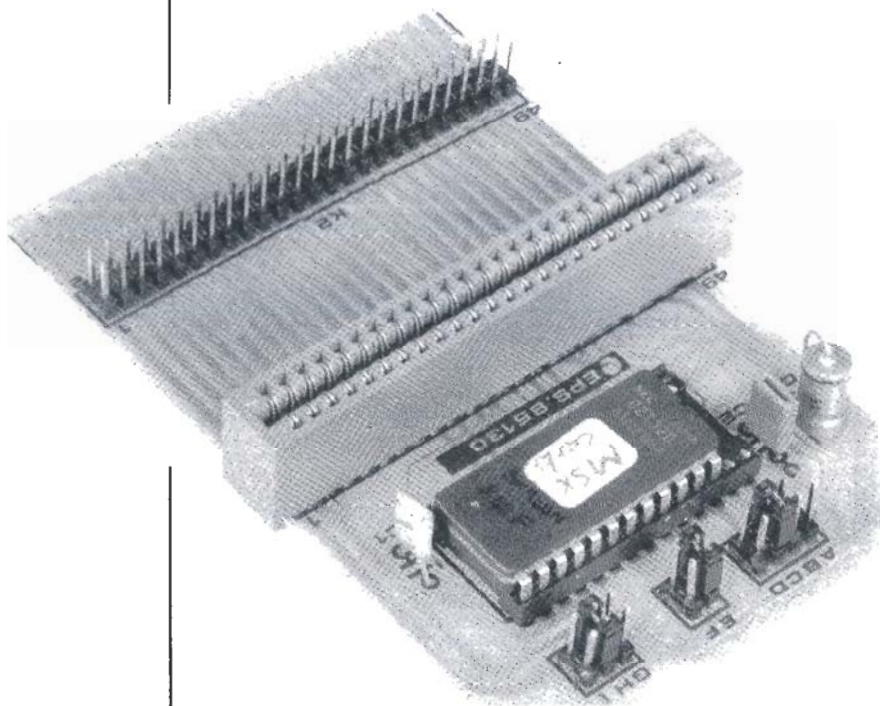


El conector de cartuchos de los MSX permite conectar tanto expansiones, por ejemplo el bus universal, como programas propios. Conozca con todo detalle las posibilidades de una pequeña placa que permite realizar ambas funciones.



MSX

EXTENSIONES

MSX (2)

Como se apuntaba en nuestro anterior artículo sobre los MSX, el conector de cartuchos se puede emplear para otro tipo de conexiones diferentes a simples juegos. Habitualmente los cartuchos comerciales contienen una

(E)PROM grabada con un programa (juegos o utilidades). Es realmente sencillo construirse uno mismo un dispositivo para contener los programas, o utilidades, propios manteniendo la posibilidad de conectar cartuchos comer-

ciales. Nuestros diseñadores le ofrecen las siguientes posibilidades:

1. Conexión fácil de expansiones hardware, como el bus universal de E/S.
2. Con un cable plano de 50 vías, se puede conectar al bus de expansión existente en algunos MSX, como el Spectravideo.
3. La placa permite conectar otro cartucho a su vez, gracias al conector hembra de la misma.
4. También está preparada para recibir memorias EPROM con 2, 4, 8, 16 ó 32 Kbytes de capacidad, permitiendo la depuración y grabación de aplicaciones personales.
5. Se puede emplear como dispositivo de conexión con un módulo sintetizador, por ejemplo el Yamaha.

Las posibilidades de esta reducida tarjeta doble cara no acaban aquí; los usuarios descubrirán por su cuenta muchas otras. La tarjeta sirve de base para un sin fin de aplicaciones, que pueden ocupar a los lectores durante largo tiempo, y todo ello con un reducido precio.

El cartucho MSX

Como se ha enumerado anteriormente, la tarjeta presenta unas características muy atractivas para el usuario: entre otras permite la medida, de forma accesible, de las líneas internas del ordenador, así como recibir los programas desarrollados por él mismo. Sin embargo para realizar todo esto se necesitan unos conocimientos previos sobre las normas de conexión de los cartuchos empleados en los MSX. Vamos a examinar primero el procedimiento típico ejecutado en el arranque del ordenador.

Tras el encendido, el Basic del MSX determina siempre la cantidad de memoria RAM disponible, entre las direcciones \$8000 y \$FFFF; el Basic toma, para los programas y variables, la mayor zona de RAM contigua dentro de este área. A continuación el Basic examina la zona de conectores en el margen \$4000...\$BFFF. Cada conector ("slot") ocupa 16 K, divididos en 4 páginas. Al principio de cada página debe existir una serie de códigos, que el ordenador lee para determinar el tipo de expansión conectada. Los có-

digos que proporcionan esta información se disponen en un orden fijo, tal como se ve en la figura 1. La función de cada código es la siguiente:

ID (identificación): un código de dos bytes que indica la presencia de un cartucho (EPROM. En caso de existir la memoria el Basic lee \$41 y \$42 ("A" y "B" en ASCII) respectivamente en estas posiciones.

INIT (inicialización): un vector (puntero de dirección) de la rutina de inicialización asociada con la función del cartucho; programación de periféricos e inicialización de variables. Si no es necesaria esta rutina el contenido de ambas posiciones es de 00.

STATEMENT: dos octetos que contienen la dirección de una

sub-rutina más extendida de tratamiento de instrucciones, si es necesaria. En caso contrario ambos bytes contienen 00. Para una mayor información dirigirse al manual del propio MSX.

DEVICE: un par de bytes que contienen la rutina de tratamiento del dispositivo (device) físico (normalmente un periférico) presente en el cartucho, si existe. El valor por defecto es también de 00. Para una información más amplia consultar el manual del ordenador.

TEXT: vector, dos bytes, que apunta a la dirección de comienzo de los mensajes (en código ASCII o palabras claves, "token", del Basic) en el cartucho. En Basic las instrucciones se codifican por el número de orden de la instrucción

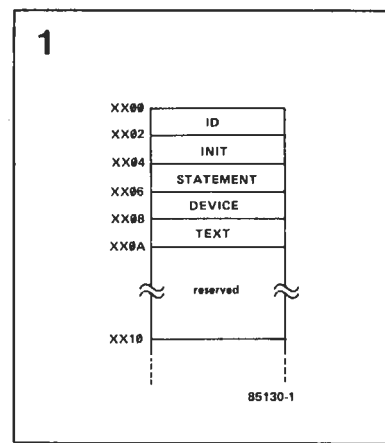


Figura 1. La presencia de estos datos al principio de la EPROM es necesaria para que el programa sea identificado correctamente por el MSX Basic.

en una tabla de consulta. En una segunda tabla, con el mismo número de orden, se encuentra la dirección de comienzo de la rutina correspondiente a la ejecución de

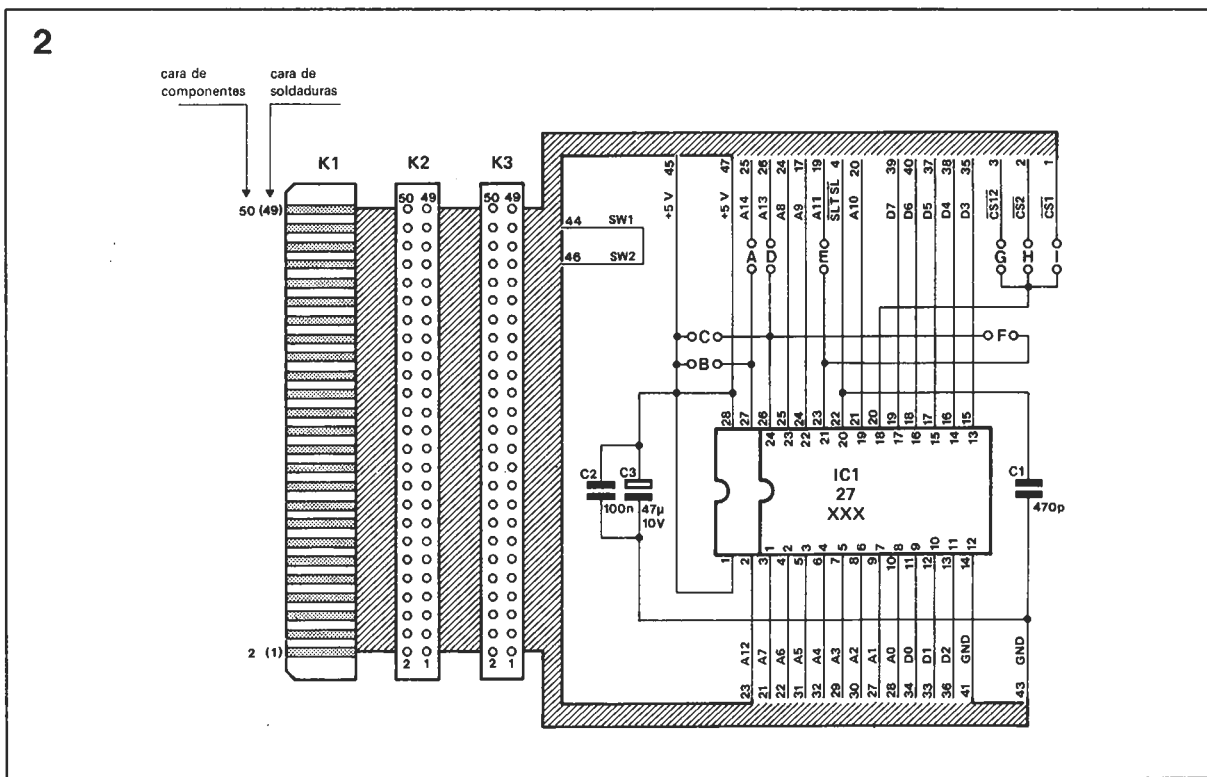
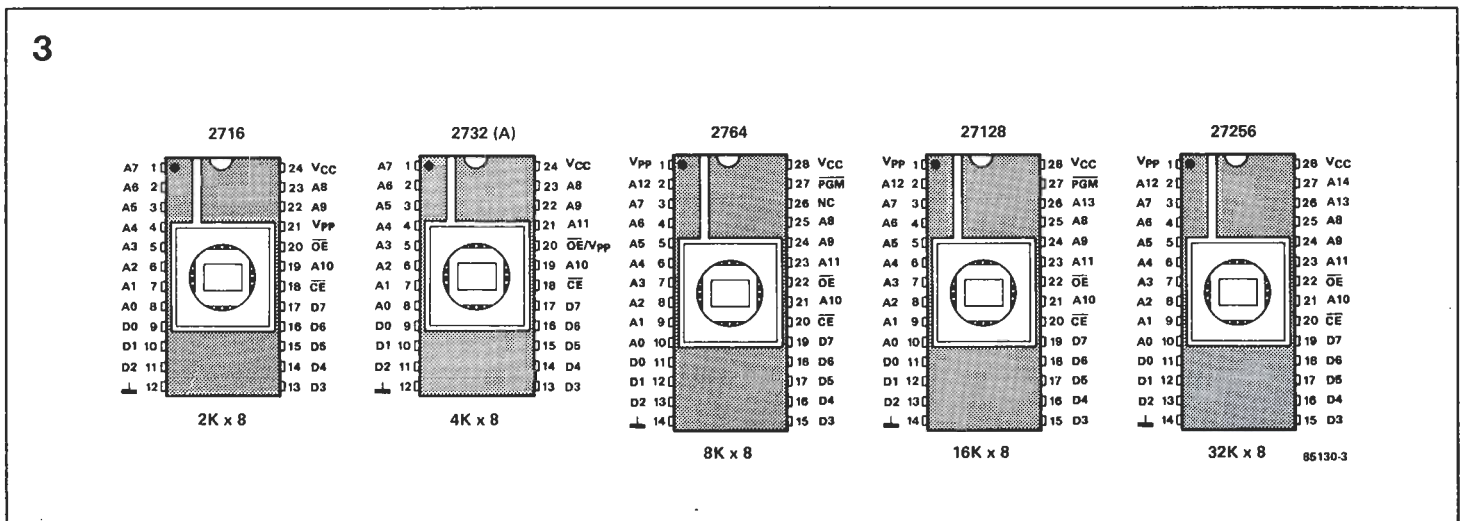


Figura 2. Esquema del circuito para cartuchos. La placa está prevista para recibir prácticamente cualquier tipo de memoria EPROM, desde la 2716 a la 27256.

Figura 3. Patillaje de las diversas memorias de la familia 27XX. Los puentes de la placa realizan la adaptación según el modelo concreto.



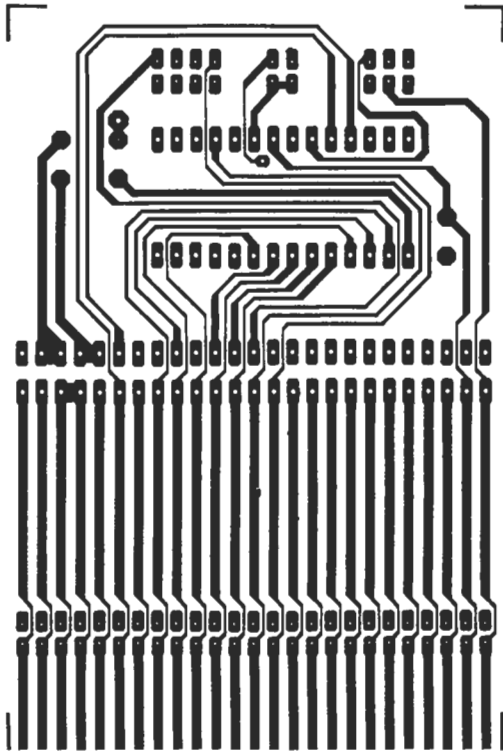


Figura 4. El circuito impreso es pequeño pero efectivo, gracias a sus conectores y zócalo capaz de recibir cualquier EPROM.

las instrucciones. Si no se trata de un programa en Basic estos dos bytes están a cero. Las direcciones se almacenan, como es habitual en el Z80, colocando primero el byte menos significativo, LSB, y a continuación el más significativo, MSB. Este puntero es de gran interés para aquellos que desean guardar sus propios programas en Basic dentro de una EPROM. Volveremos posteriormente sobre este punto.

Esquema práctico

Un vistazo a la figura 2, muestra que el término de esquema

para el circuito parece un poco presuntuoso. Sería más preciso decir que es el soporte de un circuito polivalente, adaptado para conectarse en lugar de un cartucho MSX. El circuito impreso permite colocar una memoria tipo 27XX. En él tiene cabida desde la 2716 (2K X 8) hasta la gigante, por el momento, 27256 (32K X 8). Como se aprecia los fabricantes emplean una nomenclatura que indica, tras el 27 inicial, la capacidad de la memoria.

Otra de las peculiaridades de la familia 27XX es la compatibilidad de patillaje. Con sólo unos puentes se selecciona el tipo de memoria a emplear; de esta forma no

es necesario cortar pistas o realizar delicadas soldaduras para pasar de un modelo a otro. Para poder apreciar las diferencias entre los diversos modelos, en la figura 3 se muestran el patillaje de las memorias citadas. La colocación de los puentes se realiza como sigue:

Puente A: selección entre EPROM 27128 ó 27256. Sólo se instala con este último modelo.

Puente B: alimentación para la 27128. Por tanto, puente A para 27256 y puente B para 27128.

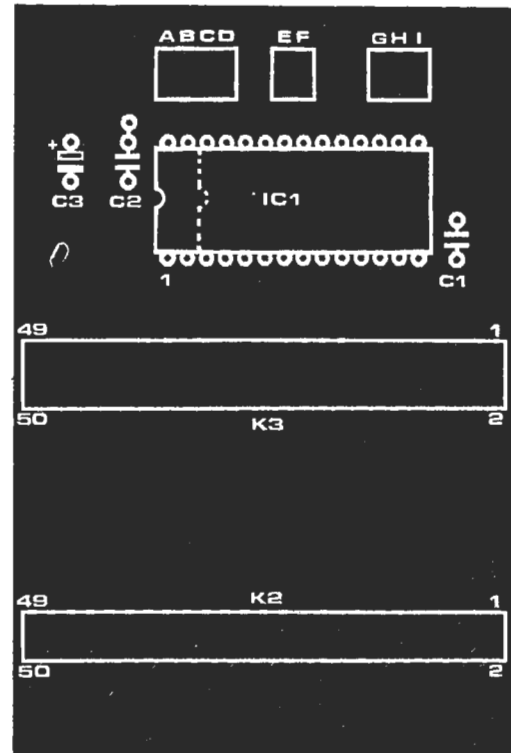
Puente C: alimentación para los modelos 2716 y 2732, patilla 24.

Puente D: conexión de la línea A13 al terminal 26 de las 27128 y 27256 (de 28 patillas). Para la 2764 hay que colocar el puente C, patilla 26 a + 5, pero no el D.

Puente E: conecta la línea A11 en la patilla 23 (para las memorias de 28) ó 21 (2716). Debe ser instalado para todas las EPROM excepto la 2716.

Puente F: conecta el terminal 21 de la 2716, Vpp, a + 5.

Puente G, H, I: unión entre la señal de selección de la EPROM, \overline{CE} (Chip Enable) a las líneas $\overline{CS1}$, $\overline{CS2}$ ó $\overline{CS12}$ en este orden. $\overline{CS1}$ es la señal de selección para direcciones entre \$4000...\$7FFF, $\overline{CS2}$ para direcciones entre \$8000...\$BFFF, y $\overline{CS12}$ incluye ambos márgenes, esto es, para direcciones entre \$4000...\$BFFF. Para la 27256 debe emplearse



Listado 1. Programa de listado hexadecimal empleado para analizar el contenido de una zona de memoria. La salida del mismo puede realizarse a través de pantalla sin más que cambiar los LPRINT por PRINT, y el valor "15", en las líneas 60 y 90, por "7".

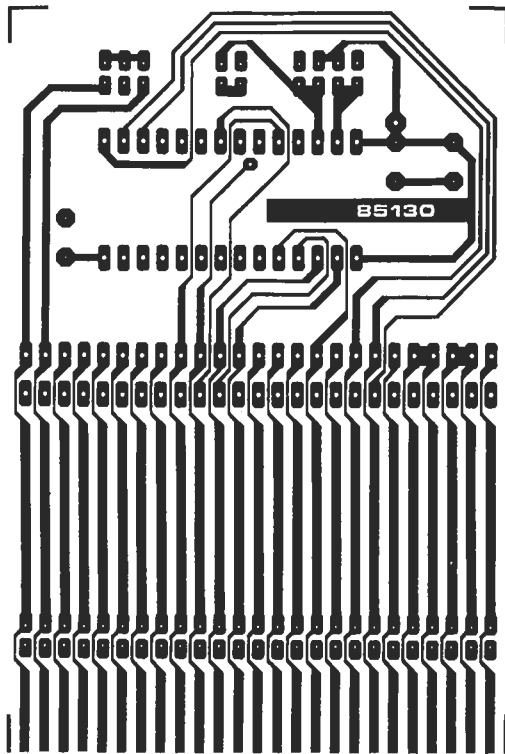
Listado 1.

DUMP

```

10 CLS
20 INPUT "comienzo";A
30 INPUT "fin";B
40 FOR C = A TO B
50 LPRINTUSING"\ \";HEX$(C);:LPRINT " ";
60 FOR D=0 TO 15
70 LPRINTUSING"\\";HEX$(PEEK(C+D));:LPRINT " ";
80 NEXT
90 C=C+15:LPRINT " ":LPRINT " "
100 NEXT
110 END

```



CS12, mientras que para el resto puede conectarse CS1 o CS2 según se prefiera; hay que tener en cuenta, en el caso de programas en código máquina cual de ambas está conectada, para evitar errores en las direcciones absolutas. La tabla 1 resume las configuraciones a emplear con cada tipo de memoria EPROM.

Desde luego la placa puede contener una ROM, si bien no mencionaremos los posibles tipos, dado que es más difícil, y costoso, la programación de estas memorias. En caso de conectar alguna ROM deberá comprobarse el patillaje de la misma para realizar la configuración de puentes adecuada.

Cuando se conecta al MSX, el puente, fijo, realizado entre las patillas 44 y 46 (SW1, SW2) sirven de dispositivo de detección de la protección para enchufar/desconectar el cartucho. Esta protección garantiza una seguridad total durante la manipulación de los cartuchos.

La placa dispone de tres conectores: el primero, K1, es simplemente el conector realizado con las pistas de cobre, en ambas caras, de la tarjeta. Este conector se enchufa directamente en la hembra del MSX prevista para los cartuchos. El conector K2 se realiza con ayuda de dos tiras, o una tira doble, de 25 contactos con separación de 2.54 mm. K3 es un co-

nectador hembra, para circuito impreso, de 2 x 25 similar al existente dentro del ordenador. En él se conectan los cartuchos de igual forma que si se realizara directamente sobre el ordenador.

Construcción

El circuito impreso, así como la distribución de componentes, se ven en la figura 4. Se trata de un circuito impreso de doble cara, con taladros metalizados, solución adoptada por su fiabilidad y reducido tamaño. Para garantizar un buen contacto, no sólo inicial,

y larga duración se ha realizado un estañado del conector K1. Para el zócalo de IC1 se recomienda emplear un "zócalo de fuerza de inserción nula", lo que facilitará el cambio de las diversas EPROM de forma rápida, evitando también doblar patillas o mal contacto por no insertar la memoria a fondo. En caso de emplear un zócalo normal, porque no necesita realizar frecuentes cambios, o por motivos económicos, utilice un zócalo de buena calidad. Un zócalo barato acabará realizando un mal contacto de terminales, tras un uso prolongado, con los problemas de ello derivados. Dada la sencillez no hay ninguna otra cosa digna de mención, si se emplea un soldador fino, y no demasiado potente, y estaño de calidad para realizar las soldaduras.

Aplicaciones

Ahora que se dispone de un zócalo universal para EPROM, los programas empleados más frecuentemente pueden almacenarse en una memoria, ahorrando el tiempo de carga desde cinta o disco. El programa, una vez trasladado a EPROM, actuará como los cartuchos disponibles comercialmente. Sin embargo, antes de proceder a grabar ésta, de forma satisfactoria, hay que conocer el método de almacenamiento del MSX Basic en la memoria del ordenador.

Hay que resaltar que la descripción que realizamos a continuación no es válida para los programas en código máquina, ya que éstos requieren un sistema más complejo en el empleo de los diferentes vectores.

Para un programa en Basic los vectores ID y TEXT son esencia-

Lista de componentes

Condensadores

C1 = 470 p
C2 = 100 n
C3 = 47 µ/10 V

Semiconductores:

IC1 = 2716, 2732, 2764; 27128, 27256 o (P)ROM cuyo patillaje sea compatible.

Varios:

K2 = conector hembra de 50 (2 x 25) para circuito impreso.
4 puentes
PCB 85130 (65,5 x 98 mm)

Tabla 1

	A	B	C	D	E	F	G	H	I
27256	○—○			○—○	○—○		○—○		
27128		○—○		○—○	○—○			○*○	○*○
2764		○—○	○—○		○—○			○*○	○*○
2732			○—○		○—○			○*○	○*○
2716			○—○			○—○		○*○	○*○

○—○ = puente

* = seleccione H o I (ver texto)

85130-T1

Tabla 1. Resumen de los puentes a realizar dependiendo del tipo de memoria empleada. La elección entre H o I depende de la zona de memoria deseada (ver texto).

Tabla 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8000	0	7 L8007	80	A	0	9F Tk	0	16 L8016	80	14	0	85 Tk	22	87 s	74 t	61 a
8010	72 r	74 t	22	3B ;	41 A	0 EOL	23 L8023	80	1E *	0	85 Tk	22	65 e	6E n	64 d	22
8020	3B ;	42 B	0 EOL	33 L8033	80	28 *	0	82 Tk	20 sp	43 C	20 sp	EF Tk	20 sp	41 A	20 sp	D9 Tk
8030	20 sp	42 B	0 EOL	4E L804E	80	32 *	0	9D Tk	E4 Tk	22	5C /	20 sp	20 sp	5C /	22	3B ;
8040	FF Tk	9B Tk	28 (43 C	29)	3B ;	3A :	9D Tk	22	20 sp	20 sp	22	3B ;	0 EOL	5D L805D	80
8050	3C *60	0	82 Tk	20 sp	44 D	EF Tk	11 0	20 sp	D9 Tk	20 sp	F Tk	F 15	0 EOL	7B L807B	80	46 *
8060	0 70	9D Tk	E4 Tk	22	5C /	5C /	22	3B ;	FF Tk	9B Tk	28 (FF Tk	97 Tk	28 (43 C	F1 Tk
8070	44 D	29)	29)	3B ;	3A :	9D Tk	22	20 sp	22	3B ;	0 EOL	81 L8081	80	50 *	0	83 Tk
8080	0 EOL	96 L8096	80	5A *	0	43 C	EF Tk	43 C	F1 Tk	F Tk	F 15	3A :	9D Tk	22	20 sp	22
8090	3A :	9D Tk	22	20 sp	22	0 EOL	9C L809C	80	64	0	83 Tk	0 EOL	A2 L80A2	80	6E	0
80A0	81 Tk	0 EOL	0	0	8	41 A	0	C5	32	76	80	0	0	0	0	8
80B0	42 B	0	C5	32	51	20	0	0	0	0	8	43 C	0	C5	32	59
80C0	60	0	0	0	0	8	44 D	0	41	90	0	0	0	0	0	0
80D0	3A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80E0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80F0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- * : número de líneas
- ▨ : dirección de unión
- Tk : byte con token
- sp : espacio
- EOL : fin de línea BASIC
- ▧ : fin de programa

Tabla 2. Listado hexadecimal, realizado por el programa DUMP, sobre sí mismo, cuando reside en RAM. Analizados los datos se puede volver a reconstruir el programa del listado 1.

les; están situados en las posiciones \$XX00-\$XX01 y \$XX08-\$XX09 respectivamente (ver figura 1). Debido a que las primeras 16 posiciones del cartucho están reservadas para la identificación del programa y los vectores, el programa, codificado en "token" (palabras clave) deberá situarse a partir de la dirección \$XX10 (\$ indica hexadecimal). Los programas Basic se almacenan, generalmente, en memoria, desde la dirección \$8000 en adelante. Por tanto, a partir de ahora, se toma-

rá \$80 como valor para XX. En \$8010 la UCP debe encontrar, obligatoriamente, un 00. La siguiente dirección contiene la dirección de unión (link address) de dos bytes seguido por el número de línea, también de dos bytes; a continuación viene una línea de Basic, codificada en "token", terminada con un byte a 00. Este procedimiento se repite hasta el final del programa. La llamada dirección de unión indica la dirección de memoria donde comienza la siguiente línea de Basic.

Antes de grabar el programa Basic en EPROM hay que realizar un pequeño tratamiento. Para extraer los códigos, en hexadecimal, es necesario ejecutar el programa DUMP del listado 1, preferiblemente con una impresora conectada al ordenador. En caso de no disponer de impresora, habrá que cambiar todas las sentencias LPRINT del programa por PRINT, para visualizarlo en la pantalla, y cambiar el valor 15, de las líneas 60 y 90, por 07. Como el programa DUMP debe estar también en memoria se cargará con números de línea altos, por ejemplo a partir de 10.000.

Tras realizar un RUN 10.000 (o el número de línea de comienzo asignado a DUMP) el programa pide la dirección de comienzo y de final del programa; la dirección de comienzo es siempre, en lo que a nosotros concierne, \$8000. La dirección final depende, evidentemente, de la longitud del programa (menos 160 bytes ocupados por el programa DUMP). El tratamiento realizado por DUMP se muestra en la tabla 2: un listado de la zona de memoria. Una vez obtenido el listado hexadecimal, se borra la memoria de programa, NEW, y se vuelve a cargar de nuevo DUMP, a partir de la línea 10 por ejemplo. Al ejecutarlo, introduzca \$8000 como dirección de comienzo y \$8100 como dirección de final, con lo que obtendrá un listado de este pequeño programa. Con ayuda de la tabla 2, intente ver la equivalencia con las líneas de Basic, para entender el funcionamiento del principio de almacenamiento del programa en la memoria del MSX. Observe que la dirección de unión, y el número de línea están en orden inverso al habitual, esto es, primero el byte menos significativo, LSB. Todos los comandos Basic tienen una "palabra clave" o token. No resultará difícil descubrirlas con un poco de práctica; puede comprobar por ejemplo: \$82 = FOR, \$9D = LPRINT, \$EF = "=" (signo de igualdad), \$83 = NEXT, \$F1 = "+", \$E4 = USING, etc. Si ya ha quedado claro podemos pasar a considerar los datos a introducir en la EPROM.

Grabación

Por lo comentado al principio, resulta evidente que el ordenador no considera el programa residente en \$8000 como parte de un cartucho. Para que el intérprete

sepa realmente que es así hay que añadir los bytes de identificación y vectores adecuadamente (\$8000-\$800F). Para que el programa DUMP le proporcione los datos correctos no haya que olvidarse de desplazar 16 posiciones la dirección de comienzo. Así pues, la dirección de comienzo de programa debe trasladarse a \$8010, e incrementar, en \$16, las direcciones de unión; sin olvidarse de los códigos en las primeras posiciones.

Como ejemplo práctico veamos como se realiza con ayuda de la tabla 3; de nuevo aquí hemos empleado el propio programa DUMP, pero esta vez tal como debe quedar para ser ejecutado desde la EPROM. Los programas así realizados arrancan automáticamente, aunque no es posible modificarlos o editarlos como se hace con los programas que residen habitualmente en RAM. No hay que perder de vista que las variables del programa permanecen en RAM, ya que se necesita su modificación.

Finalmente unas líneas sobre programas muy largos o muy complicados, y su almacenamiento en EPROM. Como ya se ha sugerido el programa DUMP se colocará con un número de línea más alto que el final del programa, por ejemplo 10.000. Una vez totalmente depurado el programa, ejecute DUMP (RUN 10.000). Localice las direcciones de unión (link address) de cada línea y súmeles \$10. Por último coloque el comienzo del programa en la dirección \$XX10 (por ejemplo \$8010) y escriba la secuencia de bytes. Localizar y alterar las direcciones de unión no es tan complejo como pueda parecer a primera vista. Cada una de ellas apunta a la siguiente, lo que facilita el trabajo. Un pequeño truco le ahorrará este delicado trabajo de suma hexadecimal: inserte una línea "1 REM ABCDEFGHI". Como puede comprobar con el programa DUMP, su longitud exacta (incluido el espacio entre REM y A) es de 16, por lo que la primera dirección útil del programa queda situada en \$8010, y no hay que alterar las direcciones de unión. Como se trata de una línea de comentario su eliminación no afecta al correcto funcionamiento del programa. El final del programa se identifica porque la dirección de unión es 0000; no hay que olvidar que esta es la dirección de fin de programa que incluye al programa DUMP. Para localizar

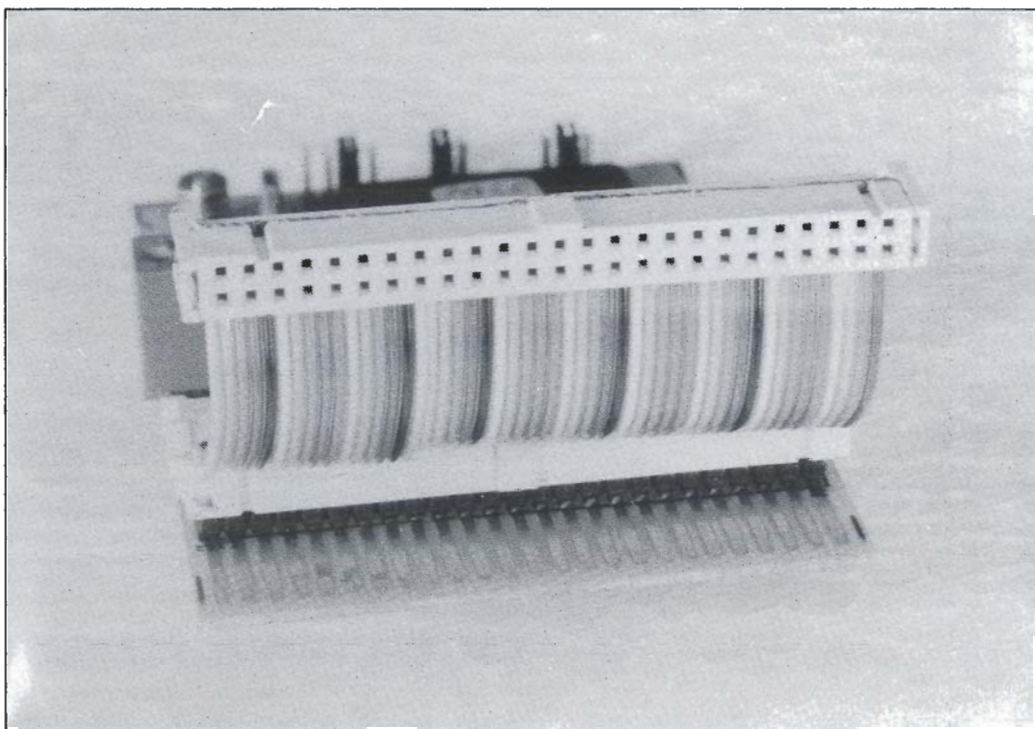


Figura 5. Al Spectravideo se le puede dotar de un conector de cartuchos, realizado con un trozo de cable plano de 50 y dos conectores.

este último busque, en los códigos de "número de línea" el equivalente hexadecimal de 10.000, o el número que haya empleado. En el caso de 10.000 este código es \$1027, recordando, como ya se mencionó anteriormente, que están en orden inverso los bytes más significativo y menos significativo. Una vez localizada la dirección que contiene este código, sustituya los dos bytes anteriores, la dirección de unión, por 0000.

Con la EPROM programada como hemos explicado, la zona de RAM comienza en \$C000. Las variables del programa aparecen a partir de \$C010, como se puede comprobar con ayuda del programa DUMP.

Un caso particular de los MSX es el Spectravideo, que no dispone de un conector de cartuchos propiamente dicho. Sin embargo, este ordenador posee un conector de expansión de 2 x 25, apto para recibir un conector de cable-cable plano. De este conector se puede

realizar una toma, como se aprecia en la figura 5, dotada de un conector hembra del tipo que incorporan los cartuchos. Hay un pequeño detalle que conviene resaltar; el conector de salida del Spectravideo tiene una distribución que se corresponde con la del conector de cartuchos, pero en realidad está realizado "en espejo" respecto a éste. Esto explica el porqué la flecha que indica la patilla 1 se corresponde, realmente, con la patilla 50. Hombre prevenido vale por dos, así que más vale prestar atención a este caso curioso dentro de la familia de los MSX. Si se sigue el ejemplo de la figura 5 se encontrarán pocos problemas.

Aquellos que deseen conectar su MSX al sintetizador Yamaha pueden emplear la placa cortando entre los conectores K2 y K3 de la misma, de forma que únicamente se conserva el zócalo K2 para realizar un adaptador macho-macho de conexión. ■

Tabla 3. Los datos a introducir en la EPROM para disponer de la utilidad DUMP sobre el cartucho. Las direcciones sombreadas (las direcciones de unión) están sumadas con \$10 debido al desplazamiento necesario para albergar la cabecera de la EPROM.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8000	41	42	0	0	0	0	0	0	10	80	0	0	0	0	0	0
8010	0	17	80	A	0	9F	0	26	80	14	0	85	22	73	74	61
8020	72	74	22	3B	41	0	33	80	1E	0	85	22	65	6E	64	22
8030	3B	42	0	43	80	28	0	82	20	43	20	EF	20	41	20	D9
8040	20	42	0	5E	80	32	0	9D	E4	22	5C	20	20	5C	22	3B
8050	FF	9B	28	43	29	3B	3A	9D	22	20	20	22	3B	0	6D	80
8060	3C	0	82	20	44	EF	11	20	D9	20	F	F	0	8B	80	46
8070	0	9D	E4	22	5C	5C	22	3B	FF	9B	28	FF	97	28	43	F1
8080	44	29	29	3B	3A	9D	22	20	22	3B	0	91	80	50	0	83
8090	0	A6	80	5A	0	43	EF	43	F1	F	F	3A	9D	22	20	22
80A0	3A	9D	22	20	22	0	AC	80	64	0	83	0	B2	80	6E	0
80B0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0