

ZXSpectr Versión 3.5

Emulador de ZX Spectrum por César Hernández Bañó

15-09-2013

Índice

1 Historia.....	4
2 Agradecimientos.....	6
3 Contactar con el autor.....	7
4 El Emulador.....	8
4.1 Ficheros distribuidos.....	8
4.2 Requisitos para usar el emulador.....	9
4.3 Línea de comandos de ZXSPCTR.COM.....	10
4.4 Emulación del Spectrum.....	10
4.4.1 La CPU Z80.....	11
4.4.2 El Teclado.....	12
4.4.3 El Joystick Kempston.....	12
4.4.4 El Sonido.....	13
4.4.5 La Pantalla.....	14
4.4.6 128k de memoria.....	15
4.4.7 Emulación de Cinta.....	16
4.4.8 Carga directa desde cinta de casete.....	17
4.4.9 Ficheros Snapshot.....	18
4.4.10 Emulación del Inves Spectrum+.....	23
4.4.11 El puerto "bus idle".....	24
5 Opciones del menú.....	26
5.1 Cargar Snapshot.....	26
5.2 Grabar Snapshot.....	26
5.3 Selección de Ordenador.....	26
5.4 Ajustes de Pantalla.....	26
5.4.1 Cargar Pantalla.....	26
5.4.2 Grabar Pantalla.....	26
5.4.3 Control de Brillo.....	27
5.4.4 Protector de Pantalla.....	27
5.4.5 Iniciar grabación vídeo.....	27
5.4.6 Fichero.....	27
5.4.7 FPS.....	28
5.5 Menú Depuración.....	28
5.5.1 Generar RESET.....	28
5.5.2 Generar NMI.....	28
5.5.3 Ver Registros.....	28
5.5.4 Pokear.....	28
5.6 Emulación de Cinta.....	28
5.6.1 Insertar/Extraer fichero de Entrada.....	28
5.6.2 Fichero.....	29
5.6.3 Insertar/Extraer fichero de Salida.....	29
5.6.4 Fichero.....	29
5.6.5 Cargar cualquier flag.....	29
5.6.6 Cargar desde casete.....	29
5.6.6.1 Baudios.....	29

5.6.6.2	Longitud onda guia.....	29
5.6.6.3	Longitud tono guia.....	30
5.6.6.4	Longitud onda ceros.....	30
5.6.6.5	Longitud onda unos.....	30
5.6.6.6	Filtro de Entrada.....	30
5.6.6.7	Volumen canal izq.....	30
5.6.6.8	Volumen canal der.....	30
5.6.6.9	Ver Franjas del Borde.....	30
5.6.6.10	Autocorrección de Checksum.....	30
5.6.6.11	Comenzar a Cargar.....	31
5.7	Ajustes de Hardware.....	31
5.7.1	Teclado Issue0/1.....	31
5.7.2	Disparador Automático.....	31
5.7.3	Frec. disparador.....	31
5.7.4	Cursores emulan.....	32
5.7.5	Valor POKE a ROM.....	32
5.7.6	Memoria compartida.....	32
5.7.7	Sincronismo.....	32
5.8	Ajustes de sonido.....	32
5.8.1	Sonido Activo.....	32
5.8.2	Chip AY presente.....	32
5.8.3	Modo de Stereo.....	32
5.8.4	Emulación de Ruido.....	33
5.8.5	Ver Registros Chip AY.....	33
5.8.6	Iniciar grabación audio.....	33
5.8.7	Fichero.....	33
5.9	Selección de Idioma.....	34
5.10	Multitarea.....	34
5.11	Velocidad CPU.....	34
5.12	Ayuda.....	34
5.13	Volver al Emulador.....	34
5.14	Salir del Emulador.....	34
6	Desarrollo y compilación.....	35
7	Utilidades incluidas.....	36
7.1	LINEASMP.....	36
7.2	SMPATAP.....	36
7.3	TAPABIN.....	36
7.4	SP_Z80.....	37
7.5	VGA2RAW.....	37
8	Utilidades en formato cinta (TAP).....	38
8.1	SPED52.TAP.....	38
8.2	CONVERSO.TAP.....	39
8.3	REALDEBU.TAP.....	39
8.4	CURSORDR.TAP.....	40

1 Historia

Desde que soy muy pequeño, en mi casa ya había un ordenador; he tenido varios, la mayor parte de ellos son Sinclair. El primero que tuvimos fue un Sinclair ZX-81. Tenía un teclado completamente plano, no media más de un palmo de ancho y se calentaba una barbaridad. Funcionaba con solo 1K de memoria RAM (que luego ampliamos a 16k), y el procesador era un Z-80. El sistema de almacenamiento era en cinta de casete, a una velocidad de 300 baudios.

Después tuvimos el ZX Spectrum 48k, de teclas de goma. A aquél ordenador le sacamos mucho provecho, aparte de tener que cambiarle unas tres veces la membrana del teclado y otras tantas el disipador de calor.

En 1984, cuando el Spectrum ya había tenido éxito, Sinclair lanzó un mega-ordenador, el QL, con procesador Motorola 68008, que también compramos. Los que hayan tenido un QL sabrán que al poco tiempo de salir a la venta, la casa Sinclair fue absorbida por Amstrad y, por lo menos aquí en España, se dejó de hacer programas para él.

Después de esto compramos dos Inves Spectrum+, que era como un Spectrum+ de Sinclair pero de la marca Investrónica, que tenía el inconveniente de que no se oía el sonido en varios juegos (se podía oír pokeando en la ROM!). Con el tiempo pude conseguir un Spectrum +2A, a cambio de un Inves.

Ya en los tiempos del PC, yo todavía seguía programando con el +2A, y tuve el primer contacto con un emulador, el SPECTRUM de Pedro Gimeno, y más tarde, el Z80 de Gerton Lunter. Desde entonces, tuve ganas de crear un emulador propio, y más que nada, poder cargar mis viejos juegos del Spectrum sin tener que comprar ningún emulador ni hacer ningún circuito electrónico.

En 1996 hice la primera versión, escrita totalmente en Assembler, usando la ROM del emulador de Pedro Gimeno y los ficheros SP. No fue hasta dos años más tarde, en 1998, cuando compramos un Pentium 133 con tarjeta Sound Blaster y pude hacer una rutina para cargar los juegos desde cinta de casete. Entonces pude cargar las ROMs de mi +2A y le añadí al emulador los 128k y la emulación del chip de sonido.

Desde entonces he ido haciendo varias versiones y mejorando constantemente el emulador, y lo he ido distribuyendo a mis amigos. Ha sido a partir de la versión 1.5 cuando empecé a distribuirlo a través de Internet.

El emulador no funciona al 100% con todos los juegos, pero no he encontrado todavía el bug. Agradecería a cualquiera que encuentre el fallo me lo notificase, para así poder hacer un emulador completo. Creo que el error tiene que ver con el flag P/V.

Pese a esto, creo que el emulador funciona bastante bien, tiene diversas opciones que no he encontrado en otros emuladores, y además no hay que pagar nada por usarlo, pudiendo incluso cargar los juegos desde cinta de casete.

Hace mucho tiempo me dije a mi mismo que no haría nuevas versiones, pero de vez en cuando se me ocurre añadir nuevas funcionalidades y saco versiones nuevas. Aunque mi intención en el futuro era crear una versión en Linux, esto de momento no está hecho. Sí que he acabado desarrollando en Linux (con el emulador de MSDOS dosbox) y traspasando algunas utilidades a formato binario Linux.

Como resumen final de la historia del emulador, y como información adicional, decir que poseo los siguientes ordenadores, como buen amante de la retroinformática:

- Amstrad CPC 464
- Apple iMac G3 333 Mhz
- Atari 1040 STE
- Atari 2600
- Cambridge Z88
- Inves Spectrum +
- MSX Philips VG-8020
- Palm Zire
- Palm Zire 22
- Palm Treo 650
- Sinclair QL
- ZX Spectrum 48k
- 2 ZX Spectrum +
- ZX Spectrum +2A
- ZX-81

2 Agradecimientos

Quisiera agradecer la colaboración y ayuda de las siguientes personas:

- A mis padres y hermanos. Recuerdo con mucho cariño los consejos y críticas de mi padre a mis primeros programas. Descansa en paz
- Ivan Daunis, por el PCGPE y el INTERVUE
- Rubén Parra, por el Spectrum +2A
- A mis amigos de México
- A Amstrad, por permitir a todos los que hacemos emuladores que usemos las ROMS
- Samir Ribic, por ayudarme a subir el emulador, y por toda la información presente en su emulador Warajevo.
- Martijn van der Heide, por instalar el emulador en la FTP
- Phillip Kendall, por decirme un fallo con la instrucción BIT
- Pedro Gimeno, por los tiempos de varias instrucciones.
- Rodolfo Edison Guerra, por incluir mi emulador en su SpecBase.
- A todos aquellos que hacen demos para el Spectrum, pues son un buen campo de pruebas para mi emulador.
- Y sobretodo a este genio que es Clive Sinclair.

3 Contactar con el autor

Si quieres darme sugerencias para el emulador o decirme los fallos que tenga, puedes contactar conmigo a través del siguiente e-mail:
chernandezba@hotmail.com

También he hecho otras utilidades relacionadas con el Spectrum:

- Conversor de cintas de casete Spectrum a ficheros .TAP: similar al SMPATAP, pero leyendo directamente desde cinta de casete y funcionando en LINUX.
- Conversor de cintas de casete de ZX-81 a ficheros .P, ejecutables con el emulador de ZX81 XTENDER de Carlo Delhez.

4 El Emulador

4.1 Ficheros distribuidos

* Ficheros necesarios para emular el Spectrum:

ZXSPECTR.COM El emulador en sí

48.ROM	Contiene la ROM del Spectrum 16k/48k
INVES.ROM	ROM del Inves Spectrum+
128.ROM	ROM del Spectrum 128k
128S.ROM	ROM del Spectrum 128k (Español)
P2.ROM	ROM del Spectrum Plus 2
P2S.ROM	ROM del Spectrum Plus 2 (Español)
P2F.ROM	ROM del Spectrum Plus 2 (Francés)
P2A40.ROM	ROM del Spectrum Plus 2A (Versión 4.0)
P2A41.ROM	ROM del Spectrum Plus 2A (Versión 4.1)
P2AS.ROM	ROM del Spectrum Plus 2A (Español)

ZXSPECTR.SCR Pantalla inicial

KEYBOARD.SCR Pantalla con la distribución del teclado. Obtenida del emulador fuse

* Documentación:

ZXSP_ESP.PDF	El fichero que estas leyendo
ZXSPECTR.PDF	El mismo fichero pero en inglés
CAMBIOS.TXT	Mejoras y bugs de todas las versiones, en español
CHANGES.TXT	Mejoras y bugs de todas las versiones, en inglés

* Utilidades:

Estas utilidades, excepto la LINEASMP, tienen versiones MSDOS y Linux 32 y 64 bits. Versiones MSDOS tienen extensión EXE, versiones Linux 32 bits tienen sufijo _x32 y versiones Linux 64 bits tienen sufijo _x64

LINEASMP.EXE	Utilidad para leer el sonido desde casete. Sólo versión MSDOS
SMPATAP	Utilidad para convertir el sonido a ficheros .TAP
TAPABIN	Utilidad para convertir el contenido de los ficheros .TAP a ficheros .BIN
SP_Z80	Utilidad para convertir entre los formatos SP y Z80 (sólo de 48k)
VGA2RAW	Utilidad para convertir el archivo de vídeo en formato VGA a formato BGR 24 bit.

* Programas y juegos:

SPED52.TAP propio formato (SPED)	Ensamblador/desensamblador para 128k. Incluye código fuente en su
CONVERSO.TAP Incluye código fuente	Conversor de códigos fuentes de formatos GENS, TED y SPED.
REALDEBU.TAP	Desensambladores para 48k y 128k, con código fuente
CURSORDR.TAP	Programa de dibujo. Incluye código fuente
ARTILLER.TAP	Mi primer juego realizado para ZX Spectrum
ROCMAN.TAP	Juego ROCMAN de Xavi Martín Puché.
TOI.TAP	Las 4 partes (y las intros) del TOI ACID GAME
SIRFRED.TAP	Juego Sir Fred de Made In Spain
RICK.TAP	El Rick Dangerous
CANCIONE.TAP	Programa en BASIC con canciones en formato PLAY
CD.TAP programas de Dominio Público para Spectrum	Programa que simula un CD, con varias canciones. Es de los primeros
HISTERIA.TAP	Juego Histeria
BUBBLE.TAP	Juego BUBBLE BOBBLE
BINARY_L.TAP, NOUMENON.TAP, THECUBE.TAP, POWER_UP.TAP	4 Fantásticas demos para 128k
TOICLAVE.ZX	Códigos para el TOI ACID GAME
ABADIA.ZX	Juego Abadía del Crimen (128k)
XENO.ZX	Juego XENO

4.2 Requisitos para usar el emulador

Los requisitos mínimos para ejecutar el emulador son:

procesador Intel 8088, sistema MS-DOS, 462 Kb libres en RAM (en su versión máxima) + el tamaño del propio ejecutable ZXSPECTR.COM, y tarjeta de vídeo VGA. Opcionalmente, para una emulación rápida de los 128k del Spectrum se debe tener memoria EMS 3.0 o superior, una tarjeta Sound Blaster Pro o compatible para poder emular el chip de sonido y cargar desde casete. Sin tarjeta Sound Blaster, se necesitan 32 Kb menos. Y si se dispone de memoria EMS, se necesitan 256 Kb menos

El emulador puede ejecutarse con una CPU 8088, pues no se usan instrucciones del 386, pero para que funcione con una velocidad comparable a la del Spectrum se necesita un procesador a 133MHz como mínimo.

Se debe ejecutar preferiblemente en sesión de MS-DOS, no en ventana de Windows; en éste sistema operativo, la sincronización no es muy correcta.

He probado incluso el emulador dentro de emuladores de MS-DOS para Linux, como el dosemu y el dosbox. Para dosbox, modifiqué las siguientes opciones de configuración del archivo dosbox.conf para un mejor rendimiento:

```
[dosbox]
machine=vgaonly
[cpu]
core=dynamic
```

```
cputype=386  
cycles=max  
[mixer]  
rate=16000
```

4.3 Línea de comandos de ZXSPECTR.COM

Al cargar el emulador, se pueden especificar parámetros opcionales y un nombre de fichero (.SP o .ZX) para cargar. Los parámetros son:

```
/?      Muestra la pantalla de ayuda  
/Red     Forzar modo de pantalla con tono Rojo  
/Green   Forzar modo de pantalla con tono Verde  
/Blue    Forzar modo de pantalla con tono Azul
```

Estas tres opciones se pueden combinar para ejecutar el emulador en distintos tonos, por ejemplo /Green/Red para ejecutarlo en amarillo, o las tres a la vez para ejecutar en tramas de grises.

```
/Nosb    No utilizar tarjeta Sound Blaster  
/No386   No detectar procesador 386  
/Noems   No usar la memoria expandida  
/Eng     Ver los textos del menú en Inglés  
/Esp     Ver los textos del menú en Español
```

```
/16k     Emular Spectrum 16k  
/48k     Emular Spectrum 48k  
/Inves   Emular Inves Spectrum+  
/128k    Emular Spectrum 128k  
/128ks   Emular Spectrum 128k (Español)  
/P2      Emular Spectrum Plus 2  
/P2F     Emular Spectrum Plus 2 (Francés)  
/P2S     Emular Spectrum Plus 2 (Español)  
/P2A40   Emular Spectrum Plus 2A (ROM v4.0)  
/P2A41   Emular Spectrum Plus 2A (ROM v4.1)  
/P2AS    Emular Spectrum Plus 2A (Español)
```

4.4 Emulación del Spectrum

Los modelos que se emulan del Spectrum son los siguientes:

(Grupo 48k)

Sinclair 16k (Emulación parcial, solo al escribir)

Sinclair 48k

Inves Spectrum+

(Grupo 128k) - Los 5 son prácticamente iguales, lo único que cambia es la ROM y la apariencia

externa:

Sinclair 128k
Sinclair 128k Español
Amstrad +2
Amstrad +2 - ROM en Francés
Amstrad +2 - ROM en Español

(Grupo +2A)
Amstrad +2A (ROM v4.0)
Amstrad +2A (ROM v4.1)
Amstrad +2A - ROM en Español

4.4.1 La CPU Z80

La emulación de la CPU está sincronizada, pese a que no emula la memoria compartida. Sin embargo, se puede activar una pseudo-emulación de la memoria compartida, que queda lejos de ser real pero que en algunos casos es efectiva: lo que hace es que si una instrucción se ejecuta en la memoria compartida (16384-32767 en 48k y RAMS 4,5,6,7 en 128k) o se hace un IN o OUT a un puerto par, ésta instrucción durará un 15% más de su tiempo real. Además, en el menú hay una opción para especificar la velocidad relativa de la emulación.

La sincronización en el emulador se puede realizar teniendo en cuenta dos factores: el Timer, o contador interno de la CPU, o utilizando la SoundBlaster. Así, tendremos las 3 combinaciones siguientes:

- Utilizando el Timer: En este caso, se produce una interrupción 50 veces por segundo. Cuando se acaba de dibujar un frame completo en el Spectrum, se comprueba si se ha producido esa interrupción. Sino, se espera a que llegue. Es el método más aproximado a la velocidad del Spectrum, pero provoca que el sonido se desincronice con la imagen; teniendo en cuenta que en este caso el buffer de sonido se reasigna mediante el Timer, y la SoundBlaster no sincroniza exactamente igual, al cabo de unos segundos podremos escuchar parte de sonido repetida dos veces, y parte de otro fragmento que no escucharemos.
- Utilizando la Sound Blaster: Si la tarjeta de sonido está disponible, se produce una interrupción cada vez que la tarjeta avisa de que se ha vaciado el buffer de sonido. En el emulador, el buffer tiene el tamaño de 10 frames de vídeo (0.2 segundos). Cuando se han dibujado 10 frames, se espera a que llegue esa interrupción.
- Utilizando el Timer y la Sound Blaster simultáneamente: Con este método, cada frame de pantalla se sincroniza con el Timer. Cada 10 frames, el último se sincroniza mediante la Sound Blaster. Con este método, el sonido se puede oír perfectamente sincronizado.

Cada método es recomendable en los siguientes casos:

- Si no queremos escuchar sonido, utilizar sólo el Timer
- Si se quiere escuchar sonido:
 - En entorno MS-DOS, Windows 95, Windows 98: Se tiene un entorno real de MS-DOS, para la mayoría de tarjetas de sonido es recomendable utilizar Timer+SoundBlaster
 - En entornos Windows 2000, XP: El método SoundBlaster puede ser el más adecuado, la música no desincroniza, pero provoca “saltos” en pantalla (5 por segundo), o sea, si

hay un objeto en movimiento, veremos como se acelera y frena 5 veces por segundo. En el caso que no se consiga escuchar bien la música con este método, utilizar el Timer o Timer y Sound Blaster.

El método Timer siempre está disponible; los otros dos métodos, están disponibles siempre que se detecte la Sound Blaster (y no se use el switch /nosb), independientemente de que en el emulador deshabilitemos la salida hacia la tarjeta, o el chip AY no esté presente (modos de 48k...).

El emulador soporta la emulación de instrucciones no documentadas, como la de los registros HX,LX,HY,LY, las instrucciones SLL, los 8 RETS con prefijo ED:RETN(ED45), RETI(ED4D), RET3(ED55), RET4(ED5D), RET5(ED65), RET6(ED6D),RET7(ED75), RET8(ED7D), y los NEG y IM repetidos. También el OUT (C),F (ED71) y el IN F,(C) (ED70).

También las instrucciones de manipulación de bits del tipo (XY+d) (DD o FD + CB) con 1 argumento más, es decir, se realiza la instrucción normal y el resultado se almacena en (XY+d) y en el registro indicado, por ejemplo: SET 3,(IX+0),C ->Se pone el bit 3 de (IX+0) a 1 y el valor de (IX+0) se guarda en C.

También soporta el registro R, que es incrementado en 1 después de cada instrucción, entendiendo también los prefijos como instrucciones, exceptuando las instrucciones con prefijo DDCB y FDCB, que incrementan en 2. El registro R es de 8 bits, aunque al incrementarlo sólo se usan los 7 inferiores, y el bit 8 permanece constante (pudiendo modificarlo con la instrucción LD R,A).

Los bits 3 y 5 del registro de flags sólo se emulan parcialmente, no actuando exactamente como en un Spectrum real; se mantienen constantes después de ejecutar un POP AF o un EX AF,AF'. Los bits de signo y de paridad tienen su valor correcto después de una instrucción BIT.

Emula también los modos de interrupciones de Spectrum: IM0 y IM1, que saltan a la dirección 38H (56), y el modo IM2. Se ejecutan con una frecuencia de 50Hz. También se puede provocar una interrupción NMI, saltando a la dirección 66H (102).

Debo destacar que la emulación del Z80 todavía incluye bugs, que no he podido determinar cuales son.

4.4.2 El Teclado

El teclado que emula es el del Spectrum 48k, o sea, de 40 teclas. Las teclas ALT y CTRL actúan las dos como el Symbol Shift. Las teclas de cursor, junto con la tecla INS pueden emular el Joystick Kempston o los cursores del spectrum (SHIFT + 5/6/7/8), y la tecla ESC es para ir al menú del emulador. También emula las siguientes teclas extendidas: Backspace, Caps Lock y TAB (Extended Mode del Spectrum).

Emula los dos tipos de teclado Issue 1 y Issue 2 (bit 6 del puerto de teclado alzado o no).

4.4.3 El Joystick Kempston

El joystick kempston es emulado a través de las teclas de cursor y la tecla Ins como disparo

(también valen los cursores y el Ins del teclado numérico). Cualquier puerto con el bit 5=0 representa al joystick kempston.

También emula un disparador automático, de velocidad variable, que se encontraba presente en aquellos viejos joysticks del Spectrum.

4.4.4 El Sonido

El sonido del Spectrum 48k se genera por el puerto 254, en los bits 3 y 4. El bit 3 sólo se usa al grabar desde el BASIC, y el bit 4 es el que se usa para hacer sonido normalmente. El bit 3 hay pocos emuladores que lo usen, pero en éste sí que se emula, y es debido a la emulación del Inves Spectrum+ (para ver detalles del Inves, leer la sección 4.4.10).

En el Spectrum, cualquier valor a los bits 3 y 4 que sean diferentes al valor último enviado producirá sonido, y en el ZXSpectr también se hace así. El sonido es enviado a la Sound Blaster.

El emulador soporta el chip de sonido del Spectrum 128k (AY-3-8912), usando la tarjeta Sound Blaster. También se emula el sonido estéreo llamado ACB/ABC; esto quiere decir que de los tres canales de sonido, hay uno para cada altavoz y el tercero va al altavoz central (los dos altavoces a la vez). En el modo ACB, el canal A va al altavoz izquierdo, el B va al derecho y el C al central. En el modo ABC, el A va al izquierdo, el C al derecho y el B al central. Sólo he encontrado algunas demos que hagan uso del sonido estéreo.

El chip de sonido se controla por dos puertos, los cuales tienen el bit 15 del puerto a 1 y el bit 1 a cero (A15=1, A1=0). Si el bit 14 está a 1 (típicamente el puerto 65533) es el puerto de selección del registro (salida) o muestra el valor del registro (entrada), y si el bit 14 está a 0 (típicamente el puerto 49149) es el puerto de envío del valor al registro (sólo salida). Digo que típicamente son el 65533 y el 49149 porque es como se menciona en los manuales, pero he encontrado muchos juegos y demos que usan otros puertos.

Los registros del chip son:

- R0 Ajuste fino del tono, canal A
- R1 Ajuste aproximado del tono, canal A
- R2 Ajuste fino del tono, canal B
- R3 Ajuste aproximado del tono, canal B
- R4 Ajuste fino del tono, canal C
- R5 Ajuste aproximado del tono, canal C

El tono de cada canal es un valor de 12 bits que se forma combinando los bits D3-D0 del registro de ajuste aproximado y los bits D7-D0 del registro de ajuste fino. La unidad básica del tono es la frecuencia de reloj dividida por 16 (es decir, 110.83 KHz).

Como el contador es de 12 bits, se puede generar frecuencias de 27 Hz a 110 KHz.

R6 Control del generador de ruido, D4-D0

El periodo del generador de ruido se toma contando los cinco bits inferiores del registro de ruido cada periodo del reloj de sonido dividido por 16. Para la generación de ruido, se tiene en cuenta el periodo, al final de cada lapso de tiempo, se envía aleatoriamente un 0 o un 1; para generar este valor, se utiliza la misma función que el comando RND del Spectrum: $n = (75 * (n+1) - 1) / 65536$

R7 Control del mezclador y de E/S
 D7 No utilizado
 D6 1=puerta de entrada, 0=puerta de salida
 D5 Ruido en el canal C
 D4 Ruido en el canal B
 D3 Ruido en el canal A
 D2 Tono en el canal C
 D1 Tono en el canal B
 D0 Tono en el canal A

Este registro controla la mezcla de ruido y tono para cada canal y la dirección de la puerta de E/S de ocho bits. Un cero en un bit de mezcla indica que la función está activada.

R8 Control de amplitud del canal A
 R9 Control de amplitud del canal B
 RA Control de amplitud del canal C
 D4 1=utilizar generador de envolvente
 0=utilizar el valor de D3-D0 como amplitud
 D3-D0 Amplitud

Estos tres registros controlan la amplitud de cada canal y si éste debe ser modulado o no por los registros de envolvente. En el emulador no se generan envolventes, el volumen es constante a un nivel medio.

RB Ajuste aproximado del periodo de envolvente
 RC Ajuste fino del periodo de envolvente

Los valores de ocho bits de RB y RC se combinan para producir un número de 16 bits que se cuenta en unidades de 256 por el periodo del reloj de sonido. Las frecuencias de envolvente pueden estar entre 0.1 Hz 6 KHz. Estos dos registros no están emulados.

RD Control de envolventes (no emulado)
 D3 Continua
 D2 Ataque
 D1 Alternada
 D0 Sostenida

RE Controla el RS-232, el KeyPad, y el MIDI (no emulado)
 RF No se usa

4.4.5 La Pantalla

Para la pantalla se dispone de los 16 colores del ZX Spectrum y del parpadeo real (intercambio de los valores de PAPER y INK cada 16 frames). Se puede controlar el brillo de los colores.

La actualización de la pantalla se realiza cuando se completa un frame completo de vídeo (50 veces por segundo). Tiene autoframeskip, es decir, si se tarda más de 1/50 de segundo entre cada frame, se descarta y no se dibuja en pantalla, “ahorrando” tiempo para poder resincronizar. Si en el último segundo no se tiene tiempo ni de dibujar un sólo frame, se fuerza una actualización.

El emulador posee un salvapantallas, que no quiero describir cómo es; prefiero que lo veas por ti mismo. Se activa cuando no se pulsa ninguna tecla durante 2 minutos.

4.4.6 128k de memoria

Los 128k se pueden emular con memoria expandida (EMS) o con memoria RAM. Con EMS la paginación se realiza de manera instantánea; en cambio, con memoria RAM puede ser bastante lento, aunque es un poco más rápido si el procesador es 386 o superior. El emulador autodetecta si el procesador es 386 o superior; esto lo he probado en un XT y en Pentium, pero no en un 286. Si se bloquea al ejecutar el emulador, y tienes un 286 o 186, prueba con el parámetro /no386.

La memoria libre que se requiere para cargar el emulador es:

- El tamaño del propio ejecutable ZXSPECTR.COM
- Más 188 Kb libres en RAM
- Otros 45 Kb libres en RAM si se utiliza tarjeta SoundBlaster
- Otros 256 Kb libres en RAM si no se dispone de memoria expandida (EMS)

La paginación del 128k se realiza mediante el puerto 32765 (realmente es cualquier puerto con A1=0 y A15=0). La descripción del puerto 32765 es:

D0 a D2	Selección de RAM
D3	Selección de pantalla (pantalla en RAM 5 o RAM 7)
D4	Selección de ROM
D5	Inhabilitación de la paginación

Debido a un error interno en el 128k (pero no en un +2A), si se lee desde ese puerto, lo que en realidad pasa es que se hace un OUT a ese puerto con el valor 255, por lo que si estamos en 128 BASIC, se reseteará el ordenador. Esta característica también se emula.

La paginación del +2A se realiza mediante dos puertos: el 32765 (pero solamente el 32765) y el 8189.

La descripción del puerto 8189 es:

D0	Decide si D1 y D2 afectan a la ROM o a la RAM ¹
D1 y D2	Conmutación de ROM/RAM ²
D4	Motor del disco (no emulado)
D5	Señal STROBE en la puerta paralelo (activa a nivel bajo) (no emulado)

Cuando el bit 0 del puerto 8189 está a 0, la ROM se selecciona mediante el bit 4 de 32765 (bit bajo de la ROM) y el bit 2 de 8189 (bit alto de la ROM):

¹ En el manual se menciona erróneamente como bit D3

² En el manual se mencionan erróneamente como bits D0 y D1

Bit 2 de 8189	Bit 4 de 32765	ROM que entra
0	0	0
0	1	1
1	0	2
1	1	3

Cuando el bit 0 del puerto 8189 está a 1, los bits 1 y 2 controlan qué combinación de páginas de RAM ocupan los 64K posibles:

Bit 2 de 8189	Bit 1 de 8189	Páginas de RAM
0	0	0,1,2,3
0	1	4,5,6,7
1	0	4,5,6,3
1	1	4,7,6,3

4.4.7 Emulación de Cinta

La emulación de la cinta desde dentro del emulador se realiza mediante ficheros .TAP. Estos ficheros fueron creados en el emulador Z80 de Gerton Lunter. Los ficheros .TAP contienen una serie de bloques en el que cada uno identifica los datos grabados de cinta. Cada bloque tiene el formato:

- WORD que indica la longitud de los datos que vienen a continuación (incluyendo flag y checksum).
- BYTE que indica el flag del bloque grabado (como en una cinta real: 0 para cabeceras y 255 para datos).
- DATOS: Datos grabados.
- BYTE que indica el checksum de los datos (incluyendo el flag). Resulta de hacer un XOR de todos los datos.

Ejemplo:

SAVE "ROM" CODE 0,2

Fichero .TAP generado:

		Datos generados por el Spectrum																		
13	00	00	03	52	4f	4d	7x20	02	00	00	00	00	80	f1	04	00	ff	f3	af	a3

Datos generados por el Spectrum		
	Longitud: 19 bytes (17 bytes+flag+checksum)	
	Flag	
	Primer byte de la cabecera, indicando BYTES (3)	
nombre		
información de cabecera		
checksum de cabecera (incluyendo flag)		
longitud segundo bloque		
flag		
primeros 2 bytes de la ROM		
checksum		

Estos ficheros son usados desde dentro del emulador de manera que cualquier operación LOAD o SAVE es efectuada mediante estos ficheros. Se deben especificar los ficheros .TAP a usar desde el menú.

Las rutinas de LOAD y SAVE empiezan en la ROM en las direcciones 1366 y 1218 respectivamente. Si se ejecutasen esas rutinas de la ROM tal cual, lo único que obtendríamos desde el emulador seria una serie de franjas en el BORDER al grabar y al cargar. Para poder usar los ficheros .TAP, el emulador sabe cuando se accede a una de las dos rutinas de la ROM, y se redirigen las operaciones hacia funciones propias de emulación de cintas .TAP. Se permite la carga de datos mas cortos de lo especificado en el fichero .TAP, tomando como byte de checksum el siguiente a leer.

La verificación de un bloque no se permite (el emulador carga el bloque independientemente de si se especifica verificar o no).

El flag Z' es puesto a 0 en la rutina de la ROM, pero se puede entrar alternativamente en la dirección 1378 poniendo el flag Z' a 1, de manera que "engaña" al Spectrum y le dice que no ha de distinguir el flag, cargándolo como un byte cualquiera. Esto se comporta así en un Spectrum, y también en el emulador. Se usa en programas copiadores y en algunos juegos; sin embargo, he encontrado fallos con algunos juegos, como el ROCMAN, de manera que se puede desactivar desde el Menú la opción de poder cargar cualquier flag (flag Z' siempre irá a 0). Este mismo juego lo he intentado cargar desde el emulador Z80 (3.05) y se borra; en el Warajevo (2.51) no se borra, pues no permite la carga de cualquier flag (en modo de carga fast).

4.4.8 Carga directa desde cinta de casete

El emulador posee en el menú una opción para poder leer un programa desde cinta de casete y convertirlo a formato .TAP. Para poder realizar esto se debe tener una tarjeta Sound Blaster Pro compatible o superior. Además, hace falta un casete (naturalmente!) y un cable mono que se conecta al casete y a la entrada Line In de la Sound Blaster. Esta opción no se puede realizar desde dentro de Windows, se debe realizar en sesión MS-DOS. También es recomendable no tener en memoria programas de caché de disco, como el SMARTDRV.

En el emulador se permite cambiar la velocidad de lectura de los datos, para poder leer datos turbo (a más de 1500 baudios); sin embargo, hay tarjetas compatibles SB que no lo permiten. Si este es el caso, se pueden cambiar las constantes de la carga para poder leer turbo. Estas constantes están explicadas en el apartado 5. Opciones del menú.

4.4.9 Ficheros Snapshot

Las primeras versiones del emulador trataban exclusivamente ficheros en formato SP, del emulador SPECTRUM de Pedro Gimeno. Más adelante, me inventé un formato llamado ZX, que está derivado del formato SP pero que incluye muchas mejoras (la cabecera es del formato SP más 256 bytes adicionales). Sé que existe un emulador que tiene un formato de ficheros llamado también ZX, pero no tiene nada que ver con el mío.

La versión actual del emulador permite la carga de ficheros SP o ZX, pero la grabación se hace exclusivamente en formato ZX. La descripción del formato SP está sacada literalmente del fichero SPECTRUM.DOC del emulador de Pedro Gimeno:

Ficheros *.SP:

Offset	Longitud	Descripción
0	2 bytes	"SP" (53h, 50h) Signatura
2	1 palabra	Longitud del programa en bytes (el emulador actualmente sólo genera programas de 49152 bytes)
4	1 palabra	Posición inicial del programa (el emulador actualmente sólo genera programas que comiencen en la pos. 16384)
6	1 palabra	Registro BC del Z80
8	1 palabra	Registro DE del Z80
10	1 palabra	Registro HL del Z80
12	1 palabra	Registro AF del Z80
14	1 palabra	Registro IX del Z80
16	1 palabra	Registro IY del Z80
18	1 palabra	Registro BC' del Z80
20	1 palabra	Registro DE' del Z80
22	1 palabra	Registro HL' del Z80
24	1 palabra	Registro AF' del Z80
26	1 byte	Registro R (de refresco) del Z80
27	1 byte	Registro I (de interrupciones) del Z80
28	1 palabra	Registro SP del Z80
30	1 palabra	Registro PC del Z80

Offset	Longitud	Descripción																		
32	1 palabra	Reservada para uso futuro, siempre 0																		
34	1 byte	Color del borde al comenzar																		
35	1 byte	Reservado para uso futuro, siempre 0																		
36	1 palabra	Palabra de estado codificada por bits. Formato: <table><tr><th>Bit</th><th>Descripción</th></tr><tr><td>15-8</td><td>Reservados para uso futuro</td></tr><tr><td>7-6</td><td>Reservados para uso interno, siempre 0</td></tr><tr><td>5</td><td>Estado del Flash: 0 - tinta INK, papel PAPER 1 - tinta PAPER, papel INK</td></tr><tr><td>4</td><td>Interrupción pendiente de ejecutarse</td></tr><tr><td>3</td><td>Reservado para uso futuro</td></tr><tr><td>2</td><td>Biestable IFF2 (uso interno)</td></tr><tr><td>1</td><td>Modo de interrupción: 0=IM1; 1=IM2</td></tr><tr><td>0</td><td>Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI) 1 - Interrupciones activadas (EI)</td></tr></table>	Bit	Descripción	15-8	Reservados para uso futuro	7-6	Reservados para uso interno, siempre 0	5	Estado del Flash: 0 - tinta INK, papel PAPER 1 - tinta PAPER, papel INK	4	Interrupción pendiente de ejecutarse	3	Reservado para uso futuro	2	Biestable IFF2 (uso interno)	1	Modo de interrupción: 0=IM1; 1=IM2	0	Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI) 1 - Interrupciones activadas (EI)
Bit	Descripción																			
15-8	Reservados para uso futuro																			
7-6	Reservados para uso interno, siempre 0																			
5	Estado del Flash: 0 - tinta INK, papel PAPER 1 - tinta PAPER, papel INK																			
4	Interrupción pendiente de ejecutarse																			
3	Reservado para uso futuro																			
2	Biestable IFF2 (uso interno)																			
1	Modo de interrupción: 0=IM1; 1=IM2																			
0	Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI) 1 - Interrupciones activadas (EI)																			

El formato ZX tiene tres versiones, siendo la inicial la versión 1 y la actual es la versión 3. Los valores usados a partir de la versión 2 están indicados con (v. 2+). El único valor que incluye de más la versión 3 es el del ordenador emulado. La descripción del formato ZX es:

Campo	Descripción
signatura	"ZX"
long_prog	WORD=49152
pos_inicial	WORD=16384
reg_c	BYTE
reg_b	BYTE
reg_e	BYTE
reg_d	BYTE
reg_l	BYTE
reg_h	BYTE
reg_f	BYTE
reg_a	BYTE

Campo	Descripción																
reg_ixl	BYTE																
reg_ixh	BYTE																
reg_iyl	BYTE																
reg_iyh	BYTE																
Registros '																	
reg_c_	BYTE																
reg_b_	BYTE																
reg_e_	BYTE																
reg_d_	BYTE																
reg_l_	BYTE																
reg_h_	BYTE																
reg_f_	BYTE																
reg_a_	BYTE																
reg_r	BYTE																
reg_i	BYTE																
reg_sp	WORD																
reg_pc	WORD																
reservado1	WORD=0																
border	BYTE																
reservado2	BYTE=0 Antes de la versión 2 de la cabecera, esto era un word y reservado20 no existía.																
bits_estado	<table><tr><th>Bit</th><th>Descripción</th></tr><tr><td>7-6</td><td>Reservados para uso interno, siempre 0</td></tr><tr><td>5</td><td>Estado del Flash: 0 - tinta INK, papel PAPER. 1 - tinta PAPER, papel INK</td></tr><tr><td>4</td><td>Interrupción pendiente de ejecutarse</td></tr><tr><td>3</td><td>Reservado para uso futuro</td></tr><tr><td>2</td><td>Biestable IFF2 (uso interno)</td></tr><tr><td>1</td><td>Modo de interrupción: 0=IM1, 1=IM2</td></tr><tr><td>0</td><td>Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI) 1 - Interrupciones activadas (EI) En este emulador sólo se usan el bit 5(estado del flash), el bit 1 (si se está en modo IM0 o IM1=0, modo IM2=1), y el bit 0 (DI o EI)</td></tr></table>	Bit	Descripción	7-6	Reservados para uso interno, siempre 0	5	Estado del Flash: 0 - tinta INK, papel PAPER. 1 - tinta PAPER, papel INK	4	Interrupción pendiente de ejecutarse	3	Reservado para uso futuro	2	Biestable IFF2 (uso interno)	1	Modo de interrupción: 0=IM1, 1=IM2	0	Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI) 1 - Interrupciones activadas (EI) En este emulador sólo se usan el bit 5(estado del flash), el bit 1 (si se está en modo IM0 o IM1=0, modo IM2=1), y el bit 0 (DI o EI)
	Bit	Descripción															
	7-6	Reservados para uso interno, siempre 0															
	5	Estado del Flash: 0 - tinta INK, papel PAPER. 1 - tinta PAPER, papel INK															
	4	Interrupción pendiente de ejecutarse															
	3	Reservado para uso futuro															
	2	Biestable IFF2 (uso interno)															
	1	Modo de interrupción: 0=IM1, 1=IM2															
0	Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI) 1 - Interrupciones activadas (EI) En este emulador sólo se usan el bit 5(estado del flash), el bit 1 (si se está en modo IM0 o IM1=0, modo IM2=1), y el bit 0 (DI o EI)																
reservado20	BYTE=0 (v. 2+)																
Lo que sigue a continuación es la cabecera extendida para los ficheros ZX																	
version	BYTE=1, 2 o 3 Versión de la cabecera ZX																

Campo	Descripción																		
cabecera_velocidad	WORD No se usa desde ZXSpectr V2.0																		
cabecera_dirs_pant	WORD=64 Ya no se usa. Se usaba antiguamente para la versión CGA y la actualización de pantalla. No usar nunca en versiones futuras																		
cabecera_frecuencia	BYTE=1 Ya no se usa. Se usaba antiguamente para la version CGA y la actualización de pantalla. No usar nunca en versiones futuras																		
control_brillo	BYTE Valor del brillo																		
disparador_defecto	BYTE Frecuencia del disparador automático. Se calcula: frecuencia = 50/disparador_defecto																		
sonido	BYTE A 1 si está activado el sonido (puerto 254)																		
bits_estado0	<p>BYTE El significado es el siguiente:</p> <table> <tr> <th>Bit</th><th>Descripción</th></tr> <tr> <td>7</td><td>Contiene el bit 6 del puerto del teclado (Issue 1 o 2)</td></tr> <tr> <td>6</td><td>Indica si está activado el disparador automático</td></tr> <tr> <td>5</td><td>A 0 indica que el refresco de Flash está activo, sino a 1. Obsoleto</td></tr> <tr> <td>4</td><td>A 1 indica que el programa a cargar es de 128k (versión 2+)</td></tr> <tr> <td>3</td><td>A 1 indica que no es posible cambiar el color del borde mediante OUT 211, valor (versión 2+)</td></tr> <tr> <td>2</td><td>No usado</td></tr> <tr> <td>1</td><td>A 1 indica que está habilitado el protector de pantallas</td></tr> <tr> <td>0</td><td>Paleta CGA (versión 2) 0=Negro, Verde, Rojo, Amarillo 1=Cyan, Magenta, Blanco</td></tr> </table>	Bit	Descripción	7	Contiene el bit 6 del puerto del teclado (Issue 1 o 2)	6	Indica si está activado el disparador automático	5	A 0 indica que el refresco de Flash está activo, sino a 1. Obsoleto	4	A 1 indica que el programa a cargar es de 128k (versión 2+)	3	A 1 indica que no es posible cambiar el color del borde mediante OUT 211, valor (versión 2+)	2	No usado	1	A 1 indica que está habilitado el protector de pantallas	0	Paleta CGA (versión 2) 0=Negro, Verde, Rojo, Amarillo 1=Cyan, Magenta, Blanco
Bit	Descripción																		
7	Contiene el bit 6 del puerto del teclado (Issue 1 o 2)																		
6	Indica si está activado el disparador automático																		
5	A 0 indica que el refresco de Flash está activo, sino a 1. Obsoleto																		
4	A 1 indica que el programa a cargar es de 128k (versión 2+)																		
3	A 1 indica que no es posible cambiar el color del borde mediante OUT 211, valor (versión 2+)																		
2	No usado																		
1	A 1 indica que está habilitado el protector de pantallas																		
0	Paleta CGA (versión 2) 0=Negro, Verde, Rojo, Amarillo 1=Cyan, Magenta, Blanco																		
puerto_32765	BYTE Valor del ultimo OUT al puerto 32765 (v. 2+)																		
puerto_8189	BYTE Valor del ultimo OUT al puerto 8189 (v. 2+)																		
paginas_actuales	4 BYTES Contiene para cada segmento de memoria (0000-3ffffh, 4000h-7ffffh, 8000h-bffffh, c000h-ffffh) la página (ROM o RAM) asignada, siendo de 0 a 3 ROMS, y de 4 a 11 RAMS (v. 2+). Si se emula un ordenador de 128k (pero no un +2A), la ROM1 se identifica como ROM3.																		
puerto_65533	BYTE Valor del ultimo OUT al puerto 65533 (v. 2+)																		
ay_3_8912_registros	16 BYTES Contenido de los registros del chip de sonido (v. 2+). Se implementó a partir de una versión avanzada de la version 2. Si se lee una versión anterior, se leerá 255, pues estaba reservado																		

Campo	Descripción
ordenador_emulado	BYTE Ordenador que se emula: (v. 3+) 0=Sinclair 16k 1=Sinclair 48k 2=Inves Spectrum+ 3=Sinclair 128k 4=Amstrad +2 5=Amstrad +2 - Frances 6=Amstrad +2 - Español 7=Amstrad +2A (ROM v4.0) 8=Amstrad +2A (ROM v4.1) 9=Amstrad +2A - Español
reservado4	222 BYTES=255 Reservado para uso futuro
DATOS	X BYTES

A partir de “DATOS” se graban los datos de la memoria en sí. Si el programa es de 48k, se graba desde la dirección 16384 hasta la 65535; si es de 128k, se graba desde la RAM 0 hasta la RAM 7. El programa grabará un fichero de 48k si se ejecuta el emulador en modo /48k, o si se está en modo 48 BASIC; en caso contrario, se grabarán 128k.

Los datos están comprimidos, de la manera siguiente:

- Si un byte se encuentra repetido más de 4 veces, se graba como:
221,221,byte a repetir,numero de veces
- Si justo antes de la repetición hay un código 221, se graba como:
221 de antes de la repetición,byte a repetir,221,221,byte a repetir,numero de veces-1

Cuando se graban 128k, se hace en dos bloques de 64k, el primero se hace desde la RAM 0 hasta la RAM 3, y el segundo es desde la RAM 4 hasta la RAM 7, por lo que no se detectan repeticiones de bytes que acaben en la RAM 3 y sigan en la RAM 4, por ejemplo:

Si tenemos en memoria lo siguiente:

RAM3

16377=10

16378=20

16379=20

16380=20

16381=20

16382=20

16383=20

RAM4

0 =20

1 =20

2 =20

3 =20

4 =20

5 =20

6 =20

7 =4

Al grabarlo, el código generado será el siguiente:
10, 221,221,20,6, 221,221,20,7, 4

4.4.10 Emulación del Inves Spectrum+

El Inves Spectrum+ fue un ordenador creado por Investrónica hacia el año 1988. Presenta algunos inconvenientes respecto a un Spectrum normal:

Posee una ROM similar a la ROM 1 del Spectrum 128k, pese a que sólo tiene 48k de RAM, de manera que tiene las rutinas de paginación de la dirección 14446.

La actualización de pantalla se realiza de manera distinta a la de un Spectrum normal, de manera que cuando se produce una interrupción la ULA comienza a dibujar a la altura de la pantalla (o sea, la dirección 16384) y no en la parte superior de borde. De este modo, en algunos juegos que utilizan ese tiempo del borde superior para borrar objetos, en el Inves se ve un parpadeo. Además. En muchas pantallas de juegos se pueden ver líneas "falsas" cuando coinciden determinados colores (todavía no se que combinación produce este efecto). Las temporizaciones del Inves no las sé seguro, aunque supongo que son las mismas que en un Spectrum.

La propiedad más interesante del Inves Spectrum+ es los efectos producidos al POKEar en la ROM, sí, en la ROM. Existen determinadas direcciones "privilegiadas" de la ROM en las que se puede alterar el funcionamiento normal del Inves; estas direcciones son las que el byte bajo de la dirección es 254 (XXFEh), como el puerto del sonido.

Pokeando en todas estas direcciones de la ROM (254,511,...,16383) con un mismo valor, lo que se crea es una máscara AND que se aplicará en el valor enviado al puerto 254, y ese será el valor real enviado. Debemos asumir que el POKE en la ROM al enchufar el ordenador (y no al hacer un RESET) es 255. De esta manera, se producen dos efectos:

1. El primero es que los tres bits inferiores constituyen una máscara a la hora de cambiar el borde con un OUT 254, es decir, al valor enviado al puerto 254 se le hace un AND con el valor que se POKEa la ROM, y ese será el color del borde: por ejemplo: con un POKE en la ROM con 6, tendremos Borde de color par, es decir, el borde 0 y 1 será el 0, el 2 y 3 será 2, etc. Por otra parte, si POKEamos con 0, todos los bordes que pongamos serán negros (0).
2. El segundo efecto (y el peor en muchos juegos con música) es el sonido. Intentaré explicarlo de manera sencilla, aunque es un poco complicado. A los bits 3 y 4 del valor que se envía al puerto 254 se le hace un AND con el valor del POKE a la ROM. Entonces, se hace un XOR de los bits 3 y 4 resultantes, de manera que obtenemos 1 bit de resultado. Entonces, este bit será el enviado al altavoz, de manera que si se envía un bit igual al anterior, el altavoz no se conmuta y no hay sonido. Hay que recordar que en un Spectrum normal, cualquier valor de los bits 3 y 4 que sean diferentes a los anteriores producen sonido. De esta manera, tenemos las siguientes combinaciones: (S significa que hay sonido, N significa que no)

	Bits 4 y 3 POKE a ROM			
Bits 4 y 3 puerto 254 (primer y segundo valor enviados)	11	10	01	00

	Bits 4 y 3 POKE a ROM			
00 01	S	N	S	N
00 10	S	S	N	N
00 11	N	S	S	N
01 10	N	S	S	N
01 11	S	S	N	N
10 11	S	N	S	N

En un Spectrum normal, cualquier combinación de estos bits 4 y 3 en el puerto 254 producirá sonido; en un Inves (y en el emulador) ésto no es así. En un Inves real, se deben POKEar todas las direcciones de la ROM con el byte bajo de la dirección que valga 254 (XXFEH); si no se POKEan todas con el mismo valor, los efectos producidos serán una combinación de todos (hay una determinada combinación la cual al hacer un BEEP, se producen franjas negras en el borde!). En el emulador, basta con POKEar una sólo de estas direcciones para que se tenga el byte de POKEo (también se puede hacer desde el menú).

Esta característica del sonido en el Inves produce que en muchos juegos no se escuche la música, o sólo un pequeño chasquido en músicas a dos canales (todos los juegos de Code Masters, el Lemmings,...). También, el sonido se alterará si hacemos POKE 23659,0 desde el BASIC. Como se ve en el cuadro, no hay ni un valor para la ROM en que se comporte de manera igual que un Spectrum.

4.4.11 El puerto "bus idle"

Recibe el nombre de "bus idle" cualquier puerto del Spectrum que no tenga dispositivo asignado. Un puerto así, debe devolver el valor que lee la ULA desde la pantalla, es decir, cuando la ULA está dibujando la pantalla lee bytes correspondientes a los pixeles y a los atributos. Este byte leído por la ULA puede saberse haciendo IN en cualquier puerto que no tenga asignado dispositivo (típicamente el 255). Cuando la ULA no lee bytes de pantalla (al dibujar el border o en los sincronismos) se devuelve el valor 255. Esto funciona así en los Spectrum 16k, 48k, y 128k (y +2). En cambio, en un Inves Spectrum+ siempre leeremos un 255.

En un Spectrum +2A, la cosa se complica un poco:

Cuando la paginación está bloqueada (bit 5 del puerto 32765 puesto a 1), siempre leeremos 255; cuando no lo está, leeremos el byte que lee la ULA (con el bit 0 alzado) y siempre que el numero del puerto corresponda a la siguiente máscara: 0000XXXXXXXXX01b, es decir, empezando por el puerto 1 y con incrementos de 4 en 4, hasta el puerto 4093 (1,5,9,13,..., 4093). Esta característica la he descubierto haciendo pruebas con mi +2A, pues en toda la documentación que hay en distintos emuladores y en páginas Web, se dice que el +2A siempre devuelve 255. Posiblemente el fallo esté en que se lea sólo el puerto número 255, el cual sí devuelve 255.

Esta característica del "bus idle" se usa en algunos juegos, como el Arkanoid o el Renegade. Desde la versión 3.0 del emulador está soportada, funcionando de manera bastante real.

5 Opciones del menú

Para elegir una opción del menú hay usar los cursores y las teclas Enter, + y -. En el caso de opciones con valores numéricos, se debe pulsar + para incrementar el valor, y la tecla - para decrementar el valor.

Hay determinadas opciones que no se pueden seleccionar, en cuyo caso el cursor normal (->) se convertirá en (-x).

En las opciones de selección de archivo se muestra el cursor como (F>). En ese caso podemos seleccionar el fichero pulsando Enter y luego usar los cursores, Re Pag, Av Pag. Si queremos escribir el nombre del fichero, debemos seleccionar la opción con la tecla - o Espacio.

5.1 Cargar Snapshot

Con esta opción podremos cargar un fichero snapshot de tipo ZX o SP; el tipo de fichero lo detecta de su cabecera, no de la extensión, p.ej: es posible tener un fichero con extensión .TXT y que sea de tipo ZX. Al cargar un fichero SP, las opciones de brillo, disparador, etc. se conservan, es decir, serán las que habían antes de cargar el fichero SP.

5.2 Grabar Snapshot

Con esta opción podremos crear un fichero snapshot de tipo ZX; se debe especificar el nombre y su extensión. Si el fichero ya existe, se preguntará si se quiere sobrescribir.

5.3 Selección de Ordenador

Con esta opción podremos elegir el ordenador a emular. Al seleccionarlo, se hace un RESET, se pone la velocidad relativa a 100% y se extraen las cintas.

5.4 Ajustes de Pantalla

5.4.1 Cargar Pantalla

Con esta opción cargaremos un fichero .SCR en la dirección 16384.

5.4.2 Grabar Pantalla

Con esta opción grabaremos el contenido de las direcciones 16384-23295 en un fichero .SCR (se

debe especificar la extensión).

5.4.3 Control de Brillo

Con esta opción podremos aumentar o disminuir el brillo de los colores; los colores del menú no varían.

5.4.4 Protector de Pantalla

Con esta opción podremos activar o desactivar la aparición del protector de pantalla; a veces conviene desactivarlo cuando estamos escuchando la música de un juego.

5.4.5 Iniciar grabación vídeo

Con esta opción se inicia la grabación de la salida generada por pantalla a un archivo de disco. Se genera un archivo de vídeo “en crudo”, sin cabecera ni compresión. Se trata del contenido de la pantalla VGA para cada frame capturado (cada frame ocupa 64k). Dado que el modo VGA usado utiliza paleta de colores, se debe posteriormente usar la utilidad VGA2RAW para convertir el archivo de vídeo del modo paleta a modo color RGB. Exactamente, el modo resultante es: 320x200, BGR24 (24 bit por cada píxel en pantalla).

Se puede reproducir el archivo resultante con el programa Mplayer, por ejemplo:

```
mplayer -demuxer rawvideo -rawvideo fps=2:w=320:h=200:format=bgr24 archivo.rwv
```

Para un archivo guardado a 2 FPS.

Si se quiere reproducir simultáneamente un archivo de audio en crudo (ver sección 5.8.6 Iniciar grabación audio):

```
mplayer -demuxer rawvideo -rawvideo fps=2:w=320:h=200:format=bgr24 archivo.rwv  
-audiofile archivo.rwa -audio-demuxer 20 -rawaudio channels=2:rate=15550:samplesize=1
```

Se debe tener en cuenta que la grabación de vídeo, al igual que la de audio, se genera siempre que se esté en modo multitarea o cuando se sale del menú y se vuelve a la emulación del Spectrum. En el caso que se grabe simultáneamente audio y vídeo, se recomienda empezar con la multitarea desactivada. De esa manera, al volver al emulador, el audio y vídeo empezará simultáneamente y estarán sincronizados. Después de empezar a grabar se puede activar la multitarea. Aunque mientras está la grabación activa el sonido se puede escuchar mal o el vídeo puede ir a menos frames de lo normal, el archivo generado siempre será correcto.

5.4.6 Fichero

Con esta opción se elige el archivo utilizado al grabar vídeo en archivo. Al utilizar el selector de archivos, se prefiere la extensión .VGA.

5.4.7 FPS

En esta opción se indican los frames por segundo generados del archivo de vídeo.

5.5 Menú Depuración

5.5.1 Generar RESET

Con esta opción generaremos un RESET del Spectrum.

5.5.2 Generar NMI

Con esta opción provocaremos una interrupción No Enmascarable (llamar a la dirección 66H).

5.5.3 Ver Registros

Con esta opción podremos ver los registros de la CPU. También se permite activar y desactivar las interrupciones, y también cambiar el modo de interrupciones.

También podremos ver las páginas activas, la pantalla en uso (5 o 7) y si la paginación está activa (en 128k). También se puede ver el valor del puerto 254 (color del borde, bits 4 y 3), y el número de frames de pantalla por segundo (FPS)

5.5.4 Pokear

Se usa para modificar direcciones de memoria, útil para conseguir vidas infinitas en los juegos. Se muestra el valor anterior de la dirección a modificar.

5.6 Emulación de Cinta

5.6.1 Insertar/Extraer fichero de Entrada

Con esta opción indicaremos al emulador que debe usar el fichero especificado (de formato TAP) como cinta de entrada (si se selecciona insertar), o que el fichero ya no se usa como entrada (si se selecciona extraer). Si el fichero no existe, se preguntará si deseamos crearlo.

5.6.2 Fichero

Con esta opción indicamos el fichero que queremos usar como entrada de cinta (de formato TAP). Para que el emulador lo use, debemos también insertarlo.

5.6.3 Insertar/Extraer fichero de Salida

Con esta opción indicaremos al emulador que debe usar el fichero especificado (de formato TAP) como cinta de salida (si se selecciona insertar), o que el fichero ya no se usa como salida (si se selecciona extraer). Si el fichero no existe, se preguntará si deseamos crearlo.

5.6.4 Fichero

Con esta opción indicamos el fichero que queremos usar como salida de cinta (de formato TAP). Para que el emulador lo use, debemos también insertarlo.

5.6.5 Cargar cualquier flag

Aquí indicamos si la rutina de carga permite cargar datos indistintamente de su flag (Z' alzado), o si no se permite (Z' siempre a 0).

5.6.6 Cargar desde casete

Es el menú para cargar programas desde cinta real de Spectrum. Se debe tener una tarjeta compatible Sound Blaster para que la opción esté activa. Además, no se debe ejecutar desde dentro de Windows (que raro!), si se hace no se leerá nada.

5.6.6.1 Baudios

Indica la velocidad a la que se cargan los datos. Se hace cambiando la frecuencia de lectura de la tarjeta de sonido. En mi tarjeta de sonido no ocurre nada si se toca la velocidad, pues siempre lee a 11KHz. También se puede cargar a distintas velocidades cambiando los parámetros de longitud onda ceros y unos.

5.6.6.2 Longitud onda guía

Indica cuánto dura una onda del tono guía; la duración (en segundos) de este valor representa: segundos=longitud onda guía/frecuencia. Normalmente, en las cargas turbo, el tono guía no cambia.

5.6.6.3 Longitud tono guía

Indica cuántas ondas de tono guía se deben encontrar seguidas para que el emulador sepa que eso es el tono guía. Al cargar, cuando se sobrepasa ese valor, en pantalla aparece "Leyendo tono guía...".

5.6.6.4 Longitud onda ceros

Indica cuánto dura la onda de un bit que sea cero, con margen de error de ± 2 . La duración se calcula igual que para el tono guía. Se debe modificar sólo si la tarjeta de sonido no permite cambiar la frecuencia: si hay que cargar un programa turbo a 3000 baudios, el valor de ceros y el de unos deben ser la mitad.

5.6.6.5 Longitud onda unos

Indica cuánto dura la onda de un bit que sea uno, con margen de error de ± 2 . La duración se calcula igual que para el tono guía.

5.6.6.6 Filtro de Entrada

Indica el filtro que aplica la tarjeta de sonido a la lectura de datos. Según he leído, el filtro sólo esta disponible en tarjetas Sound Blaster Pro, y no en las Sound Blaster superiores.

5.6.6.7 Volumen canal izq.

Indica el volumen de entrada del canal izquierdo.

5.6.6.8 Volumen canal der.

Indica el volumen de entrada del canal derecho.

5.6.6.9 Ver Franjas del Borde

Con esta opción puedes ver el borde mientras se carga; no es exactamente como el borde en un Spectrum real.

5.6.6.10 Autocorrección de Checksum

Dice al emulador si el byte final de carga (checksum) se debe ajustar cuando hay error de carga.

5.6.6.11 Comenzar a Cargar

Esta es la opción que yo considero más importante de todo el emulador, que permite cargar programas. Antes que nada, hay que tener un fichero .TAP insertado en salida.

Hay que tener en cuenta que no se puede cargar desde ventana de Windows (se debe hacer en sesión MS-DOS), y es recomendable que no haya en memoria ningún caché de disco (como el SMARTDRV).

Cuando seleccionamos esta opción, aparece en la mitad de la pantalla un indicador del volumen. Este indicador no importa que volumen indique, pues tanto se puede cargar a un volumen alto como a un volumen bajo, simplemente es un aviso de que se lee sonido. Aún así, por los altavoces de la Sound Blaster podremos oír el sonido. También puedes ver las franjas del borde mientras carga; es similar a un Spectrum real.

Para poder cargar correctamente, el sonido se debe oír lo más agudo posible (ajustando el tornillo o "azimut" del casete si hace falta) y a un volumen que no distorsione, al igual que en un Spectrum normal.

Debajo del volumen hay una franja que indica los bytes que se están cargando, incluyendo el flag. Respecto a estos datos, puede ser que la visualización se vaya parando cada 1.5 segundos. Esto pasa cuando se ejecuta el emulador con algún emulador de EMS, como el EMM386. La razón de esta pausa reside en que la lectura directa (DMA) no se hace byte a byte, sino que se lee primero todo el buffer (de 16k) y luego se traslada a la memoria del emulador. No pasa nada, pues la lectura no se detiene, sino que primero pasa por el buffer intermedio. En cambio, si ejecutamos el emulador en modo MS-DOS sin EMS, la visualización es continua.

Cuando se ha leído un bloque, se escribe la cabecera, si es de 17 bytes y tiene flag 0, y si no, se escribe el flag y la longitud (esa longitud muestra también el byte de checksum y el flag). También indica si ha habido error de carga.

Entonces, podemos grabar el bloque pulsando la tecla G. Si ha habido error de carga y no hay autocorrección de checksum, cuando luego ejecutemos LOAD "" dará error de carga desde el BASIC.

5.7 Ajustes de Hardware

5.7.1 Teclado Issue0/1

Permite elegir entre los dos tipos de teclado del Spectrum.

5.7.2 Disparador Automático

Permite activar o desactivar el disparador automático del joystick kempston.

5.7.3 Frec. disparador

Permite cambiar la frecuencia del disparador automático. Esta frecuencia indica cuantas veces por segundo se invierte el bit 4 del puerto kempston.

5.7.4 Cursores emulan

Permite elegir si los cursores del teclado emulan el cursor del spectrum (SHIFT + 5/6/7/8) o el Joystick Kempston. Las versiones previas a la 3.5 emulaban siempre el Kempston.

5.7.5 Valor POKE a ROM

Permite cambiar el valor de POKE a la ROM (sólo en modo Inves).

5.7.6 Memoria compartida

Permite activar la pseudo-emulación de la memoria compartida. Esta se basa en que las instrucciones que se ejecuten en una RAM lenta tardan un 15% más en ejecutarse. En algunos juegos es bastante real, pero recomiendo desactivarla para las demos.

5.7.7 Sincronismo

Permite seleccionar el tipo de sincronización de la CPU: Timer, SoundBlaster o SoundBlaster+Timer

5.8 Ajustes de sonido

5.8.1 Sonido Activo

Indica si se escucha sonido o no.

5.8.2 Chip AY presente

Indica si el chip de sonido existe o no. Puede estar presente el chip pero en cambio no tener tarjeta Sound Blaster; de esta manera los puertos del chip son manipulables pero no se oye nada.

5.8.3 Modo de Stereo

Indica el modo de Stereo usado para el chip AY: ACB,ABC o Mono.

5.8.4 Emulación de Ruido

Permite desactivar la emulación de canales de ruido.

5.8.5 Ver Registros Chip AY

Con esta opción podremos ver los 16 registros internos del chip de sonido (siempre que éste exista).

5.8.6 Iniciar grabación audio

Con esta opción se inicia la grabación de la salida de sonido generado en el emulador, tanto del beeper como del chip AY. No es necesario tener tarjeta Sound Blaster para ello. Se genera un archivo de audio “en crudo”, sin cabecera ni compresión. Exactamente, el modo resultante es: stereo, 8 bit signed, 15550 Hz (15600 Hz para modos de 48k).

Se puede reproducir el archivo resultante con el programa Mplayer, por ejemplo:

```
mplayer -demuxer 20 -rawaudio channels=2:rate=15550:samplesize=1 archivo.rwa
```

Para un archivo guardado en modo 128k.

Si se quiere reproducir simultáneamente un archivo de vídeo en crudo (ver sección 5.4.5 Iniciar grabación vídeo):

```
mplayer -demuxer rawvideo -rawvideo fps=2:w=320:h=200:format=bgr24 archivo.rwv  
-audiofile archivo.rwa -audio-demuxer 20 -rawaudio channels=2:rate=15550:samplesize=1
```

Se debe tener en cuenta que la grabación de vídeo, al igual que la de audio, se genera siempre que se esté en modo multitarea o cuando se sale del menú y se vuelve a la emulación del Spectrum. En el caso que se grabe simultáneamente audio y vídeo, se recomienda empezar con la multitarea desactivada. De esa manera, al volver al emulador, el audio y vídeo empezará simultáneamente y estarán sincronizados. Después de empezar a grabar se puede activar la multitarea. Aunque mientras está la grabación activa el sonido se puede escuchar mal o el vídeo puede ir a menos frames de lo normal, el archivo generado siempre será correcto.

5.8.7 Fichero

Con esta opción se elige el archivo utilizado al grabar audio en archivo. Al utilizar el selector de archivos, se prefiere la extensión .RWA.

5.9 Selección de Idioma

Con esta opción podremos cambiar el idioma del menú (Español o Inglés).

5.10 Multitarea

Con esta opción indicamos si queremos que mientras está el menú activo, la emulación del Z80 también esté activa.

5.11 Velocidad CPU

Desde aquí podremos cambiar la velocidad relativa del Z80.

5.12 Ayuda

Esta opción muestra una pantalla de ayuda.

5.13 Volver al Emulador

Como su nombre indica, volveremos a la emulación del Spectrum.

5.14 Salir del Emulador

Como también indica su nombre, saldremos del emulador.

6 Desarrollo y compilación

El código fuente del emulador está dividido en varios tipos de archivos:

- Archivos .INC: Código assembler.
- Archivos .ASM: Código assembler, al igual que los archivos .INC. Estos archivos .ASM incluyen a los archivos .INC y cada uno genera un archivo objeto (.OBJ)
- macros.mac: Archivo con varios macros auxiliares

Estos tres tipos de archivos son los utilizados para generar el ejecutable final: ZXSPECTR.COM
Se debe compilar utilizando el programa TASM de Borland.

Para compilar el programa debemos ejecutar, desde MS-DOS:

`compile.bat`

Esto generará los archivos objetos. Para enlazarlos todos en el ejecutable:

`link.bat`

Para generar el archivo zip final con los binarios se puede ejecutar:

`make_bin.bat`

O `make_bin.sh` en consola unix/linux

Para generar el archivo zip final con el código fuente se puede ejecutar:

`make_src.bat`

O `make_src.sh` en consola unix/linux

Las utilidades aparte, que tienen versiones para MSDOS y Linux, se han compilado con Turbo C (en MS-DOS) y con GCC (en Linux)

Para compilar las utilidades en Linux hay que ejecutar:

`make -f Makefile.utils`

Esto generará los binarios para CPU de 32 y 64 bits.

Para compilar sólo los binarios para 32 bits:

`make -f Makefile.utils all32`

Y para compilar sólo para 64:

`make -f Makefile.utils all64`

Hay que tener en cuenta que las versiones originales de MSDOS se generaron considerando un código de 16 bits, por tanto, los tipos de datos “int” son de 16 bits. En las versiones de Linux esto está corregido para usar tipos de datos “short”.

7 Utilidades incluidas

En este apartado se habla en ocasiones del formato de archivo de audio SMP. Este formato de archivo se usaba en el programa de edición de sonido Fast Tracker, que es con el que empecé a hacer pruebas de conversión de cintas de casete. Este formato es en crudo (raw), mono, a 8 bits unsigned, y a 11111Hz.

Utilizando utilidades de conversión de sonido se puede pasar de formato wav a smp y viceversa. Por ejemplo, en Linux, con el programa sox, se puede pasar de .wav a .smp con la sentencia:

```
sox archivo.wav -t raw -r 11111 -b 8 -e unsigned -c 1 archivo.smp
```

7.1 LINEASMP

Este programa sirve para leer sonido a través de la entrada LINE IN de la Sound Blaster y grabarlo como fichero SMP.

Como adivinarás, no funciona desde dentro de Windows, y tampoco va bien con el SMARTDRV en memoria.

Al ejecutarlo, se detectará la presencia de Sound Blaster, y a continuación podremos especificar la velocidad de carga (en baudios). Después deberemos introducir el fichero SMP a generar, indicando también extensión. Cuando deseemos terminar de leer el sonido, deberemos pulsar una tecla.

Se puede cargar en turbo especificando adecuadamente los baudios de la grabación.

7.2 SMPATAP

Este programa sirve para leer datos de un fichero de sonido SMP y grabarlo como formato TAP. Con este programa doy la posibilidad de poder cargar programas desde ordenadores que no tengan Sound Blaster.

Al ejecutar el programa, se debe especificar el fichero SMP y luego el TAP (ambos con extensión). Cuando empieza a cargar, se debe pulsar cualquier tecla excepto la N para grabar el bloque.

7.3 TAPABIN

Este programa sirve para sacar bloques de datos de un fichero TAP y grabarlo como formato BIN.

Al ejecutar el programa, se debe especificar el fichero TAP y luego el BIN (ambos con extensión). Cuando lea un bloque, se debe pulsar cualquier tecla excepto la N para grabar el bloque. El fichero que se graba es el que se obtiene de eliminar el flag y el checksum de un bloque; por

ejemplo: si queremos grabar una pantalla del Spectrum en formato BIN, al ejecutar TAPABIN, debemos grabar el bloque que indique: "Flag:255 Longitud:6912" y no el que diga: "Bytes:Pantalla Longitud:6912 Inicio:16384", por que si no, grabaría la cabecera, o sea, los 17 bytes.

7.4 SP_Z80

Este programa sirve para convertir ficheros entre los formatos SP y Z80. Al ejecutarlo, se nos pedirá el fichero origen y el destino, debiendo incluir las extensiones (siempre .SP o .Z80). Sólo permite hacer conversiones de ficheros de 48k.

7.5 VGA2RAW

Este programa se utiliza para convertir el archivo de vídeo generado del formato paleta de VGA a formato en crudo BGR24.

8 Utilidades en formato cinta (TAP)

8.1 SPED52.TAP

Esta cinta contiene un ensamblador/desensamblador para ZX Spectrum 128k, junto con su código fuente. Este ensamblador (SPED) aprovecha al máximo los 128k de memoria. El SPED se aloja en la RAM1, el código objeto en la RAM 4, el código fuente usa las RAM "48" (5,2,0) y las RAMs 6 y 7, lo que nos da un código fuente máximo de 74549 bytes (41781+16384+16384).

Las sentencias ORG y ENT tienen un significado ligeramente distinto al de otros Ensambladores. A diferencia del GENS, aquí el ORG siempre indica en qué dirección de memoria se va a generar el código objeto; lo normal es ensamblar en la 49152, que corresponde a la RAM 4. El ENT indica en qué dirección de memoria van referidas las etiquetas (en un programa reubicable, el ENT debe ser normalmente 0).

También, el operador \$ siempre indica en que dirección de memoria está referida la línea donde esté situado (indicado con ENT). Y el operador !, que me he inventado, indica donde se está generando el código objeto (indicado con ORG).

Para que aparezca el listado mientras se ensambla se debe indicar en el código fuente la orden LST+, y para que no aparezca, LST- .

El ensamblador puede generar información de depuración, que consiste en tener una tabla que contiene cada dirección de memoria del código objeto junto con su número de línea del código fuente. Para esta tabla usa la RAM 7; de esto se deduce que si se usa el tercer bloque de texto no se podrá generar información de depuración. El tamaño máximo de la tabla es de 16k (4096 líneas). Para decir al ensamblador que cree la información, se debe incluir en el código fuente la orden DPR+ (y para desactivarla, DPR-). Al ir al desensamblador, si hay tabla de depuración, se mostrará la línea del código fuente.

La tabla de etiquetas usa la RAM 3, lo que nos da un máximo de 16k para etiquetas (esto supone unas 1260 etiquetas como máximo).

El editor también utiliza la RAM 3, para copiar texto en un buffer y poder hacer las acciones típicas de copiar/pegar/mover; el máximo debe ser 16k de texto.

Debo decir que si no tienes memoria EMS, al ensamblar puedes estar *muchiiiiisimo* tiempo esperando; lo mejor es salir al MS-DOS con la EMS activa.

No me quiero extender sobre las opciones del programa, pues posee ayuda propia (en el entorno principal y el desensamblador, tecla H; en el editor, Modo Extendido y H). Además, tienes el código fuente para poder verlo enterito.

Sólo una cosa más, el formato SPED. El ensamblador sólo entiende códigos fuente con cabeceras en formato SPED, que son de 34 bytes pero que tienen la ventaja de que se pueden leer desde el BASIC. La descripción de la cabecera es:

<i>Offset</i>	<i>Longitud</i>	<i>Descripción</i>
0	1	Byte que está a 3. Indica bloque de BYTES
1	10	Nombre del fichero
11	2	Word que indica la longitud del primer bloque de texto
13	4	4 Bytes no usados
17	1	Byte de Checksum. Se utiliza para "engañar" al BASIC y decirle que la cabecera es de 17 bytes; se puede leer la cabecera en un Spectrum real y en el emulador
18	1	Día de grabación del código fuente
19	1	Mes de grabación del código fuente
20	1	Año-1980 de grabación del código fuente. Está preparado para el efecto 2000!
22	2	Word que indica la longitud del segundo bloque de texto. Si no lo hay, a 0
24	2	Word que indica la longitud del tercer bloque de texto. Si no lo hay, a 0
26	9	9 Bytes a 255, usados en futuras versiones

Después de la cabecera viene el texto, formado por 1, 2 o 3 bloques con flag 255. Los bloques de texto se graban en formato ASCII, poniendo el bit 7 de un carácter a 1 cuando queremos indicar el tabulador, y el código 13 para saltos de línea. De los ensambladores que tengo para Spectrum (SPED, GENS, TED), éste es el que utiliza un formato de texto más corto.

8.2 CONVERSO.TAP

Este programa es un conversor de códigos fuente de formatos GENS, TED y SPED. Las conversiones posibles son: GEN-> TED, GEN->SPED, TED->SPED. Se incluye su código fuente en formato SPED. Nota: El TED es un formato de un editor/ensamblador que apareció en la revista MICROHOBBY, de Hobby Press.

8.3 REALDEBU.TAP

Consiste en dos desensambladores, uno para 48k y otro para 128k. La apariencia que tienen es parecida al MONS. Se incluye los códigos fuentes de ambos, en formato SPED. El nombre de ambos es (R)EAL DEBUG, y es debido a que controlan perfectamente el registro R, cosa que otros

desensambladores, como el MONS, no hacen.

El de 48k es totalmente reubicable. Para poderlo cargar hacer:
CLEAR dirección-1: LOAD "RDEBUG6.CO" CODE dirección:
RANDOMIZE USR dirección

Para obtener las teclas del de 48k puedes verlas del de 128k, pues tiene ayuda; sólo hay dos opciones no disponibles, la de selección de RAM y la de ensamblar.

El de 128k se ubica en la RAM 1, más un bloque de 256 bytes que debe estar por debajo de la 49152. Para cargarlo hacer:

Desde el 128 BASIC:
RANDOMIZE USR 0.
Con esto, iremos a la ROM3. Luego:

CLEAR 29999: OUT 32765,17: LOAD "R37DE128.C" CODE 49152: RANDOMIZE
dirección: RANDOMIZE USR 49152

El CLEAR es arbitrario, aunque debe estar por debajo de 49152. La variable indicada como "dirección" es la que indica la posición de memoria donde se ubican los 256 bytes; se puede poner, por ejemplo, en la 23296. Si se pone en otra que no sea la 23296, el CLEAR debe estar por debajo de "dirección".

8.4 CURSORDR.TAP

Se trata de un programa de dibujo, el CURSOR DRAW, que incluye también su código fuente en formato SPED.

Y esto es todo por mi parte. Espero que disfrutes del emulador. :-)

CESAR HERNANDEZ BAÑO
chernandezba@hotmail.com