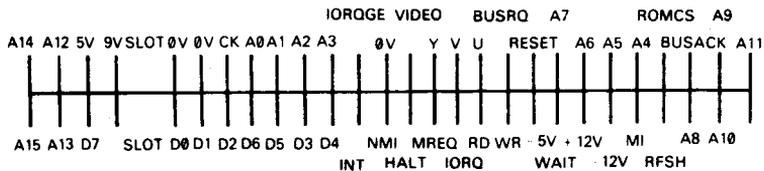


## CAPÍTULO DUODÉCIMO

### Accesorios

#### COMPLEMENTOS DE HARDWARE

El Spectrum tiene un panel en la parte trasera que le permite usar la mayoría de las líneas del Z80A. Tenga en cuenta que el Spectrum usa todo el mapa de memoria en su versión de 48K y que si añade dispositivos de memoria mapeados al de 16K puede causarle problemas si luego quiere poner la ampliación de memoria. Aquí tiene un diagrama del puerto:



Como puede ver trae las líneas de  $-12$ ,  $+12$ ,  $-5$ , y  $+5$  voltios para que sea fácil la implementación de un interface RS232. También están las salidas necesarias para un monitor de vídeo. Para ello necesita unir con un cable la salida VID y el modulador de UHF. También necesitará resistencias de 300 Ohm en serie. Desgraciadamente, Sinclair ha olvidado traer la línea RAMCS, así que no será fácil añadir dispositivos de memoria que cierren la memoria interna cuando se desee. Yo pienso que una línea RAMCS puede ser conectada, ya que la posición que ocupa en el ZX81 (si está familiarizado con él) está vacía (corresponde al conector que está al lado del D7). Si usted sólo tiene 16K de RAM y desea utilizar alguno de los muchos dispositivos que se han creado para ampliar la memoria del ZX81 para

la región del Spectrum comprendida entre los 32 y 64K, puede hacerlo identificando las líneas y colocando un conector intermedio que las reordene como las del ZX81.

También se puede colocar una impresora del tipo Centronics. Sólo necesitaría usar las líneas de la 0 a la 6 para los datos, y la línea 7 como BUSY y STROBE. Esto se puede hacer fácilmente por software, aunque es mejor en código máquina, el principio en BASIC sería:

```
100 POKE xxxx,128  strobe (xxxx es dirección de puerto)
110 POKE xxxx,n    donde n es 128 + datos
120 POKE xxxx,128  strobe
130 IF PEEK xxxx = 128 THEN GOTO 130  ¿todavía
140 etc . . . . .                          ocupado?
```

#### CONVERSACIÓN ENTRE BASICS

Aunque la mayoría de microordenadores trabajan en BASIC, estos BASICS entre sí son un poco diferentes. Otros dos BASICS conocidos son el del ZX81 y el BASIC de Microsoft.

#### EL BASIC DEL ZX81

El ZX81 tiene un BASIC muy parecido al del Spectrum, hasta tiene también los comandos con una sola tecla, pero hay diferencias en cuanto a la gestión de la pantalla y en el hecho de que el ZX81 no dispone ni de color ni de sonido sin un equipo especial añadido. El Spectrum dispone de alta resolución, mientras que el ZX81 no, ya que el punto más pequeño corresponde a una cuarta parte del cuadrado de un carácter. Así, aunque el comando PLOT de ambas máquinas realice una función similar, da unos resultados muy distintos. En el ZX81 se dispone de una matriz de  $64 \times 44$  mientras que en el Spectrum ésta es de  $256 \times 176$ . Quizá la mayor diferencia es que el ZX81 dispone de UNPLOT mientras que con el Spectrum usted debe hacer PLOT con OVER 1.

En otros aspectos, los dos BASICS son idénticos, aunque el Spectrum ofrece más comandos y funciones. También hay pequeños cambios, ya que en el ZX81 CLEAR simplemente borra las variables, y en el Spectrum también coloca

a RAM TOP en la dirección indicada. Por último, diré que el ZX81 usa "\*\*\*" para las potencias cuando el Spectrum utiliza "↑".

#### OTROS BASICS (CON ESPECIAL REFERENCIA AL DE MICROSOFT)

##### *Conjuntos*

En otros ordenadores, DIM X\$ (8) dimensiona un conjunto de 9 elementos, no ocho como en el Spectrum, porque se cuenta también el elemento 0. En algunos ordenadores, se permite dimensionar varios conjuntos en una sola sentencia DIM. En el Spectrum DIM A\$ (6), B\$ (4) debe sustituirse por DIM A\$ (6): DIM B\$ (4).

##### *GET, GET\$, INKEY, INKEY\$*

En el Spectrum sólo tiene la opción de usar INKEY\$ que devuelve la tecla que se está pulsando en aquel momento. Muchos otros BASICS también disponen de esto, pero algunos requieren que se añada detrás un número entre paréntesis que indique por cuánto tiempo se deben esperar. Para hacer esto con el Spectrum tendría que utilizar un bucle FOR NEXT alrededor de INKEY\$.

Algunos BASICS, también disponen de INKEY, que devuelve el código ASCII de la tecla pulsada. GET lo tienen otros ordenadores, y es similar a INKEY. Sin embargo, GET se espera indefinidamente. Finalmente GET\$ es como INKEY\$ pero también espera indefinidamente. El INKEY\$ del Spectrum también puede simular el GET\$ pues para obtener el código de ASCII sólo tiene que hacer: LET A = CODE INKEY\$.

##### *GOTO Y GOSUB*

Algunos ordenadores, no permiten poner una expresión detrás de GOTO sino únicamente un número.

##### *IF-THEN, IF-THEN-ELSE*

Algunos ordenadores le permiten escribir frases como:

IF A\$ = "YES" THEN 400 ELSE...

El Spectrum le obliga a poner GOTO antes del número y no dispone de ELSE que es lo que procede a la sentencia que se ejecutará en caso de que la condición no se cumpla. Otros ordenadores poseen diferente lógica y no permiten algo que hemos usado mucho a lo largo del libreo:

```
10 IF A THEN GOTO 100
```

### *ON-GOTO, ON-GOSUB*

Estas dos instrucciones de las cuales carece el Spectrum trabajan del siguiente modo:

```
10 INPUT X
20 ON X GOTO 200,300, 456, 9998
```

Esto provocaría que el programa saltara a la línea 200 si se introduce un 1, a la 300 si es un 2, a la 456 si se entra un 3 y a la 9998 si lo que se introduce es un 4. Simular esto con el Spectrum no es difícil, basta emplear un uso directo de AND:

```
10 INPUT A
20 GOTO (200 AND A = 2) + (456
AND A = 3) + (9998 AND A = 4)
```

También puede hacerse dimensionando un conjunto:

```
10 DIM A(4)
20 LET A (1) = 200: LET A(2) = 300: LET A(3) =
456: LET A(4) = 9998
```

```
500 INPUT X
510 GOTO A(X)
```

De un modo similar se puede reproducir ON-GOSUB

### *Until*

Until significa hasta. Algunos ordenadores le permiten repetir un proceso hasta que se cumple una cierta condi-

ción. Esto se puede copiar en el Spectrum usando un GOTO y un IF-THEN-GOTO que se salga del bucle si se detecta el cumplimiento de la condición:

```
10 LET A = 0
20 LET A = A + 1
30 IF A<>20 THEN GOTO 20
or
10 IF INKEY$<>"S" THEN GOTO 10
```

### *PROC, END PROC*

Un procedimiento es una subrutina más sofisticada a la que se le pueden pasar variables y que puede ser llamada en cualquier momento. Ya vimos el modo de simular la llamada (asignando a una variable el número de línea en que empieza la subrutina). Pero lo de pasar variables ya es un poco más complicado.

### *LET*

Algunos BASICS permiten omitir la palabra LET, pero en el Spectrum no se puede.

### *LEFT\$, RIGHT\$, MID\$, ETC*

En otros ordenadores, la manipulación de cadenas se hace usando estos comandos. Aquí se muestran sus equivalentes con los del Spectrum:

```
LEFT$ (A$,3)      A$ (TO 3)
RIGHT$ (A$,4)     A$[(LEN A$ - 4) TO LEN A$]
MID$ (A$,1,3,)    A$(1 TO 3)
```

INSTR\$ también existe en algunos ordenadores, y tiene la forma:

INSTR\$ (A\$, B\$)

y busca la presencia de la segunda cadena en el interior de la primera. En el Spectrum se puede substituir por esta rutina:

```
10 FOR A = 1 TO LEN A$ - LEN B$
20 IF A$(A TO A + LEN B$) = B$ THEN PRINT
   "YES"
30 NEXT A
```

### *Matemáticas*

Algunos BASICS permiten definir números enteros y de coma flotante; % significa entero. DIV da el número entero resultado de la división:

```
PRINT 13 DIV 4
```

da como resultado 3 (olvidando el resto de 1). Este resto, se encuentra usando MOD:

```
PRINT 13 MOD 4
```

da como resultado 1.

Algunos BASICS también pueden convertir números entre decimal y hexadecimal usando HEX o H. Además un número se puede definir directamente en hexadecimal poniendo previamente HEX o H. Así HA4 indica que 44 es un número hexadecimal.

### *PLOT*

En muchos ordenadores en lugar de PLOT hay que usar SET y en lugar de PLOT OVER hay que usar RESET. Para la alta resolución existen dos palabras más, que son PSET y PRESET. En algunos ordenadores, se usa LINE en lugar de DRAW, del siguiente modo:

```
LINE (5,3) — (6,6), PSET
```

Esto dibuja una línea desde el punto (5,3) hasta el punto (6,6). En el Spectrum hay que calcular el punto final a partir del inicial usando una combinación de PLOT y DRAW.

### *Almacenamiento en cinta*

Muchos ordenadores tienen CLOAD y CSAVE en lugar

