79,2,24
1440 FOR a=0 TO 25:GOSUB 1470:NEXT
1450 SOUND 7,0,0,0,2,0,31:OUT &BC00
,8:OUT &BD00,1
1460 GOTO 1480
1470 CALL &BD19:OUT &BC00,1:OUT &BD
00,INT(1.6*a):OUT &BC00,6:OUT &BD00
,a:SOUND 2*(a MOD 3),0,7,13,0,0,a+5

```

```

:RETURN
1480 INK 1,8:GOSUB 1010:'ESTADO DE
LOS REGISTROS
1490 OUT &BC00,8:OUT &BD00,0
1500 RETURN
1510 :
1520 '==== EJEMPLO 1 ====
1530 :
1540 FOR a=40 TO 0 STEP -1:GOSUB 15
80:NEXT
1550 GOSUB 1830
1560 FOR a=0 TO 40:GOSUB 1580:NEXT
1570 RETURN
1580 CALL &BD19:OUT &BC00,1:OUT &BD
00,a:RETURN
1590 :
1600 '==== EJEMPLO 2 ====
1610 :
1620 FOR a=25 TO 0 STEP -1:GOSUB 16
60:NEXT
1630 OUT &BC00,1:OUT &BD00,0:GOSUB
1830:OUT &BC00,1:OUT &BD00,40
1640 FOR a=0 TO 25:GOSUB 1660:NEXT
1650 RETURN
1660 CALL &BD19:OUT &BC00,6:OUT &BD
00,a:RETURN
1670 :
1680 '==== EJEMPLO 3 ====
1690 :
1700 FOR a=40 TO 0 STEP -1:GOSUB 17
40:NEXT
1710 GOSUB 1830
1720 FOR a=0 TO 40:GOSUB 1740:NEXT
1730 RETURN
1740 CALL &BD19:OUT &BC00,1:OUT &BD
00,a:OUT &BC00,2:OUT &BD00,a+6:RETU
RN
1750 RETURN
1760 :
1770 '==== EJEMPLO 4 ====
1780 :
1790 FOR b=0 TO 30:CALL &BD19:FOR a
=0 TO 5:GOSUB 1810:NEXT:FOR a=4 TO
0 STEP -1:GOSUB 1810:NEXT:NEXT
1800 RETURN
1810 OUT &BC00,5:OUT &BD00,a:RETURN
1820 :
1830 FOR t=0 TO 500:NEXT:RETURN

```



P ara que tus dedos
no realicen el trabajo duro, M.H. AM-
STRAD lo hace por ti. Todos los lectores que incluyan
este logotipo se encuentran a tu disposición en un cas-
sette mensual, solicitados.