

ZXAM

Antonio J. Rosselló

Copyright © 1993-95 WareSoft

COLLABORATORS

	<i>TITLE :</i> ZXAM	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY	Antonio J. Rosselló	February 12, 2023
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ZXAM	1
1.1	ZXAM Guide	1
1.2	Distribución del ZXAM	2
1.3	Capacidades del emulador	4
1.4	Requerimientos para usar el emulador	4
1.5	La cuestión del AGA...	5
1.6	¿ Que es esta cosa ?	6
1.7	Formatos de snapshot de otros emuladores	6
1.8	Como arrancar el emulador	7
1.9	ToolType PATTERN	8
1.10	ToolType LOADPATH	9
1.11	ToolType SAVEPATH	9
1.12	ToolType DEFFORMAT	9
1.13	ToolType REXXPATH	9
1.14	ToolType ROMFILE	9
1.15	ToolType PRERUN	10
1.16	ToolType POSTRUN	10
1.17	ToolType LOADMODE	10
1.18	ToolType SAVEMODE	10
1.19	ToolType OUTPUT	10
1.20	ToolType AUDIO	11
1.21	ToolType AYEMULATION	11
1.22	ToolType SPEED	11
1.23	ToolType JOYSTICK	11
1.24	ToolType TASKPRI	11
1.25	ToolType DISPLAY	11
1.26	ToolType REFRESH	12
1.27	ToolType SCREENMODE	12
1.28	ToolType PUBSCREENNAME	12
1.29	La ventana de control del emulador	12

1.30	Gadget Load Program	13
1.31	Gadget Reload	14
1.32	Gadget Save Program	14
1.33	Gadget Poke	14
1.34	Recuadro Save Format	14
1.35	Gadget RUN	14
1.36	Gadget STOP	14
1.37	Gadget Run ARexx	15
1.38	Gadget Abort ARexx	15
1.39	String Dirección	15
1.40	String Dato	15
1.41	Texto Fichero Cargado	15
1.42	Texto Formato	15
1.43	Opciones de la emulación	16
1.44	Las entrañables teclas de chicle...	17
1.45	Funciones ARexx incorporadas al ZXAM	18
1.46	ZXAMGetReg	23
1.47	ZXAMSetReg	24
1.48	ZXAMPeek	24
1.49	ZXAMPoke	24
1.50	ZXAMDPeek	24
1.51	ZXAMDPoke	24
1.52	ZXAMGetMem	24
1.53	ZXAMPutMem	25
1.54	ZXAMFindByte	25
1.55	ZXAMRun	25
1.56	ZXAMQuit	25
1.57	ZXAMppLoadFile	25
1.58	ZXAMppSaveFile	25
1.59	ZXAMLoadRequester	26
1.60	ZXAMSaveRequester	26
1.61	ZXAMNameFormat	26
1.62	ZXAMActName	26
1.63	ZXAMActLoadPath	26
1.64	ZXAMActSavePath	26
1.65	ZXAMActFormat	27
1.66	ZXAMActSaveFormat	27
1.67	ZXAMProgVersion	27
1.68	ZXAMSaveFormat	27

1.69 ZXAMParseLoaded	27
1.70 ZXAMParseToSave	28
1.71 ZXAMJoinPathName	28
1.72 ZXAMGetScr	28
1.73 ZXAMClearNameFormat	28
1.74 ZXAMLoadPath	28
1.75 ZXAMSavePath	28
1.76 ZXAMFilePart	28
1.77 ZXAMPathPart	29
1.78 ZXAMPreRun	29
1.79 ZXAMActPreRun	29
1.80 ZXAMPostRun	29
1.81 ZXAMActPostRun	29
1.82 ZXAMPattern	29
1.83 ZXAMActPattern	29
1.84 ZXAMBASICToken	30
1.85 ZXAMReadAbort	30
1.86 ZXAMClearAbort	30
1.87 ZXAMNoReload	30
1.88 ZXAMEnableAbort	30
1.89 ZXAMDisableAbort	30
1.90 ZXAMDisassemble	31
1.91 ZXAMLoadMode	31
1.92 ZXAMSaveMode	31
1.93 ZXAMActLoadMode	31
1.94 ZXAMActSaveMode	32
1.95 ZXAMAbout	32
1.96 ZXAMFindBlock	32
1.97 ZXAMDoChecksum	32
1.98 ZXAMBreakPoint	33
1.99 ZXAMTrace	33
1.100ZXAMDoInt	33
1.101ZXAMBlockOfRegs	33
1.102Registros	34
1.103ZXAMAUDIO	35
1.104ZXAMAYEmulation	35
1.105ZXAMSpeed	35
1.106ZXAMJoystick	35
1.107ZXAMTaskPri	36

1.108ZXAMDisplay	36
1.109ZXAMRefresh	36
1.110ZXAMActAudio	36
1.111ZXAMActAYEmulation	36
1.112ZXAMActSpeed	37
1.113ZXAMActJoystick	37
1.114ZXAMActTaskPri	37
1.115ZXAMActDisplay	37
1.116ZXAMActRefresh	38
1.117ZXAMEmulToFront	38
1.118ZXAMControlToFront	38
1.119ZXAMActRun	38
1.120ZXAMAYBlockOfRegs	38
1.121ZXAMWriteROM	38
1.122ZXAMCheckSumString	39
1.123ZXAMCompareString	39
1.124ZXAMStop	39
1.125ZXAMTapeAction	39
1.126ZXAMTapeLoadName	39
1.127ZXAMTapeSaveName	40
1.128ZXAMTapeAddress	40
1.129ZXAMTapeSize	40
1.130ZXAMTapeFlag	40
1.131ZXAMTapeLoadOffset	40
1.132ZXAMTapeSaveOffset	41
1.133ZXAMTapeSetResult	41
1.134ZXAMTapeMark	41
1.135ZXAMTapeSetLoadName	41
1.136ZXAMTapeSetSaveName	41
1.137ZXAMTapeSetLoadOffset	42
1.138ZXAMTapeSetSaveOffset	42
1.139ZXAMTapeSetMark	42
1.140Curiosidades...	42
1.141Sobre esta versión....	43
1.142Problemas conocidos...	44
1.143Versiones del ZXAM	47
1.1441.0 β	48
1.1451.1 β	48
1.1461.2 β	48

1.1471.3 ß	49
1.1481.3b ß	49
1.1491.4 ß	50
1.1501.5	51
1.1511.6	52
1.1521.6b	53
1.1532.0	54
1.154Futuro del emulador	58
1.155¡Gracias a toda esta gente!	58
1.156¿ Quien ? ¿ Donde ?	59
1.157Mensaje para los usuarios registrados	60

Chapter 1

ZXAM

1.1 ZXAM Guide

ZXAM SPECTRUM EMULATOR v2.0

© 1993-95 WareSoft Todos los derechos reservados
Programado por Antonio J. Pomar Rosselló

Distribución

Distribución y registro.

Capacidades

¿ Que puede hacer el ZXAM ?

Introducción

Sobre el ZXAM...

Requerimientos

¿ Que necesitas para usarlo ?

Versión AGA

¿ Por que la versión AGA ?

Formatos de ficheros

Formatos soportados.

Arrancando el ZXAM

Opciones de arranque.

La ventana de control

Manejo del emulador.

Menus

Opciones de la emulación.

Teclado del Spectrum

Las viejas teclas de chicle.

Puerto ARexx

Para el que quiera hacer maravillas.

Información técnica

Curiosidades.

Bugs conocidos

"Cualidades" imprevistas...

Sobre esta versión

Notas específicas sobre esta versión.

Historia

Versiones del ZXAM, hasta este momento.

Futuro

¿ Como acabará esto ?

Agradecimientos

Un montón de gente.

Contactos

Como contactar con el autor.

Usuarios registrados

Nota importante para usuarios registrados.

1.2 Distribución del ZXAM

Este programa es SHAREWARE. Esto significa que tienes un mes de tiempo para probarlo, y si te gusta deberías mandar 15 US\$ (unas 2000 pts si quieres mandarlo en pesetas) al autor. A cambio recibirás la última versión disponible (probablemente la versión 128k estará disponible para otoño de 1995). Con esta contribución estás asegurando el desarrollo y mejora posterior del emulador. Gracias por adelantado.

Esta versión SHAREWARE puede circular libremente siempre que se incluyan todos los ficheros asociados al programa. Está prohibido utilizar el programa para fines comerciales, o sea que solo se pueden cargar los gastos normales de copia y envío que se cargan habitualmente en las librerías de Dominio Público. También puedes subirlo a BBS, incluirlo en discos de revista o en CDROMs recopilatorios de dominio público, freeware y shareware (como los CDs Aminet).

Si te interesa conseguir la versión mas reciente disponible del emulador puedes hacer File Request (FidoNet) a Tanit-BBS con el Magic Name 'ZXAM' (tfno: 971-392829, USRobotics 28.800 baudios), o entrar por terminal y buscarla en la lista de ficheros.

El paquete completo de distribución comprende:

ZXAM_Spectrum	Versión GENERICA del ZXAM (con icono).
ZXAM_Spectrum_FASTER	Versión rápida del anterior (con icono).
ZXAM_SpectrumAGA	Versión ESPECIAL AGA del ZXAM (con icono).
ZXAM_SpectrumAGA_FASTER	El mismo, pero versión rápida (con icono).
ZXAM_REXX/	Cajón con varios scripts ARexx de ejemplo.
ZXAM_Español.guide	El texto que lees ahora (con icono).
ZXAM_English.guide	Este mismo GUIDE, pero en inglés (con icono).
LEEME!!	Notas de distribución (con icono).
README!!	Lo mismo, pero en inglés (con icono).
INTERFACE.IFF	Esquema del interface de cinta (con icono).
DIGIT_INTERFACE.IFF	Digitalización de mi interface (con icono).
powerpacker.library	Necesaria para cargar ficheros comprimidos.

Los ejecutables están comprimidos para facilitar la distribución de este paquete en discos de doble densidad (revistas y librerías de DP), ya que el paquete es casi exactamente del tamaño de un disco de doble densidad en formato FastFileSystem. Los ejecutables sin comprimir ocupan 450Kb cada uno (debido a la imposibilidad del sistema de unir, en memoria, varios hunks DATA y BSS en un orden determinado), por lo que, obviamente, la compresión es imprescindible.

El autor del programa no acepta ninguna responsabilidad sobre cualquier perjuicio que puede suponer el uso o abuso del programa, ni ofrece ninguna garantía sobre el programa y sus capacidades. Tampoco se garantiza nada sobre la precisión o estilo literario de este documento ni se acepta ninguna responsabilidad sobre las consecuencias que puede acarrear su lectura. Tampoco acepta ninguna responsabilidad sobre cualquier consecuencia que puede tener la realización o uso del interface de cassette. Esta prohibido montar y vender el interface para sacar un provecho económico.

Se reconocen todas las marcas registradas, marcas comerciales y Copyrights de los productos mencionados en este documento.

Por lo que he podido leer en el DOC del Spectrum Emulator v1.7 la casa AMSTRAD, que posee ahora los derechos sobre los componentes del Spectrum, ha dado permiso especial a los autores de emuladores para poder distribuir la ROM del Spectrum original. De todas formas el Copyright sobre la ROM sigue siendo de AMSTRAD.

La pantalla KeysHelp.iff ha sido incluida con el permiso del autor (Mikael Ostergren) para ayudar a localizar teclas en el spectrum 48k.

Los iconos MagicWB del emulador han sido creados por Richard Harris.

Los iconos Newicons los he dibujado yo mismo, o sea que si no te gustan ya sabes quien es el culpable ;-)

La powerpacker.library es © Nico François.

El Sinclair ZX Spectrum Emulator es © Peter McGavin.

El Sinclair ZX Spectrum Emulator "Z80" es © Gerton Lunter.

El emulador ZX Spectrum (SPEC386) es © Pedro Gimeno.

1.3 Capacidades del emulador

- Emulación de un spectrum 48k.
- Sonido 48k estabilizado.
- Carga/grabación directamente a cinta, o desviada hacia disco o hacia scripts ARexx especializados para gestionar formatos como el .TAP.
- Puerto ARexx con 93 funciones.
- Multitarea completa.
- Ajuste automático de velocidad.
- Consumo optimizado de CPU. La potencia no necesaria se deja disponible para otras tareas (en un 1200/030/28MHz se pueden ejecutar dos Manic Miner en multitarea a velocidad real, y aún sobra un 10% de CPU).
- Emulación del chip de sonido del Spectrum 128k (AY-3-8912), incluyendo el canal de ruido blanco, las envolventes de volumen y los samples de 4 bits.
- El stereo sigue las distribuciones de canales ABC y ACB usadas en los interfaces AY para Spectrum 48k (el canal izquierdo de Amiga es usado para el canal A y el de sonido 48k, y el derecho para los canales B y C).
- Asignación dinámica del canal de ruido del chip AY para sonido 128k o 48k.
- Inclusión de la pantalla del Spectrum emulado en una ventana sobre el workbench o cualquier otra pantalla pública. Esto incluye emulaciones de workbench con tarjetas RTG, como la Retina. El emulador se adapta a los colores de la pantalla para reasignar los colores de la ventana del Spectrum y conseguir el mayor parecido posible.
- Posibilidad de abrir pantallas legales con el sistema, incluso al desactivar la multitarea, para monitores que no soportan PAL.
- Emulación de joysticks tipo Kempston, Sinclair (Interface II) y Cursor (AGF/Protek).
- Incluye un transfer para pasar a cinta los juegos de spectrum.
- Border real, aunque no esta sincronizado con el barrido de la pantalla
- FLASH emulado (solo versión AGA).
- Versión FASTER para maxima velocidad a costa de una emulación menos completa.
- El emulador se adapta al hardware disponible. En el caso de no disponer de los canales de sonido o de las CIAs para temporizar, el emulador desactivará automáticamente las opciones que usan dicho hardware.
- El hardware necesario para la emulación (canales de sonido, CIA, etc...) se aloja solo en el momento de arrancar la emulación, y es devuelta al sistema cuando ésta es detenida. Solo se aloja el hardware necesario para las opciones seleccionadas en el menú 'Options'.
- Las ventanas del emulador son appwindows (tanto la de control como la de emulación), o sea, permiten que se les 'tiren' iconos dentro, para cargar los snapshots.
- Interface de cassette propio, mucho mas barato y fiable que un sampler.
- Posibilidad de ejecutar tantos Spectrums en multitarea como se quiera. El único límite para el numero de copias simultaneas del ZXAM que pueden correr es la memoria y la potencia del procesador (o la paciencia del usuario ;-)
- Ejecución de código Z80 paso a paso, desemsamblador Z80 y breakpoints.
- Emulación completa del registro R del Z80.
- Emulación completa del BCD.
- Emulación de todas las instrucciones 'secretas' del Z80.
- Versión AGA conmutable para trabajar en B/N o Color.

1.4 Requerimientos para usar el emulador

La versión genérica (Generic) del emulador necesita un Amiga con procesador

68020 (o superior) y la versión 2.04 (o superior) del sistema operativo. La memoria FAST 32 bits es casi imprescindible ya que sin ella la emulación se frenará considerablemente.

La versión especial AGA necesita WorkBench 3.0 (o superior), los chips gráficos AGA y un microprocesador 68020 (o superior). En resumen, necesita un Amiga 1200 como mínimo.

Todas las versiones del ZXAM necesitan, además, la `powerpacker.library` en `libs:` o en el mismo directorio del programa para poder acceder a los ficheros, aunque no se encuentren comprimidos.

Por otro lado, para los requesters de ficheros se usa la librería standard del sistema '`asl.library`'. Esta librería tiene el problema de releer el directorio cada vez que abre el requester, con lo que puede ser desesperante cargar ficheros desde disquete, o cuando en el disco duro hay muchos ficheros (como en mi caso, que son 900). Para evitar esto lo mejor sería usar la `reqtools.library`, ya que su requester tiene un buffer que evita tener que leer el directorio cada vez, pero debido a graves problemas técnicos aún no he podido implementarlo. Mientras tanto la mejor opción es usar el programa `RTPatch`, de Nico François, que parchea las librerías `asl/req/arp` para que usen el requester de la `reqtools`. Así, además, todos los demás programas que usen `asl/req/arp` usarán dicho requester, que es mucho mejor que los originales. Curiosamente con `RTPatch 1.3` (que es el parche que yo uso) el buffer funciona perfectamente, pero con el `ReqChange 3.0` (otro parche que se supone hace lo mismo) el buffer no funciona.

Otra cosa interesante puede ser instalar la commodity del sistema operativo '`MouseBlanker`'. En modos multitarea puede ser bastante molesto tener el puntero del ratón en pantalla. La mejor solución, con diferencia, es instalar la commodity mencionada, que hace desaparecer el puntero del ratón al pulsar cualquier tecla, y que reaparece al mover el ratón. De esta forma se soluciona el problema de una forma totalmente legal y compatible con el sistema.

1.5 La cuestión del AGA...

¿Que tiene de AGA la versión especial del emulador?: Pues usa algunas de las nuevas posibilidades específicas de los chips AGA para la conversión de pixels y colores de la pantalla del Spectrum, para adaptarla a los bitplanos del Amiga. Estas nuevas posibilidades aceleran considerablemente el acceso a pantalla de la emulación, siendo notable la diferencia de velocidad en juegos con grandes scrolls, como el `Uridium`, y en demos.

Por otro lado este modo especial solo puede usarse desactivando la multitarea y mostrando la imagen en formato PAL. Es por esto que la versión AGA no incorpora las nuevas posibilidades de multitarea de la versión 2.0, y se recomienda su uso solo cuando se necesite la máxima potencia gráfica disponible. En caso de usar monitores VGA, tarjetas RTG o de necesitar la multitarea, tendrás que usar la versión genérica. Otro detalle es que solo la versión AGA incorpora FLASH. La versión genérica probablemente incorporará FLASH en posteriores versiones, pero de momento lo que hace es mostrar el FLASH como un INVERSE (intercambiando papel y tinta).

1.6 ¿ Que es esta cosa ?

Como habrás deducido de forma inmediata este programa tiene como función principal (y única) la de emular un Sinclair ZX Spectrum 48k, con sus entrañables atributos y su sonido de 1 bit (lagrimilla...). A partir de la versión 2.0 el ZXAM incorpora la emulación del chip de sonido AY-3-8912 del Spectrum 128k, en preparación para la futura versión 128k del emulador.

1.7 Formatos de snapshot de otros emuladores

Antes de poder comprender la función de algunos de los gadgets y ToolTypes del programa debemos discutir algo sobre formatos de programas de Spectrum. Actualmente hay varios emuladores de Spectrum 48k para diferentes ordenadores (Amiga, PC, Atari, etc...) y cada uno de estos emuladores graba los programas en disco en un formato de fichero diferente (a partir de este punto llamaremos 'snapshot' a estos ficheros, ya que es la nomenclatura habitual). Esto significa que para poder usar snapshots grabados con un emulador en otro debes disponer de algún tipo de utilidad que convierta los programas de un formato a otro. El ZXAM intenta evitar la necesidad de conversor admitiendo cargar y grabar directamente varios formatos en vez de tener un solo formato propio. Así también se puede usar el emulador como conversor de formatos, si surge la necesidad.

De momento hay implementados cuatro formatos (los únicos que conozco) a los que denomino PC, MIRAGE, Z80 y KGB, y que son:

PC: Este formato proviene del emulador de Spectrum para PC programado por Pedro Gimeno. El ejecutable se llama SPEC386.EXE y la versión que he usado (yo no tengo PC) es la 0.99D B. Supongo que a estas alturas debe haber una versión mas completa en circulación (creo recordar que este emulador ya tiene unos años). Necesita un 386DX a 25MHz con un caché de 256Kb para acercarse a la velocidad del Spectrum real. Estos ficheros se reconocen por que ocupan 49190 bytes en el disco y suelen llevar la extensión .SP al final del nombre.

MIRAGE: Este es el formato del Spectrum Emulator 1.7 de Peter McGavin. Es un emulador para AMIGAs sin chips AGA. Los ficheros se reconocen por que ocupan 49179 bytes y suelen llevar la extensión .snapshot, aunque yo prefiero cambiarsela por .mirage para evitar confusiones, ya hay varios emuladores que usan esa extensión.

Hay que notar que hasta la versión 1.6 del Spectrum Emulator este formato no guardaba el color actual del borde, por lo que al cargar ficheros de formato pre-1.6 el borde arrancará en negro. Para arreglar esto puedes regrabar en dicho formato, ya que el ZXAM utiliza el formato moderno, que almacena el borde correctamente. Este formato también lo usan los emuladores (para PC) VGASPEC (*.SNA, no sé el autor) y JPP (*.SP, de Arnt Gulbrandsen).

Z80: este formato (implementado a partir de la versión 1.6) es el que usa el emulador de Spectrum (para PC) 'Z80' de G.A.Lunter. El ZXAM admite la carga de ficheros .Z80 generados por las versiones 1.45, 2.01 y 3.0 del Z80, siempre que sea un fichero de Spectrum 48k. En caso de que sea 128k o necesite de la emulación de interface I o cualquier otro hardware no soportado por el ZXAM, el fichero será

rechazado como desconocido. Este formato solo está disponible para cargar, pero no para grabar.

KGB: este fomato (implementado a partir de la versión 1.6b) es el usado por el viejo emulador KGB para Amiga. Se puede reconocer por que el tamaño es siempre 49486 bytes. No se quien es el autor de dicho emulador (la versión que yo tengo no incluye DOC ni nada parecido) ni si aún se usa debido a que es incompatible con la versión 2.0 y superiores del sistema operativo. Este fomato está disponible solo para cargar.

En todos los casos el emulador admite que los programas estén comprimidos con powerpacker.

Hay que notar que el emulador no usa NUNCA la extensión del nombre del fichero para identificar su formato. En vez de eso se hacen una serie de comprobaciones en la estructura del fichero para reconocerlo. Si quieres poner otra extensión a un fichero, puedes hacerlo sin que ello afecte la posibilidad de cargarlo en el emulador. Eso si, tienes que añadir esa extensión a la plantilla (tooltype PATTERN en el icono del emulador) del requester de ficheros, ya que si no esos ficheros no aparecerán en él.

Además de estos formatos hay otro, el llamado .TAP, que no es un snapshot en si mismo si no que es una serie de cabeceras y bloques de datos unidos en un solo fichero, como si de una cinta de tratara. Junto al emulador hay un script ARexx que puedes usar para descomponer un fichero .TAP en la serie del .header y .bytes que contiene, y poder cargar así dichos bloques con la opción de 'Load Mode -> Disk', o en el Spectrum Emulator de Peter McGavin. A partir de la versión 2.0 se incluye un script ARexx que permite cargar y crear automáticamente ficheros .TAP directamente.

En caso de que tengas información sobre otros formatos de otros emuladores, aunque sean de otros ordenadores, solo tienes que mandarme un disco con varios programas en ese formato y la información de que dispones. Procuraré incluir en la siguiente versión del emulador la posibilidad de usar dicho formato. Es MUY importante que me mandes algún programa grabado en ese formato, para poder probar la carga en dicho formato.

1.8 Como arrancar el emulador

El programa está pensado para arrancar desde WorkBench ya que ←
las opciones
se leen del icono del programa. De todas formas, si lo arrancas desde el SHELL el programa buscará el icono en el mismo directorio del ejecutable, para cargar dichas opciones. Además, se "despegará" de la ventana del SHELL. y cargará el snapshot que le hayas especificado en la línea de comandos, si ese es el caso.

Si hay algún problema al arrancar el programa, éste lo comunicará con un requester y volverá al sistema operativo.

También se puede meter el ZXAM como 'Default Tool' en el icono de un snapshot, para que al darle a ese snapshot se arranque el ZXAM y se cargue el fichero. También se admite multiselección al arrancar, aunque solo se cargará el primer programa seleccionado.

Las ToolTypes ("Tipos Herramientas" en WB3.0 en castellano) que se pueden meter en el icono son, en buena parte, equivalentes directos de los menús de control del emulador, y sirven para especificar el estado inicial al que se desean dichos menús al arrancar. Los disponibles en la versión 2.0 son:

LOADPATH

SAVEPATH

PATTERN

REXXPATH

ROMFILE

PRERUN

POSTRUN

OUTPUT

DEFFORMAT

Valores por defecto para el menú 'Options':

LOADMODE

SAVEMODE

AUDIO

AYEMULATION

SPEED

JOYSTICK

PUBSCREENNAME

ToolTypes NO disponibles en versión AGA....

TASKPRI

DISPLAY

REFRESH

SCREENMODE

1.9 ToolType PATTERN

PATTERN= Le indicamos el pattern (plantilla) para el requester de ficheros. Este pattern es un filtro que especifica que ficheros deben mostrarse en

dicho requester. Consulta el manual del ordenador sobre el uso de plantillas de fichero.

1.10 ToolType LOADPATH

LOADPATH= Indica en que directorio queremos que se abra inicialmente el requester de cargar. Esto es útil para dirigirlo al directorio del disco duro donde guardas normalmente los programas de Spectrum y no tener que meter cada vez el path (yo tengo en el disco duro un directorio con 900 juegos comprimidos [4.7 Megas]).

1.11 ToolType SAVEPATH

SAVEPATH= Indica en que directorio queremos que se abra inicialmente el requester de grabar.

1.12 ToolType DEFFORMAT

DEFFORMAT= Indica el formato por defecto usado para grabar los programas de Spectrum (formato PC, MIRAGE, PC_PP y MIRAGE_PP), o sea, el estado inicial del recuadro 'SAVE FORMAT'.

1.13 ToolType REXXPATH

REXXPATH= Indicamos el directorio donde se debe abrir el requester para seleccionar scripts ARexx. El nombre de los scripts debe acabar en '.zxm'.

1.14 ToolType ROMFILE

ROMFILE= Indicamos la ROM a cargar, substituyendo la interna del emulador (la original del Spectrum 48k). La ROM puede estar comprimida con powerpacker. CUIDADO con las ROM extrañas por que pueden provocar incompatibilidades con algunos juegos (no es culpa mia ;-)

1.15 ToolType PRERUN

PRERUN= Fija una línea de comandos AmigaDOS para ejecutar justo antes de arrancar la emulación. Por ejemplo, con "PRERUN=cpu nocache noburst" se desactivarían los cachés y el burst del procesador. Antes de ejecutar la línea PRERUN el emulador guarda el estado de los cachés y los restaura al volver de la emulación (con la tecla HELP), justo antes de ejecutar la línea POSTRUN. De esta forma se puede poner un comando que desactive los cachés en el momento de emular sin que en el WorkBench tengamos que tener los cachés desactivados. En multitarea esta línea de comandos NO es ejecutada.

1.16 ToolType POSTRUN

POSTRUN= Fija una línea de comandos a ejecutar justo al salir de la emulación (con la tecla HELP). En multitarea esta línea de comandos NO es ejecutada.

1.17 ToolType LOADMODE

LOADMODE= Fija el modo de la rutina LOAD de la ROM (TAPE, DISK o AREXX).

1.18 ToolType SAVEMODE

SAVEMODE= Fija el modo de la rutina SAVE de la ROM (TAPE, DISK, DISK_PP para que se compriman los datos al grabar, o AREXX).

1.19 ToolType OUTPUT

OUTPUT= Especifica un fichero de salida (normalmente una ventana CON:) para que los comandos ejecutados por PRERUN y POSTRUN, y los scripts ARExx, puedan imprimir mensajes. Es una opción muy útil para depurar scripts ARExx, ya que todos los errores generados por el ARExx se dirigirán hacia este canal. Si no se especifica, toda salida de PRERUN, POSTRUN y ARExx es dirigida hacia NIL:.

1.20 ToolType AUDIO

```
AUDIO=          Determina el estado inicial del
menu
'Options ->
Audio'. Los valores posibles son OFF, FAST y
CIASYNC.
```

1.21 ToolType AYEMULATION

```
AYEMULATION=   Determina el estado inicial del
menu
'Options ->
AY-3-8912'. Loas valores posibles son OFF, ON y
SMART.
```

1.22 ToolType SPEED

```
SPEED=         Determina el estado inicial del
menu
'Options ->
Speed'. Los valores posibles son FAST y AUTO.
```

1.23 ToolType JOYSTICK

```
JOYSTICK=      Determina el estado inicial del
menu
'Options ->
Joystick'. Los valores posibles son KEMPSTON,
SINCLAIRRIGHT, SINCLAIRLEFT y CURSOR.
```

1.24 ToolType TASKPRI

```
TASKPRI=       Determina el estado inicial del
menu
'Options ->
TaskPri'. Los valores posibles son +1, 0 y -1.
```

1.25 ToolType DISPLAY

```
DISPLAY=       Determina el estado inicial del
menu
'Options ->
Display'. Los valores posibles son SCREEN, WINDOW,
NOSYSTEMSCREEN y NOSYSTEMPAL.
```

1.26 ToolType REFRESH

REFRESH= Determina el estado inicial del menu 'Options -> Refresh'. Los valores posibles son 1, 2, 4 y 8.

1.27 ToolType SCREENMODE

SCREENMODE= Aqui se puede especificar el modo de pantalla que se quiere usar para los modos de pantalla propia del emulador (modos SCREEN y NOSYSTEMSCREEN). Es válido cualquier modo de pantalla disponible en las preferencias Screenmode del sistema, siempre que soporte un tamaño de 256x192 pixels a 16 colores. Lo que aqui debe ponerse es el nombre del modo de pantalla exactamente igual como aparece en las preferencias 'screenmode' del sistema.

En caso de no especificarse un modo de pantalla con esta ToolType, el emulador usará una pantalla tipo PAL (en sistemas 2.04 y 2.1) o intentará usar el mismo monitor que la pantalla donde se encuentra la ventana de control, (en sistema 3.0 y superiores) mediante la función BestModeID. El emulador admite que la pantalla sea promocionada mediante parches externos (como el NewMode) mientras se cumpla la condición de que la pantalla sea exactamente de 256x192 pixels y 16 colores. La pantalla se abre con el nombre 'ZXAM Screen', aunque no tenga barra de título.

Debido a la forma en la que el emulador accede a esta pantalla, no se puede promocionar hacia una tarjeta RTG. Para usar el emulador sobre RTG tendrás que usar el emulador sobre una ventana en una pantalla pública.

1.28 ToolType PUBSCREENNAME

PUBSCREENNAME= Nombre de la pantalla pública donde deben abrirse tanto la ventana de control como la de emulación. En caso de no especificarse las ventanas se abrirán en la pantalla pública por defecto, que suele ser el Workbench.

1.29 La ventana de control del emulador

Si has arrancado el programa verás que abre una ventana en el WorkBench con

los gadgets de cargar, grabar, poke, etc. Todos los gadgets tienen una letra subrayada, que significa que si pulsas la tecla correspondiente a dicha letra es como si pulsaras el gadget con el ratón. El equivalente de teclado del gadget de cerrar ventana es la tecla Esc, y provoca la salida del emulador. La tecla TAB selecciona el primer gadget numérico de la ventana para poder modificarlo

sin tener que seleccionarlo con el ratón. Mientras estás modificando un gadget numérico puedes pulsar TAB para pasar al siguiente gadget numérico, o Shift+TAB para pasar al anterior. La tecla HELP muestra el requester de 'About...'.
 Aquí tienes un esquema de la ventana de control del emulador. Selecciona el gadget que te interesa para una explicación de lo que hace.

```

+-----+
|o| ZXAM Spectrum Emulator v2.0 (full Generic version) (stopped) | | |
+-----+
|
|                               Loaded File                               |
|
|                               Load Program                             |
|
|                               --- NO FILE ---                           |
|                               Format                                     |
|                               -- NO FORMAT --                           |
|                               |                                         |
|                               Reload                                     |
|
|                               Poke                                       |
|
|                               16384                                     |
|                               ,                                         |
|                               255                                       |
|
|                               SAVE FORMAT                               |
|                               |                                         |
|                               Save Program                               |
|
|                               Mirage O |                               |
|                               Mirage pp O |                               |
|
|                               Run ARexx                               |
|
|                               Run                                       |
|
|                               Stop                                       |
|                               PC O |                               |
|
|                               Abort ARexx                               |
|
|                               PC pp O |                               |
+-----+

```

1.30 Gadget Load Program

Load Program .- Abre el requester de ficheros para que puedas seleccionar el programa que quieres cargar. El emulador detectará automáticamente en que formato está el programa, por lo que no es necesario especificarle nada. **IMPORTANTE:** los programas de Spectrum se pueden comprimir con el programa PowerPacker, y se descomprimirán al cargarlos. Así nos ahorramos un montón de espacio en disco (en un disco de 880k suelen caber unos 45 programas una vez comprimidos). Una vez

cargado se verá el nombre del fichero en el recuadro "Loaded File" y el formato en que está en el recuadro "Format". Si intentas cargar un fichero que no es un programa de Spectrum lo único que puede pasar es que se borre el contenido anterior de la memoria del Spectrum. Hay que notar que el emulador YA es capaz de cargar programas directamente desde cinta (V1.3 ß), pero debes montar el interface descrito en la pantalla INTERFACE.IFF. A partir de la versión 1.5 la ventana del emulador es una AppWindow, o sea, puedes tirar dentro los iconos de los ficheros a cargar. Si lo que metes es un cajón o un disco, se abrirá el requester de ficheros en dicho cajón/disco. A partir de la versión 2.0 también la ventana de emulación es una AppWindow, pudiendo tirarle iconos dentro también.

1.31 Gadget Reload

Reload.- Vuelve a cargar de disco el programa de Spectrum que aparece en la ventana "Loaded File", sin abrir el requester de ficheros.

1.32 Gadget Save Program

Save program.- Graba la memoria del Spectrum en el formato indicado por los gadgets de la derecha (recuadro "SAVE FORMAT").

1.33 Gadget Poke

POKE.- Es facil imaginarselo, ¿no?.

1.34 Recuadro Save Format

Recuadro "Save Format".- En este recuadro hay 4 gadgets que determinan el formato en el que se grabará el snapshot si seleccionamos el gadget "Save Program". Los formatos son MIRAGE y PC, y sus versiones comprimidas con powerpacker, MIRAGE_PP y PC_PP.

1.35 Gadget RUN

RUN.- Arranca el Spectrum. En el modo elegido en el menu 'Options'.

1.36 Gadget STOP

Este gadget se denomina 'STOP' en la versión Genérica, y 'Reset & RUN' en la versión especial AGA, cumpliendo una función diferente en cada caso, que aquí se explican:

Stop.- Para la emulación, como si se hubiese pulsado HELP.

Reset & RUN.- Arranca la emulación (como el gadget RUN) pero forzando un RESET del Spectrum.

1.37 Gadget Run ARexx

Run ARexx.- Abre un requester de ficheros para elegir el script ARexx a ejecutar. Se abrirá en el path fijado por la ToolType REXXPATH. Solo se mostrarán los ficheros con la extensión '.zxam'.

1.38 Gadget Abort ARexx

Abort ARexx.- Es un gadget que está totalmente bajo el control del script en cuanto a activarlo/desactivarlo y leerlo periódicamente. El emulador simplemente informa al script de que ha sido pulsado y es éste el que debe decidir lo que debe hacer.

1.39 String Dirección

Dirección del Poke.- Aquí debes meter la dirección donde quieres hacer el POKE.

1.40 String Dato

Dato para Poke.- Aquí se inserta el dato que quieres 'Pokear'.

1.41 Texto Fichero Cargado

Aquí se muestra el nombre snapshot actualmente en memoria.

1.42 Texto Formato

Aquí se muestra el formato del snapshot actualmente en memoria, o sea, el formato en el que se encuentra grabado en disco.

1.43 Opciones de la emulación

La barra de menús consiste en dos menús desplegables, uno es el 'Project', que contiene las entradas 'Load program', 'Save program', 'Run', 'About...' y 'Quit', y el otro es el 'Options', que permite controlar muchas de las características de la emulación. Aquí tienes el menú 'Options' desglosado:

Load mode	Permite cambiar el modo de carga.
Disk	Cargar desde disco los .header y .bytes.
Tape	Cargar de cinta a través del parche de la ROM (necesita una CIA)
ARexx	Ejecutar el script ZXAM_TAPE.zxam (ver sección ARexx).

(El modo 'Tape' solo está disponible en modos SIN multitarea)

Save mode	Permite cambiar el modo de grabación.
Disk	Salvar a disco los .header y .bytes.
Disk Crunched	Salvar a disco, comprimiendo con powerpacker.library.
Tape	Salvar hacia cinta (necesita 'Audio -> CIA Sync').
ARexx	Ejecutar el script ZXAM_TAPE.zxam (ver sección ARexx).

(El modo 'Tape' solo está disponible en modos SIN multitarea)

Audio	Modos del sonido 48k.
Off	Desactiva completamente el sonido 48k.
Fast	Sonido a velocidad máxima.
CIA Sync	Sonido estabilizado por CIA.

(El modo CIA Sync debe estar activado para que funcione correctamente el Transfer y el Save a cinta)

AY-3-8912	Modos del chip de sonido 128k.
Off	Desactiva completamente el chip de sonido.
On	Activa el chip de sonido.
Smart	Modo 'inteligente' en el que se reorganizan dinámicamente los canales del chip para conseguir una mezcla mas fiel del canal de ruido blanco (solo necesario cuando se escucha el audio del Amiga a través de un sistema stereo). Opcional en previsión de posibles problemas.

Speed	Estabilizador de velocidad.
Fast	Máximo de velocidad posible (consume el 100% de CPU disponible).
Auto	Estabilizar velocidad, consumo solo la CPU necesaria.

(En multitarea se usa el timer.device para temporizar, pero en modos NO multitarea se necesita alojar una CIA para las temporizaciones)

(El modo recomendado para TODOS los ordenadores es el 'Auto', ya que solo ralentizará la emulación cuando el ordenador es demasiado rápido, pero nunca la ralentizará cuando el ordenador es demasiado lento. Esta es la razón de que este modo se llame 'Automático')

Joystick	Tipo de Joystick emulado por el de Amiga.
Kempston	Tipo kempston (disponible en la mayoría de juegos).
Sinclair Right	Tipo Sinclair (Interface II) lado derecho (teclas 6 a 0).
Sinclair Left	Tipo Sinclair lado izquierdo (teclas 1 a 5).

Cursor Tipo Cursor (también conocido como AGF y PROTEK).

(Esta opción NO desactiva la emulación Kempston, si no que añade emulaciones simultaneas. Por ejemplo, en modo Sinclair el joystick puede leerse como Sinclair y como Kempston desde el Spectrum)

(Opciones referentes a multitarea, no disponibles en la versión AGA)

TaskPri Prioridad de la emulación en multitarea.

+1
0
-1

Display Modo de visualización de la emulación.

Screen Pantalla legal, multitarea.

Window Ventana sobre la pantalla pública, multitarea.

No System (Screen) Pantalla legal, NO multitarea.

No System (PAL) Copperlist especial, NO multitarea. Este es el modo que usaban las versiones anteriores del ZXAM.

(Ver la sección

Bugs conocidos

para problemas de los modos

'No System (Screen)' y 'Window')

Refresh Intervalo entre redibujados de la imagen de la ventana.

1 Frame (solo para máquinas muy potentes)

2 Frames (movimiento suave, necesita potencia)

4 Frames (el recomendable para uso general y máquinas de potencia media)

8 Frames (máquinas lentas o imágenes muy estáticas, como el BASIC)

NOTA: Los modos que requieren usar una CIA para funcionar son 'Load Mode -> Tape', 'Audio -> CIA Sync' y 'Speed -> Auto' (este último solo en modos NO multitarea). Estos modos no necesitan una CIA para cada uno, si no que comparten la misma CIA. Esto significa que una copia del emulador en marcha alojará, como máximo, una CIA.

1.44 Las entrañables teclas de chicle...

Dentro de la emulación el teclado del Amiga equivale al de un Spectrum 48k, aunque hay configuradas algunas teclas especiales:

<- Borra un caracter (Shift+0)

Ctrl Equivale a pulsar Caps+Symbol, o sea, pasa a modo E (extendido)

Alt Equivale a Symbol Shift

Del Equivale a Shift+1 (Edit)

Tab Equivale a Shift+9 (Graph)

Help Para la emulación

Esc Reset

F1 Conmuta entre modo color o blanco y negro. En ordenadores sin memoria FAST el conflicto existente entre el procesador y los chips AGA para acceder a la memoria CHIP frena el procesador. Si necesitas el máximo de velocidad puedes poner el emulador en blanco y negro, con lo que se acelerará un 10%, aproximadamente. Además el sonido irá MUCHO mejor. SOLO VERSION AGA.

F2 En modo Blanco y Negro invierte la imagen. SOLO VERSION AGA.
 F6 Arranca el transfer (el programa se graba en cinta listo para ser cargado en un Spectrum real o en el propio emulador). Solo en modos NO multitarea.
 Cursores Equivalen a los cursores del Spectrum original (Shift+5 6 7 8).
 , (tecla a la derecha de la M) equivale a la coma (Symbol+N)
 . (2ª tecla a la derecha de la M) equivale al punto (Symbol+M)
 ' (3ª tecla a la derecha de la M) equivale a las comillas (Symbol+P)
 Ñ (tecla a la derecha de la L) equivale a dos puntos (Symbol+Z)
 ; (2ª tecla a la derecha de la L) equivale al punto y coma (Symbol+O)

El keypad funciona completo.

El joystick se emula con el de Amiga, comportandose según lo especificado en el menú 'Options'.

Debo decir que seguramente más de un usuario de Amiga 1200 notará algo extraño en el teclado, y es que no se pueden pulsar al mismo tiempo 2 teclas en la misma linea. Esto significa que si pulsas, por ejemplo, la P y la Q al mismo tiempo el emulador creará que no tienes pulsada ninguna tecla. Este comportamiento no se debe a un fallo en el emulador si no a una "curiosidad" bastante molesta del teclado del Amiga 1200 (al menos del alemán, que es el que yo tengo).

En modos multitarea el teclado/joystick se leen solo cuando la ventana/pantalla del spectrum está seleccionada.

1.45 Funciones ARexx incorporadas al ZXAM

A partir de la versión 1.4 β el emulador tiene un puerto ARexx. ←

Este puerto

es ideal para crear scripts que permitan soportar nuevos formatos, ensamblar, desensamblar, hacer pokeadores, y cualquier cosa que se te ocurra. El puerto tiene el nombre 'ZXAM_REXX', aunque solo usarás el nombre para comprobar si el emulador está presente, como verás más abajo.

Junto con el emulador encontrarás scripts de ejemplo para cargar formatos de snapshot, para grabar la pantalla del Spectrum como IFF, para conversión de formatos en bloque, para listar el programa BASIC que pueda haber en la memoria del Spectrum, para sacar la ROM del Spectrum del emulador, por si quieres modificarla por alguna razón, para desensamblar la memoria del spectrum y para descomponer ficheros .TAP. La mayoría de estos scripts son simples ejemplos de uso de las funciones que incorpora el ZXAM. Una muestra de las posibilidades del puerto ARexx es el script Monitor.zxam (hecho por Leonardo Cocaña Galán), que es un simple desensamblador que te permite 'hurgar' por las tripas del Spectrum. Es solo un principio, que mejorandolo puede llegar a ser un completo desensamblador/monitor de C/M. Algunos de los scripts usan algunos comandos del sistema operativo (como requestchoice o multiview). Es caso de no disponer de estos, deberás modificar los scripts para que usen los que tengas.

El puerto ARexx del emulador tiene una peculiaridad, y es que es un Function Host, no un Command Host. Traducido al 'cristiano' significa que lo que tiene el emulador no son comandos, sino funciones, que se añaden a las del ARexx como si de una librería se tratara. De esta forma no es necesario hacer ADDRESS ZXAM_REXX o OPTIONS RESULTS al principio de los scripts. Además,

así se pueden usar las funciones directamente dentro de expresiones. La única regla es que los parámetros se entregan entre paréntesis, y que las funciones sin parámetros necesitan un paréntesis vacío tras ellas. Además, al ser un function host, todas las funciones del emulador empiezan por ZXAM, para evitar conflictos con nombres de funciones de otros host o librerías. Los nombres son independientes de mayúsculas/minúsculas. Las funciones sin parámetros se pueden ejecutar como comandos (ADDRESS...).

NOTA IMPORTANTE: Ahora que el emulador puede funcionar en multitarea, varias de las funciones ARexx (como acceder a registros o leer/grabar snapshots) solo deben ejecutarse con la emulación parada (de no hacerlo así los resultados devueltos serían totalmente incorrectos). Para evitar esto, estas funciones devolverán un error si se las intenta ejecutar con la emulación en marcha. Por eso hay que tener la precaución de parar la emulación (con ZXAMStop()) al principio de los scripts que así lo requieran (si en necesario se puede almacenar el estado inicial con ZXAMActRun() por si al salir se quiere dejar el emulador en el mismo estado que estaba al entrar).

En la lista de funciones están marcadas con (*) las funciones que NECESITAN la emulación parada, y con (+) las funciones que NO aparecen en la versión AGA.

Las 93 funciones implementadas en la versión 2.0 (37.24 y 39.24) son:

Disponibles a partir de la versión 1.4 B (39.14):

*
ZXAMGetReg
*
ZXAMSetReg

ZXAMPeek
*
ZXAMPoke

ZXAMDPeek
*
ZXAMDPoke

ZXAMGetMem
*
ZXAMPutMem

ZXAMFindByte

ZXAMppLoadFile

ZXAMppSaveFile

ZXAMRun

ZXAMQuit

ZXAMNameFormat

ZXAMClearNameFormat

ZXAMLoadRequester

ZXAMSaveRequester

ZXAMActFormat

ZXAMActName

ZXAMActLoadPath

ZXAMActSavePath

ZXAMSaveFormat

Disponibles a partir de la versión 1.5 (39.17):

ZXAMActSaveFormat

ZXAMProgVersion

*

ZXAMParseLoaded

*

ZXAMParseToSave

ZXAMJoinPathName

ZXAMGetScr

ZXAMLoadPath

ZXAMSavePath

ZXAMPathPart

ZXAMFilePart

ZXAMBASICToken

ZXAMActPreRun

ZXAMActPostRun

ZXAMPreRun

ZXAMPostRun

ZXAMActPattern

ZXAMPattern

ZXAMReadAbort

ZXAMClearAbort

ZXAMNoReload

ZXAMEnableAbort

ZXAMDisableAbort

Disponibles a partir de la versión 1.6 (39.20):

ZXAMDisassemble

Disponibles a partir de la versión 1.6b (39.21):

ZXAMLoadMode

ZXAMSaveMode

ZXAMActLoadMode

ZXAMActSaveMode

ZXAMAbout

ZXAMFindBlock

ZXAMDoChecksum

*

ZXAMBreakPoint

*

ZXAMTrace

*

ZXAMDoInt

*

ZXAMBlockOfRegs

Disponibles a partir de la versión 2.0 (37.24 y 39.24):

ZXAMAudio

ZXAMAYEmulation

ZXAMSpeed

ZXAMJoystick

+

ZXAMTaskPri

+

ZXAMDisplay

+

ZXAMRefresh

ZXAMActAudio

ZXAMActAYEmulation

ZXAMActSpeed

ZXAMActJoystick

+

ZXAMActTaskPri

+

ZXAMActDisplay

```
+
ZXAMActRefresh

ZXAMEmulToFront

ZXAMControlToFront

ZXAMActRun
*
ZXAMAYBlockOfRegs
*
ZXAMWriteROM

ZXAMCheckSumString

ZXAMCompareString

ZXAMStop
Scripts especiales:
```

ZXAM_INIT.zxam Este script se ejecuta al arrancar el ZXAM. Mas concretamente, se ejecuta tras mostrar el requester de About en el arranque.

ZXAM_EXIT.zxam Este script se ejecuta al salir del ZXAM, justo antes de cerrar la ventana.

ZXAM_TAPE.zxam Este script es llamado desde los modos 'Load Mode -> ARexx' y 'Save Mode ->ARexx'. Por ejemplo, cuando está seleccionado el modo 'Load Mode -> ARexx' y desde el BASIC del Spectrum se hace 'LOAD ""' este script se ejecuta, para suministrar al emulador con los bloques de datos que requiere. El script que se incluye con el emulador permite cargar desde ficheros .TAP de forma automatizada, sin necesidad de descomponerlos en los correspondientes .header y .bytes (como habia que hacer en versiones anteriores).

El entorno en que es ejecutado el script es muy específico. Al entrar, el script debe leer (con la función ZXAMTapeAction()) la acción requerida (0=nada, 1=cargar, 2=verificar y 3 =salvar). El script debe decidir lo que debe hacer según la acción especificada. Si la acción es 0 se deberá a que se ha arrancado externamente el script (desde CLI, o desde el gadget 'Run ARexx'), por lo que el script debe avisar de que no es un script de ejecución manual, y volver. Para este script hay una serie de funciones específicas, que se detallan a continuación:

```
ZXAMTapeAction

ZXAMTapeLoadName

ZXAMTapeSaveName

ZXAMTapeAddress

ZXAMTapeSize

ZXAMTapeFlag

ZXAMTapeLoadOffset
```

ZXAMTapeSaveOffset

ZXAMTapeSetResult

ZXAMTapeMark

ZXAMTapeSetLoadName

ZXAMTapeSetSaveName

ZXAMTapeSetLoadOffset

ZXAMTapeSetSaveOffset

ZXAMTapeSetMark

Al volver al emulador, el script debe usar la función ↵

ZXAMTapeSetResult para

informar de los resultados de la operación al ZXAM. El primer parámetro de esta función es un numero que informa del éxito o fracaso: 2 significa error, 1 significa éxito y 0 significa que el script no sabe que hacer, en cuyo caso se abrirá el habitual requester de ficheros de la opción 'Load/Save Mode -> Disk' para cargar/grabar los .header o .bytes habituales. Cuando se comunica éxito (1) debe entregarse un segundo parámetro, que es el path+nombre del fichero resultante, que será el que se entregará a la rutina de carga del spectrum. En caso de ser un SAVE no hace falta entregar ese parámetro.

En principio parece un poco confuso, pero si miras el script ZXAM_TAPE.zxam que viene junto al emulador podrás comprender mejor el funcionamiento de este sistema, y la potencia que tiene. En principio el script está preparado para cargar y grabar automáticamente ficheros .TAP. Con un poco de paciencia se puede preparar para leer ficheros .header y .bytes separados automáticamente, o para cargar desde dentro de un archivo .LHA una serie de .header y .bytes, como si de un .TAP comprimido se tratara.

Un detalle sobre este script es que se puede ejecutar en cualquier modo de emulación (con y sin multitarea), pero que en modo NOSYSTEMPAL (o sea, modo no multitarea y con copperlist especial, no 'legal') al tener que volver al sistema para cargar un bloque, luego volver a la emulación, otra vez al sistema, etc... puede ser bastante molesto ver constantemente la pantalla del spectrum apareciendo y desapareciendo. Para este tipo de operacion de carga se recomienda usar modos multitarea o de pantalla legal (NOSYSTEMSCREEN), ya que en estos modos se tiene la pantalla del Spectrum constantemente al frente. En la versión AGA no hay solución para esto ya que el único modo del que dispone es 'no legal'.

1.46 ZXAMGetReg

ZXAMGetReg: Toma el valor de un registro del Z80 de 8 o 16 bits. ↵

El parámetro

opcional 'formato' fija el formato en que debe darse el valor devuelto ('h' para hexadecimal y 'd' para decimal). El formato por defecto es decimal.

```
formato:      valor = ZXAMGetReg(
                registro
```

, [formato])

1.47 ZXAMSetReg

ZXAMSetReg: Mete un valor de 8 o 16 bits en un registro del Z80.

formato: ZXAMSetReg(
registro
, valor)

1.48 ZXAMPeek

ZXAMPeek: Lee un byte de la dirección dada de la memoria del Spectrum.

formato: valor = ZXAMPeek(direccion spectrum)

1.49 ZXAMPoke

ZXAMPoke: Escribe un byte en la memoria del Spectrum.

formato: ZXAMPoke(direccion spectrum , valor 8 bits)

1.50 ZXAMDPeek

ZXAMDPeek: Lee un valor de 16 bits de la memoria del Spectrum en formato Z80, o sea, el byte menos significativo está primero.

formato: valor = ZXAMDPeek(direccion spectrum)

1.51 ZXAMDPoke

ZXAMDPoke: Escribe un valor de 16 bits en la memoria del Spectrum. Lo escribe en formato Z80, o sea, el byte menos significativo se coloca primero.

formato: ZXAMDPoke(direccion spectrum , valor 16 bits)

1.52 ZXAMGetMem

ZXAMGetMem: Lee un trozo de memoria de la dirección y el tamaño dados y lo entrega como cadena.

formato: cadena = ZXAMGetMem(direccion spectrum , tamaño)

1.53 ZXAMPutMem

ZXAMPutMem: Coloca la cadena dada en la dirección especificada de la memoria de Spectrum. La cadena suele ser una secuencia de bytes, no un texto.

formato: ZXAMPutMem(direccion spectrum , cadena)

1.54 ZXAMFindByte

ZXAMFindByte: Busca el byte a partir de la dirección dada. Si llega al tope superior de la memoria del Spectrum sin encontrarlo, entregará -1. En caso de encontrarlo, entrega la dirección donde se encuentra el byte.

formato: direccion = ZXAMFindByte(direccion spectrum , byte_a_buscar)

1.55 ZXAMRun

ZXAMRun: Arranca la emulación (simple, ¿no?).

formato: ZXAMRun()

1.56 ZXAMQuit

ZXAMQuit: Provoca la salida del emulador (como cerrar la ventana del emulador).

formato: ZXAMQuit()

1.57 ZXAMppLoadFile

ZXAMppLoadFile: Carga un fichero a través de la powerpacker.library y lo entrega descomprimido en forma de cadena. Si el fichero no estaba comprimido, se cargará igual.

formato: bloque = ZXAMppLoadFile(path_y_nombre)

1.58 ZXAMppSaveFile

ZXAMppSaveFile: Salva un bloque de bytes a disco, comprimiéndolo con la powerpacker.library. En caso de no querer comprimir el fichero, usar los comandos del ARexx para acceso a ficheros.

formato: ZXAMppSaveFile(path_y_nombre , bloque)

1.59 ZXAMLoadRequester

ZXAMLoadRequester: Abre el requester de LOAD del emulador. El título es para colocar en la parte superior del requester. El path (opcional) fija en que directorio debe abrirse el requester. En caso de no entregar path se usará el último que se usó con el requester. Si 'path_y_nombre' está vacío significa que se ha pulsado el CANCEL.

formato: path_y_nombre = ZXAMLoadRequester(titulo , [path])

1.60 ZXAMSaveRequester

ZXAMSaveRequester: Abre el requester de SAVE del emulador. El título es para colocar en la parte superior del requester. El path (opcional) fija en que directorio debe abrirse el requester. En caso de no entregar path se usará el último que se usó con el requester. Si 'nombre_y_path' está vacío significa que se ha pulsado el CANCEL.

formato: path_y_nombre = ZXAMSaveRequester(titulo , [path])

1.61 ZXAMNameFormat

ZXAMNameFormat: Coloca las cadenas entregadas en las casillas 'Loaded File' y 'Format' de la ventana del emulador.

formato: ZXAMNameFormat(nombre , formato)

1.62 ZXAMActName

ZXAMActName: Entrega el nombre contenido en la casilla 'Loaded file'. Si no hay nombre, entregará una cadena vacía.

formato: nombre = ZXAMActName()

1.63 ZXAMActLoadPath

ZXAMActLoadPath: Entrega el último path utilizado en el requester de LOAD.

formato: path = ZXAMActLoadPath()

1.64 ZXAMActSavePath

ZXAMActSavePath: Entrega el último path utilizado en el requester de SAVE.

formato: path = ZXAMActSavePath()

1.65 ZXAMActFormat

ZXAMActFormat: Entrega el nombre de formato contenido en la casilla 'Format'. Si no hay formato, entregará una cadena vacía.

```
formato:          formato = ZXAMActFormat()
```

1.66 ZXAMActSaveFormat

ZXAMActSaveFormat: Entrega el nombre del actual formato seleccionado para save. Puede ser PC, MIRAGE, PC_PP y MIRAGE_PP (los dos últimos son los comprimidos con powerpacker.library).

```
formato:          formato = ZXAMActSaveFormat()
```

1.67 ZXAMProgVersion

ZXAMProgVersion: Entrega la versión interna del programa. Sirve por si se quiere usar una función ARexx que solo está presente a partir de cierta versión del emulador. Para saber la versión a la que aparece una función, mira la HISTORIA.

```
formato:          version = ZXAMProgVersion()
```

1.68 ZXAMSaveFormat

ZXAMSaveFormat: Selecciona un formato para SAVE. Puede ser PC, MIRAGE, PC_PP o MIRAGE_PP.

```
formato:          ZXAMSaveFormat( nombreformato )
```

1.69 ZXAMParseLoaded

ZXAMParseLoaded: Coge el bloque entregado e intenta reconocerlo como snapshot. Si es de formato conocido (PC, MIRAGE, Z80 o KGB, SIN comprimir) colocará los registros y la memoria a los valores de acuerdo al snapshot. A la salida entrega el nombre del formato del bloque (PC, MIRAGE, Z80 o KGB). Si no ha podido reconocer el formato no modificará nada (ni registros ni memoria) y entregará una cadena vacía como nombre de formato.

```
formato:          formato = ZXAMParseLoaded( bloque )
```

1.70 ZXAMParseToSave

ZXAMParseToSave: Entrega la memoria y los registros del Spectrum colocados según el formato seleccionado para SAVE (SIN comprimir).

formato: bloque = ZXAMParseToSave()

1.71 ZXAMJoinPathName

ZXAMJoinPathName: Une el path y el nombre entregados.

formato: nombrecompleto = ZXAMJoinPathName(path , nombre)

1.72 ZXAMGetScr

ZXAMGetScr: Entrega la pantalla actual del Spectrum en formato de 4 bitplanos entrelazados, listo para grabar en una estructura IFF ILBM.

formato: bloque = ZXAMGetScr()

1.73 ZXAMClearNameFormat

ZXAMClearNameFormat: Borra las casillas 'Loaded file' y 'Format'.

formato: ZXAMClearNameFormat()

1.74 ZXAMLoadPath

ZXAMLoadPath: Modifica el path para requester de LOAD sin abrirlo.

formato: ZXAMLoadPath(path)

1.75 ZXAMSavePath

ZXAMSavePath: Modifica el path para requester de SAVE sin abrirlo.

formato: ZXAMSavePath(path)

1.76 ZXAMFilePart

ZXAMFilePart: Entrega el nombre del fichero de un path completo.

formato: nombre = ZXAMFilePart(path_y_nombre)

1.77 ZXAMPathPart

ZXAMPathPart: Entrega el path (sin el nombre del fichero) de un path completo.

formato: path = ZXAMPathPart(path_y_nombre)

1.78 ZXAMPreRun

ZXAMPreRun: Modifica la linea de comandos PRERUN.

formato: ZXAMPreRun(linea_de_comandos)

1.79 ZXAMActPreRun

ZXAMActPreRun: Entrega la actual linea de comandos PRERUN.

formato: linea_de_comandos = ZXAMActPreRun()

1.80 ZXAMPostRun

ZXAMPostRun: Modifica la linea de comandos POSTRUN.

formato: ZXAMPostRun(linea_de_comandos)

1.81 ZXAMActPostRun

ZXAMActPostRun: Entrega la actual linea de comandos POSTRUN.

formato: linea_de_comandos = ZXAMActPostRun()

1.82 ZXAMPattern

ZXAMPattern: Modifica el pattern para los requesters de ficheros.

formato: ZXAMPattern(pattern)

1.83 ZXAMActPattern

ZXAMActPattern: Entrega el pattern actual para los requesters de ficheros.

formato: pattern = ZXAMActPattern()

1.84 ZXAMBASICToken

ZXAMBASICToken: Entrega el token expandido correspondiente al caracter entregado, o sea, traduce el caracter según la tabla de caracteres y palabras claves (tokens) del Spectrum. Por ejemplo ZXAMBASICToken('ff'x) entregará 'COPY'. Todos los caracteres del 0 al 255 son traducidos.

formato: cadena = ZXAMBASICToken(caracter)

1.85 ZXAMReadAbort

ZXAMReadAbort: Entrega un 0 o un 1, indicando si se ha pulsado el gadget 'Abort ARexx'.

formato: pulsado = ZXAMReadAbort()

1.86 ZXAMClearAbort

ZXAMClearAbort: Borra (a 0) el indicador de que se ha pulsado 'Abort ARexx'.

formato: ZXAMClearAbort()

1.87 ZXAMNoReload

ZXAMNoReload: Desactiva el gadget "Reload". Esto debe hacerse cuando hemos dejado modificada la caja 'Loaded File', indicando así que al pulsar 'Reload' no se cargará lo que aparece en dicha caja.

formato: ZXAMNoReload()

1.88 ZXAMEnableAbort

ZXAMEnableAbort: Activa el gadget 'Abort ARexx'. Esto debe hacerse si quieres leer el gadget desde el script (al arrancar el script el gadget está desactivado por defecto).

formato: ZXAMEnableAbort()

1.89 ZXAMDisableAbort

ZXAMDisableAbort: Desactiva el gadget 'Abort ARexx'. Si no vas a leer el gadget debes desactivarlo, para evitar que el usuario intente interrumpir el script inutilmente.

formato: ZXAMDisableAbort()

1.90 ZXAMDisassemble

ZXAMDisassemble: Desensambla la instrucción de la dirección dada, entregando a la salida una cadena con el formato:

- 1 caracter ASCII que indica el numero de bytes que ocupa la instrucción (para incrementar la dirección entregada y saber así la dirección de la siguiente). Este caracter debe ser eliminado de la cadena antes de imprimirla en pantalla o impresora.
- el resto de la cadena es un desensamblado en el habitual formato 'dirección+código objeto+mnemónico'. En el caso de que la instrucción desensamblada fuera un salto relativo (JR, DJNZ, etc...) se añadirá al final de la línea un comentario que indicará la dirección destino del salto. Las instrucciones ilegales se representarán como '---'.

El parametro opcional "formato" determina el formato en que se imprimen las direcciones/datos. Si es 'H' se imprimirán en hexadecimal, y si es 'D' se imprimirán en decimal. Si no se entrega este parámetro, se imprimirá en hexadecimal

formato: ZXAMDisassemble(direccion , [formato])

1.91 ZXAMLoadMode

ZXAMLoadMode: Indica el modo del parche LOAD (o sea, controla el menu LOAD

MODE). Se le entrega el modo deseado ('TAPE', 'DISK' o 'AREXX').

formato: ZXAMLoadMode(modo)

1.92 ZXAMSaveMode

ZXAMSaveMode: Indica el modo del parche SAVE (o sea, controla el menu SAVE

MODE). Se le entrega el modo deseado ('TAPE', 'DISK', 'DISK_PP' o 'AREXX').

formato: ZXAMSaveMode(modo)

1.93 ZXAMActLoadMode

ZXAMActLoadMode: Indica el modo en el que está actualmente el parche LOAD (o

sea, muestra el estado del
 menu
 LOAD MODE). El resultado es una cadena ('TAPE'
 'DISK' o 'AREXX').

formato: modo = ZXAMActLoadMode()

1.94 ZXAMActSaveMode

ZXAMActSaveMode: Indica el modo en el que está actualmente el ←
 parche SAVE (o
 sea, muestra el estado del
 menu
 SAVE MODE). El resultado es una cadena ('TAPE',
 'DISK', 'DISK_PP' o 'AREXX').

formato: modo = ZXAMActSaveMode()

1.95 ZXAMAbout

ZXAMAbout: Muestra el requester de 'About...' del emulador.

formato: ZXAMAbout()

1.96 ZXAMFindBlock

ZXAMFindBlock: Busca un bloque de bytes en la memoria del Spectrum. El primer argumento es la dirección donde empezar la búsqueda, y el segundo es el conjunto de bytes (en forma de cadena) a buscar. Esto significa que para buscar 'HOLA' debemos hacer ZXAMFindBlock(0,'HOLA'), y para buscar los bytes 0F 45 7B haremos ZXAMFindBlock(0,'0F457B'x). En caso de que el número hexadecimal esté en una variable, usar x2c(variable) en el segundo argumento, para que el AREXX convierta la cadena hexadecimal en los bytes correspondientes. El resultado es la dirección donde esa secuencia de bytes ha sido encontrada, o -1 si no lo ha sido.

formato: dirección = ZXAMFindBlock(dirección , bloque_a_buscar)

1.97 ZXAMDoChecksum

ZXAMDoChecksum: Hace una suma de comprobación como la que hace la rutina SAVE de la ROM, o sea, un XOR (OR exclusiva) de todos los bytes del bloque de datos. Los parámetros son la dirección de inicio del bloque y el número de bytes que lo componen. Por ejemplo, ZXAMDoChecksum(0,16384) entregará el resultado de hacer XOR de todos los bytes de la ROM.

formato: resultado = ZXAMDoChecksum(dirección , numero_bytes)

1.98 ZXAMBreakPoint

ZXAMBreakPoint: Coloca un breakpoint (punto de ruptura) en la dirección dada de la memoria del Spectrum. Si durante la ejecución se llega al breakpoint el emulador sale al workbench (como si se pulsase la tecla 'HELP'). En memoria solo puede haber UN breakpoint, por lo que al poner uno se elimina el anterior. Si la dirección es 0 se borra el breakpoint anterior sin colocar ninguno nuevo. El breakpoint es una instrucción, por lo que si el programa modifica esa parte del código desaparece el breakpoint. El breakpoint solo "sobrevive" a una entrada al emulador, o sea, tan pronto como se pulse 'HELP' para salir de la emulación, éste desaparecerá (las salidas provocadas por load/save de disco no afectan al breakpoint).

formato: ZXAMBreakPoint(dirección)

1.99 ZXAMTrace

ZXAMTrace: Ejecuta la instrucción a la que apunta PC (sin desconectar la multitarea). Una vez ejecutada la instrucción la función devuelve el desensamblado de la instrucción ejecutada (igual que ZXAMDisassemble()). El 'formato' puede ser 'd' o 'h', indicando si el desensamblado debe ser decimal o hexadecimal (como el segundo parámetro de ZXAMDisassemble()). El 'formato' es opcional, siendo hexadecimal por defecto. Al contrario que para arrancar el emulador, ni las CIAs ni los canales de audio necesitan ser alojados para ejecutar código paso-a-paso.

formato: desen = ZXAMTrace([formato])

1.100 ZXAMDoInt

ZXAMDoInt: Efectua una interrupción (como si hubiese llegado la señal INT al microprocesador Z80), o sea, mete PC en la pila del Z80, y le da a PC el valor adecuado al modo de interrupción actual (en IM1 le da el valor \$38, y en IM2 mete el vector de la tabla apuntada por I). Si las interrupciones no están permitidas, la función no hace nada.

formato: ZXAMDoInt()

1.101 ZXAMBlockOfRegs

ZXAMBlockOfRegs: Entrega un bloque de 27 bytes que contiene todos los registros y estados internos del procesador Z80, en el siguiente orden:

Pos Tamaño Contenido

0	1	A
1	1	F
2	2	BC
4	2	DE
6	2	HL
8	1	A'
9	1	F'
10	2	BC'
12	2	DE'
14	2	HL'
16	2	IX
18	2	IY
20	2	SP
22	2	PC
24	1	Estado Interrupción (0=no permitidas / 1=permitidas)
25	1	Modo de Interrupción (0, 1 o 2)
26	1	Color del Borde (0 a 7)

Los registros dobles están en formato 68000, o sea, con el byte más significativo PRIMERO.

formato: bloque = ZXAMBlockOfRegs()

1.102 Registros

REGISTROS: los registros accesibles por ZXAMGetReg y ZXAMSetReg son los habituales del Z80, en el formato:

-registros de 8 bits: su nombre directo, incluyendo las mitades de los registros de 16 bits no divisibles, o sea, SPH y SPL son las dos mitades del registro SP de 16 bits.

-registros de 16 bits: su nombre directo, como HL y PC. AF no está implementado, o sea que hay que leer A y F por separado.

-registros de 16 bits: su nombre directo, como HL y PC. AF no está implementado, o sea que hay que leer A y F por separado.

-registros de 16 bits invertidos: si se invierte el nombre de un registro de 16 bits (por ejemplo LH o XI) la lectura/escritura se realiza intercambiando los bytes alto y bajo del registro. En el script 'LoadSnap.zxam' verás un ejemplo de para que sirve.

-"registros" especiales: son

 INT : estado de las interrupciones (0=OFF, 1=ON)

 IM : modo de interrupción (0, 1 o 2)

 BOR : color actual del borde (0 a 7)

-registros del chip de sonido AY-3-8912: los 16 registros del chip de sonido del 128k se pueden acceder con los nombres AY0, AY1.... ...AYE y AYP (como verás, el ultimo caracter del nombre es el numero hexadecimal que indica el registro, de 0 a 15). Hay un registro adicional, el AYP, que contiene el último byte enviado al puerto \$FFFD, que es el usado por el spectrum para direccionar los registros del AY.

en todos los casos se le añade un '2' al final del nombre para acceder a los registros del juego alternativo, o sea, para acceder a hl' y a a' escribiremos HL2 y A2. Los nombres son independientes de mayúsculas/minúsculas. Todos los casos admiten lectura y escritura.

1.103 ZXAMAudio

ZXAMAudio: Fija el estado del menu 'Options -> Audio'. Los valores posibles son OFF, FAST y CIASYNC.

formato: ZXAMAudio(nuevo estado)

1.104 ZXAMAYEmulation

ZXAMAYEmulation: Fija el estado del menu 'Options -> AY-3-8912'. Los valores posibles son OFF, ON y SMART.

formato: ZXAMAYEmulation(nuevo estado)

1.105 ZXAMSpeed

ZXAMSpeed: Fija el estado del menu 'Options -> Speed'. Los valores posibles son FAST y AUTO.

formato: ZXAMSpeed(nuevo estado)

1.106 ZXAMJoystick

ZXAMJoystick: Fija el estado del menu 'Options -> Joystick'. Los valores posibles son KEMPSTON, SINCLAIRLEFT, SINCLAIRRIGHT y CURSOR.

formato: ZXAMJoystick(nuevo estado)

1.107 ZXAMTaskPri

```
ZXAMTaskPri: Fija el estado del
menu
'Options -> TaskPri'.
```

Los valores posibles son '+1', '0' y '-1'.

formato: ZXAMTaskPri(nuevo estado)

1.108 ZXAMDisplay

```
ZXAMDisplay: Fija el estado del
menu
'Options -> Display'.
```

Los valores posibles son SCREEN, WINDOW, NOSYSTEMSCREEN y NOSYSTEMPAL.

formato: ZXAMDisplay(nuevo estado)

1.109 ZXAMRefresh

```
ZXAMRefresh: Fija el estado del
menu
'Options -> Refresh'.
```

Los valores posibles son 1, 2, 4 y 8.

formato: ZXAMRefresh(nuevo estado)

1.110 ZXAMActAudio

```
ZXAMActAudio: Lee el estado del
menu
'Options -> Audio'.
```

Los valores posibles son OFF, FAST y CIASYNC.

formato: estado = ZXAMActAudio()

1.111 ZXAMActAYEmulation

```
ZXAMActAYEmulation: Lee el estado del
menu
'Options ->
```

AY-3-8912'. Los valores posibles son OFF, ON y SMART.

formato: estado = ZXAMActAYEmulation()

1.112 ZXAMActSpeed

ZXAMActSpeed: Lee el estado del
menu
'Options -> Speed'.

Los valores posibles son FAST y AUTO.

formato: estado = ZXAMActSpeed()

1.113 ZXAMActJoystick

ZXAMActJoystick: Lee el estado del
menu
'Options ->

Joystick'. Los valores posibles son KEMPSTON, SINCLAIRRIGHT, SINCLAIRLEFT y
CURSOR.

formato: estado = ZXAMActJoystick()

1.114 ZXAMActTaskPri

ZXAMActTaskPri: Lee el estado del
menu
'Options ->

TaskPri'. Los valores posibles son '+1', '0' y '-1'.

formato: estado = ZXAMActTaskPri()

1.115 ZXAMActDisplay

ZXAMActDisplay: Lee el estado del
menu
'Options ->

Display'. Los valores posibles son SCREEN, WINDOW, NOSYSTEMSCREEN y
NOSYSTEMPAL.

formato: estado = ZXAMActDisplay()

1.116 ZXAMActRefresh

ZXAMActRefresh: Lee el estado del menu
'Options ->

Refresh'. Los valores posibles son 1, 2, 4 y 8.

formato: estado = ZXAMActRefresh()

1.117 ZXAMEmulToFront

ZXAMEmulToFront: Pasa la pantalla de emulación al frente de todas las demás.

formato: ZXAMEmulToFront()

1.118 ZXAMControlToFront

ZXAMControlToFront: Pasa la pantalla donde se encuentra la ventana de control al frente de todas las demás.

formato: ZXAMControlToFront()

1.119 ZXAMActRun

ZXAMActRun: Lee el estado de la emulación, o sea, si está en marcha o parada. Devuelve un 0 si está parada y un 1 cuando está en marcha.

formato: estado = ZXAMActRun()

1.120 ZXAMAYBlockOfRegs

ZXAMAYBlockOfRegs: Devuelve en una cadena de 16 bytes los 16 registros del chip de sonido AY-3-8912, siendo el registro 0 el primero de la cadena.

formato: cadena = ZXAMAYBlockOfRegs()

1.121 ZXAMWriteROM

ZXAMWriteROM: Escribe una ROM nueva sobre la interna. Se le debe entregar una nueva ROM en forma de una cadena de 16384 bytes.

formato: ZXAMWriteROM(nueva ROM)

1.122 ZXAMChecksumString

ZXAMChecksumString: Devuelve el checksum de una cadena, o sea, el XOR de todos los bytes que contiene la cadena (es el tipode checksum usado por las rutinas LOAD/SAVE a cinta de la ROM del Spectrum).

formato: checksum = ZXAMChecksumString(cadena)

1.123 ZXAMCompareString

ZXAMCompareString: Compara una cadena con una zona de memoria del spectrum, e indica si son iguales o diferentes. Un 0 significa iguales, y un -1 diferentes.

formato: iguales = ZXAMCompareString(dirección , cadena)

1.124 ZXAMStop

ZXAMStop: Para la emulación.

formato: ZXAMStop()

1.125 ZXAMTapeAction

ZXAMTapeAction: Entrega un valor que indica la acción a realizar por parte del script ZXAM_TAPE.zxam. Un 0=nada, 1=load, 2=verify y 3=save.

formato: acción = ZXAMTapeAction()

1.126 ZXAMTapeLoadName

ZXAMTapeLoadName: Esta función entrega una cadena almacenada antes con ZXAMTapeSetLoadName. Estas funciones se usan simplemente para almacenar una cadena de hasta 255 caracteres para cualquier uso, dependiente del script. En concreto el script ARexx ZXAM_TAPE.zxam, que se incluye con el emulador, usa estas funciones para almacenar el path+nombre del fichero .TAP seleccionado para cargar. Hay que notar que cuando se hace RESET en el emulador, esta cadena es borrada, y esta función entrega una cadena vacia. En el script esto consigue que se 'des-seleccione' el fichero automáticamente, para que al hacer LOAD se abra el requester para elegir el .TAP.

formato: cadena = ZXAMTapeLoadName()

1.127 ZXAMTapeSaveName

ZXAMTapeSaveName: Es idéntica a `ZXAMTapeLoadName`, pero afecta a otra cadena separada de la anterior, y su función correspondiente es `ZXAMTapeSetSaveName`. Pensada para almacenar path+nombre del `.TAP` actualmente utilizado para `SAVE`.

formato: cadena = `ZXAMTapeSaveName()`

1.128 ZXAMTapeAddress

ZXAMTapeAddress: Esta función entrega la dirección, dentro de la memoria del Spectrum, del bloque entregado por la rutina `SAVE` al script `ZXAM_TAPE.zxam`. Así se puede extraer ese bloque para manipularlo y salvarlo de la forma que se desee. Cuando se llama al script para `LOAD` también se entrega la dirección del bloque a cargar en esta función.

formato: dirección = `ZXAMTapeAddress()`

1.129 ZXAMTapeSize

ZXAMTapeSize: Esta función entrega el tamaño del bloque entregado por la rutina `SAVE` al script `ZXAM_TAPE.zxam`, para extraer ese bloque para manipularlo y salvarlo de la forma que se desee. Cuando se llama al script para `LOAD` también se entrega el tamaño del bloque a cargar en esta función.

formato: tamaño = `ZXAMTapeSize()`

1.130 ZXAMTapeFlag

ZXAMTapeFlag: Esta función entrega el flag para el bloque entregado por la rutina `SAVE` al script `ZXAM_TAPE.zxam`. Cuando se llama al script para `LOAD` también se entrega el flag del bloque deseado para cargar en esta función.

formato: flag = `ZXAMTapeFlag()`

1.131 ZXAMTapeLoadOffset

ZXAMTapeLoadOffset: Esta función entrega un valor numérico almacenado antes con `ZXAMTapeSetLoadOffset`. Pensado para almacenar un offset para acceder a ficheros compuestos como los `.TAP`. Hay que notar que cuando se hace `RESET` en el emulador, este offset se pone a 0.

formato: offset = `ZXAMTapeLoadOffset()`

1.132 ZXAMTapeSaveOffset

ZXAMTapeSaveOffset: Esta función entrega un valor numérico almacenado antes con ZXAMTapeSetSaveOffset. Es idéntica a ZXAMTapeLoadOffset, pero el valor numérico se almacena separado de éste. Pensado para almacenar un offset para acceder a ficheros compuestos como los .TAP. Hay que notar que cuando se hace RESET en el emulador, este offset se pone a 0.

formato: offset = ZXAMTapeSaveOffset()

1.133 ZXAMTapeSetResult

ZXAMTapeSetResult: Esta función se usa para que el script ZXAM_TAPE.zxam indique al emulador, antes del volver, el resultado de la operación solicitada al script. El primer parámetro es un numero entre 0 y 2. 0 indica que la operación no correspondía al script, por ejemplo, si el script solo está hecho para LOAD, en caso de recibir un comando SAVE debe dar como resultado 0. Así el emulador abrirá el habitual requester de SAVE como si el menu 'Save mode -> Disk' estuviese seleccionado. El resultado 1 indica operación realizada con éxito, y el resultado 2 indica error en la operación, lo que provocará un BREAK en la rutina de LOAD/SAVE de la ROM del Spectrum.

El segundo parámetro SOLO debe incluirse cuando la operación solicitada era LOAD o VERIFY y el resultado es 1 (OK). Este parámetro es el path+nombre del fichero a cargar en la memoria del Spectrum (normalmente un fichero temporal extraído de un .TAP).

formato: ZXAMTapeSetResult(resultado , [nombre fichero])

1.134 ZXAMTapeMark

ZXAMTapeMark: Esta función entrega un valor numérico almacenado antes con ZXAMTapeSetMark. Previsto para usarlo para almacenar un offset sobre un fichero .TAP, y poder así 'rebobinar' a ese punto después.

formato: offset = ZXAMTapeMark()

1.135 ZXAMTapeSetLoadName

ZXAMTapeSetLoadName: Esta función se usa para almacenar una cadena, para recuperarla después con ZXAMTapeLoadName().

formato: ZXAMTapeSetLoadName(cadena)

1.136 ZXAMTapeSetSaveName

ZXAMTapeSetSaveName: Esta función se usa para almacenar una cadena, para recuperarla después con ZXAMTapeSaveName().

formato: ZXAMTapeSetSaveName(cadena)

1.137 ZXAMTapeSetLoadOffset

ZXAMTapeSetLoadOffset: Esta función se usa para almacenar un valor numérico, para recuperarlo después con ZXAMTapeLoadOffset().

formato: ZXAMTapeSetLoadOffset(numero)

1.138 ZXAMTapeSetSaveOffset

ZXAMTapeSetSaveOffset: Esta función se usa para almacenar un valor numérico, para recuperarlo después con ZXAMTapeSaveOffset().

formato: ZXAMTapeSetSaveOffset(numero)

1.139 ZXAMTapeSetMark

ZXAMTapeSetMark: Esta función se usa para almacenar un valor numérico, para recuperarlo después con ZXAMTapeMark().

formato: ZXAMTapeSetMark(numero)

1.140 Curiosidades...

Este programa ha sido escrito en ensamblador desde la primera línea de código a las 17750 (!!!!) actuales (151023 al expandir las MACROS). En mi Amiga 1200 con disco duro y 68030/28MHz el emulador tarda 51 seg en ensamblarse (antes eran 3 min 50 seg cuando no tenía disco duro, o 4 min 53 seg la última versión que ensamblé sobre el 500 Plus). La verdad es que se me ha ido un poco la mano...

Este es el tercer programa "completo" que hago en ensamblador. Empecé a programarlo sobre un Amiga 500 Plus con 2 Megas de RAM, 3 disqueteras y el ensamblador Devpac 3.02. En cuanto he podido comprarme un Amiga 1200 he convertido el emulador a código 68020 puro y chips AGA.

Lo empecé para aprender sobre todo en cuanto a código específico 68020, para explorar las posibilidades y ventajas de programar en código especial para este procesador. Era además un capricho de programador compulsivo (es que soy adicto a la programación). Cuando solo tenía el 500Plus programaba condicionalmente en código 68000 y 68020 al mismo tiempo y lo probaba en el Amiga 4000/030 de un amigo. Aprovechando mi reciente adquisición del Amiga 1200 también he investigado sobre las posibilidades de los chips AGA (que son muchas) y la

forma de utilizarlos.

Ahora al fin tengo un equipo en condiciones (A1200 420Mb HD, 68030/28MHz y 6 Mb de memoria). Ahora programar es un gustazo!!

1.141 Sobre esta versión....

Antes que nada quiero explicar que este ya NO es el Spectrum Emulator AGA. Esta versión es el 'ZXAM Spectrum Emulator v2.0' que incluye dos versiones principales, la 'Generic', que sirve para todos los modelos de Amiga, sea cual sea el tipo de chips que incorpora (incluso AGA), y la 'Special AGA', que aprovecha la potencia adicional de los chips AGA, a cambio de perder las posibilidades multitarea de la versión Generic.

Pero, ¿por que esta versión? ¿no tenia que ser la 128k la siguiente?

Supongo que alguno debe estar preguntándose esto....bueno, pues intentaré explicar la razón de ser de esta versión.

En principio no tenia previsto (en absoluto) implementar la multitarea en el emulador, por todos los problemas que ello suponía....la cuestión está en que tenia que reestructurar totalmente los listados del emulador para poder hacer una emulación 128k en condiciones, y al llegar al sistema gráfico de la versión ECS decidí modificarlo para poder trabajar sobre pantallas legales, y así solucionar el problema de la gente que usa monitores VGA con el Amiga. Tras solucionarlo llegué a la parte de la ejecución paso-a-paso. Resulta que cuando la incluí en la versión 1.6 tuve que preparar el Z80 para ejecutar instrucciones sin desactivar el sistema (ya que se ejecutan desde ARexx) y claro, se me presentó la combinación Z80 multiarea y pantalla 'legal' y no pude resistir la tentación....'solo' tenia que hacer unos ajustes en el sistema de sonido, el color del borde, hacer 'legal' el sistema de lectura del teclado....

Pues bueno, aquí está el resultado de todo eso. Al final la multitarea me ha costado MUCHO mas de lo que creía (¡4 meses!) aparte de que también he emulado el chip de sonido del 128k y he pasado los docs a guide (eso si que ha sido un palo...) pero ahora estoy contento de como me ha quedado esta versión. Y espero que a ti te guste también.

Aún me he quedado con las ganas de hacerle algunas cuantos arreglos mas a esta versión, pero viendo lo urgente que se está haciendo lo de la emulación 128k, y las ganas que tengo de hacerla de una vez, he decidido sacar esta versión para que la gente tenga un entretenido 'aperitivo' mientras preparo el 'plato fuerte' con calma :-)

Aquí incluyo la lista de los juegos y demos 48k con sonido 128k que tengo. Todos suenan bastante bien con el ZXAM 2.0:

```
180\textdegree{}
ATF
Blazing Thunder
Bobsleight
Brat Attack
Captain Fizz
Carlos Sainz
Complete Bastard
```

Crazy Cars II
 Crazy Cars
 Cybernoid
 Double Dash
 Egg Head II
 Enduro Racer
 Exolon
 Goody
 IK+
 Loopz
 Los Inhumanos
 Mega Phoenix
 Motos
 Mountain Bike Racer
 Mutant Zone (1)
 Mutant Zone (2)
 Plotting Velocidad correcta incluso en un 1200 sin ampliar
 Pulsator
 Rex (1)
 Rex (2)
 Shoot Out
 Silk Worm
 Skateboard Kidz
 Stormlord II
 Stormlord
 Tetris
 The A Team (1)
 The A Team (2)
 The Fury
 The Last Mission

 (Demos)

 Antares II
 Carminadle 3
 Digi Synth 2
 Gallery
 Gemini Incluso samples de 4 bits
 Geography Demo
 Judgment Day 3
 LCD Demo
 Noname 2
 TFA Demo Incluso samples de 4 bits

1.142 Problemas conocidos...

Primero de todo explicar que esta versión es casi ß (estuve a punto de distribuirla como tal) por que no ha sido probada a fondo, y las modificaciones realizadas sobre el emulador son las suficientes como para producir montones de bugs. En el caso de encontrar algún problema, por favor avisame lo antes posible para poder arreglarlo.

A continuación se explican una serie de problemas y limitaciones del emulador:

* Problemas con la prioridad del emulador:

No es recomendable poner a prioridad +1 la tarea de emulación. Esto es debido a que cualquier script ARExx que intentases arrancar podría quedarse bloqueado, ya que el REXXMAST crea una tarea de prioridad 0 para ejecutar el script y al tener el emulador una prioridad superior podría tomarse para sí todo el tiempo de CPU, y el script quedaría 'congelado' a la espera de tener tiempo de CPU para ejecutarse. En caso de que esto ocurra, lo único que puedes hacer es seleccionar la ventana/pantalla del spectrum y pulsar HELP, para parar así la emulación. En ese momento se disparará el script y se ejecutará como debería.

De todas formas, poniendo a +1 la prioridad no conseguirás un incremento de velocidad apreciable. Y en el caso de conseguirlo, será a costa de dejar completamente bloqueadas las tareas de prioridad inferior. Lo recomendable es prioridad 0 e incluso -1.

Otro posible problema es que si ejecutas varias copias del ZXAM en multitarea, sus prioridades nunca deben ser diferentes. De no ser así las copias de prioridad inferior pueden quedar completamente bloqueadas, a menos que las superiores liberen suficiente tiempo de CPU.

* Color del borde en modo NOSYSTEMSCREEN:

En el modo NOSYSTEMSCREEN (o sea, NO multitarea, con pantalla legal) no se puede cambiar el color del borde de forma permanente, o sea, siempre es negro, menos cuando algo genera rayas sobre dicho borde, como la rutina de SAVE a cinta. Esto se debe a que desconozco la forma de cambiar el color 0 de una pantalla legal desde el entorno NO multitarea, ya que las funciones de la graphics.library parecen necesitar la multitarea en marcha. Espero arreglar esto para la próxima versión.

Se recomienda usar los modos NO multitarea solo para cargar/grabar de cinta. Para uso general del emulador se pueden usar los modos multitarea. La diferencia de velocidad entre el modo SCREEN y el NOSYSTEMSCREEN es marginal, a menos que haya alguna otra tarea ejecutándose constantemente en multitarea.

Otro detalle respecto al color del borde en modo NOSYSTEMSCREEN se refiere a las bandas generadas por las rutinas LOAD/SAVE de cinta. Al usar algún modo de pantalla de diferente frecuencia al PAL el ancho de las bandas de color del LOAD/SAVE será diferente del habitual. Esto se debe a la diferente velocidad del barrido vertical de ese modo de pantalla, pero no afecta en absoluto al funcionamiento de las rutinas de cinta.

Por último decir que al ser la pantalla 'legal' también le afecta el hecho de que tengas el 'borderblanking' (borde de pantalla negro) activado mediante el comando BBlank o cualquier otro, por lo que si usas el BBlank no podrás ver el color del borde del Spectrum (un problema cuando intentas cargar de cinta o usar el transfer). Para poder ver el borde en modo NOSYSTEMSCREEN basta usar las líneas de comando PRERUN y POSTRUN, que se ejecutan antes de desactivar la multitarea (PRERUN) y tras reactivarla (POSTRUN). Basta que hagas (en el icono) PRERUN=BBBlank y POSTRUN=BBBlank, y se desactivará el borderblanking mientras la emulación está en marcha.

* No hay 'turbos':

Aún no hay carga real para 'turbos' y demás programas protegidos. La temporización necesaria para este tipo de carga es extremadamente precisa, y el emulador aún no está en condiciones de llegar a esta precisión, al menos con equipos de potencia media. Esta posibilidad tendrá que quedar para posteriores versiones.

* Problemas con el Enforcer:

Durante las pruebas del emulador he notado que el Enforcer ralentiza considerablemente la emulación. Ignoro las causas de esta ralentización, aunque sospecho que tendrá algo que ver con el control de los cachés por parte del Enforcer. Si quieres tener la emulación al máximo de velocidad posible en tu máquina, debes desactivar el enforcer.

* Problemas con los colores de la ventana en pantallas públicas:

En el modo 'Display -> Window' el emulador intenta adaptarse a los colores disponibles en la pantalla pública donde debe abrir la ventana de emulación, pero con varias limitaciones:

1.- El cálculo de aproximación de colores es poco preciso, y en algunos casos los colores usados pueden ser bastante extraños (se debe a que no conozco el algoritmo adecuado para comparar colores y determinar el nivel de parecido).

2.- El emulador NO modifica los colores de la pantalla, aunque dichos registros estén libres para modificar (esto es así por compatibilidad con versiones inferiores a la 3.0 del sistema operativo).

3.- El emulador solo puede usar los primeros 16 colores de la pantalla, aunque ésta sea de más de 16 colores, por lo que puede darse el caso de que el emulador no use los colores más parecidos a los del Spectrum, de entre los disponibles, debido a que no están entre los 16 primeros (esta limitación es debida a la complejidad que supone usar más de 16 colores para la pantalla del Spectrum).

Estas tres limitaciones probablemente serán eliminadas en la próxima versión del ZXAM.

* Velocidad automática con ventana en pantalla pública:

Parece que el modo 'Display -> Window' ralentiza el sistema automático de ajuste de velocidad, con lo que puede pasar que te sobre potencia para correr a la velocidad correcta y sin embargo sobre el workbench el programa va ligeramente más lento de lo que debería. Parece ser que la causa no está en el emulador si no en alguna imprecisión del timer.device cuando se redibuja la ventana sobre el workbench. No he podido encontrar una solución para este

problema, al menos de momento, pero la diferencia de velocidad es pequeña, o sea que puedes usar este modo sin preocuparte.

* Problemas con el sonido en algunos juegos 48k:

En algunos juegos (como el Alien 8) desaparece el sonido 48k o suenan ruidos extraños por los altavoces. Esto es debido a que algunos juegos 48k antiguos mandan datos (por alguna misteriosa razón) a puertos de entrada-salida que coinciden con los usados en el spectrum 128k para acceder al chip de sonido. En el caso concreto del Alien 8, éste manda un dato que activa un canal de tono y otro de ruido, que toma para sí el canal 0 del Amiga, desactivando el audio 48k. Esto es consecuencia de malas prácticas de programación por parte de algunos programadores de Spectrum, y solo se puede solucionar desactivando la emulación del chip AY (menu 'AY-3-8912 -> Off').

1.143 Versiones del ZXAM

Estas son la versiones que ha habido, hasta ahora, del ZXAM:

- 1.0 β
Primera distribución pública (solo AGA).
- 1.1 β
(versión no pública).
- 1.2 β
Creadas versiones FULL y FAST.
- 1.3 β
Implementado LOAD/SAVE desde cinta.
- 1.3b β
Bug fix y transfer.
- 1.4 β
Puerto ARexx (versión no pública).
- 1.5
Creada versión OCS/ECS.
- 1.6
Desensamblador Z80 y traza.
- 1.6b
Bug fix y mas formatos.

....un gran paso adelante....

- 2.0
Multitarea y sonido 128k.

1.144 1.0 B

- 1.0 B 39.00
-Primera versión en circulación.

(distribuida como ZXAM Spectrum Emulator AGA 1.0 B)

1.145 1.1 B

- 1.1 B 39.01 (ejecutable llamado ZXAM_CuelgueSeguro)
-Desactiva filtro de audio al entrar. Lo restaura al salir.
-Desactivadas las interrupciones de los Timers. Actualmente solo permanece activada la del teclado. Esto ha mejorado drásticamente el sonido.
-Debido a la desactivación de los Timers el emulador se "congela" al cabo de un cierto número de pulsaciones de tecla (en realidad el que se cuelga es el HANDLER DE TECLADO DEL SISTEMA, ya que se le desborda el buffer).

39.02 (19-ABRIL-94, tras 2 semanas sabáticas)
-añadido gadget ZOOM a la ventana del programa
-el hadler del teclado me sigue tocando las narices...

(no se llegó a distribuir)

1.146 1.2 B

- 1.2 B 39.03 (26-JUNIO-94, tras 2 meses de exámenes y MPMaster)
-¡Al fin! Arreglado el problema con el teclado. El emulador ya no se cuelga por culpa del desborde del buffer del teclado.
-Ahora, si pulsamos Caps Lock dentro del spectrum, al salir al sistema, éste es informado correctamente del estado de mayúsculas del teclado. Antes, al no recibir bien la señal, escribía minúsculas aunque Caps Lock estuviese iluminado.
-Disponibles versiones FULL (completa) y FAST (rápida) para usar en Amiga 4000 y 1200 respectivamente. La versión FAST es un 10% más rápida que la FULL, aproximadamente, aunque se aprecia en algunos programas mejor que en otros. Esto se ha conseguido eliminando algunos detalles de la emulación.

39.04
-El emulador se "despega" de la ventana del SHELL, dejándola libre por si quieres usarla, o cerrarla.
-Ahora se leen las opciones del icono incluso si arrancas el emulador desde SHELL.

39.05
-Eliminado el problema con el audio. Ahora el sonido del spectrum suena perfectamente aunque antes de arrancarlo hubiera algún tipo de player (como el Delitracker) en marcha. A la salida el tracker seguirá donde estaba (al menos, con los players que yo tengo...).

39.06

- Realizadas algunas optimizaciones preparatorias para la implementación de LOAD y SAVE de cinta.
- Ahora alojamos el Timer-B de la CIA-B para las temporizaciones del emulador, en preparación para el estabilizador de audio.

(distribuida como ZXAM Spectrum Emulator AGA 1.2 β)

1.147 1.3 β

1.3 β 39.07 (7-JULIO-94)

- Primer intento de parchear la ROM para cargar. La carga puede realizarse tanto en B/N como en color, incluso sin la presencia de memoria FAST (!). Tampoco es necesario parar las interrupciones del FLASH (!!). La estabilidad de carga parece muy buena, aunque debe probarse en otras configuraciones (68040, 68030/40MHz o 68030/50MHz). Sobre el 1200 y 4000/030 la carga funciona perfectamente.
- Sincronizada instrucción OUT. Con esto ya tenemos SAVE en TIEMPO REAL (incluso en color!!), y BEEP en tiempo real. El sonido de algunos juegos ha mejorado extraordinariamente (como el Pheenix). Por otro lado otros juegos se han ralentizado alrededor de un 3% (3 weeks in paradise) debido a que la sincronía supone instrucciones adicionales en la rutina OUT.

39.08 (11-JULIO-94)

- ¡Borde REAL! Si la velocidad es suficiente se pueden ver los efectos de borde que hacen algunos programas.
- Reajustada la correspondencia de colores para BRIGHT 0 y 1. Ahora la proporción de brillo es prácticamente idéntica a la del Spectrum original.
- Reajustes del parche de carga. Sigue funcionando igual de bien pero ahora reacciona mejor al BREAK, HELP y ESC.
- Al fin le he dado la segunda vuelta al contador de líneas del Devpac. La versión 39.08 consta de 133637 líneas con las macros expandidas. Si sigo así no me bastarán los 2 Megas del A1200 para ensamblarlo... :-)

39.09 (16-JULIO-94)

- Arreglado el problema del borde con el parche de la rutina LOAD y con la instrucción OUT. Ahora el borde es "casi" perfecto tanto con LOAD como con SAVE (y OUT).

(distribuida como ZXAM Spectrum Emulator AGA 1.3 β)

1.148 1.3b β

1.3b β 39.10 (23-JULIO-94)

- El emulador tiene ahora un "Transfer" integrado para poder pasar a cinta los programas que tenemos en disco (prueba de pulsar F6). Inicialmente incluido para usarlo yo mismo

- en las comparativas de velocidad entre el 1200 y el +2A.
- En el momento de hacer "transfer", el bloque BASIC se graba con el nombre del programa cargado desde disco.
- El transfer no usa rutina de carga propia, por lo que puedes cargar los programas en el propio emulador (;que ilusión! ;-)
- La "transfer"-encia se puede interrumpir con la tecla HELP.
- El transfer graba en el cargador incluso el color del borde, cosa que los auténticos transfers de Spectrum no hacian.
- Al hacer "Save Program" el nombre pasa al gadget "Loaded File", de forma que podemos hacer "Reload" del programa salvado, además de saber el nombre que hemos usado y el formato.
- El requester de SAVE conserva un path diferente al de LOAD, facilitando así la conversión de ficheros entre diferentes directorios.
- Si no se ha cargado ningún programa, al hacer transfer el programa se graba con el nombre LOADER.

39.11 (31-JULIO-94)

- Realizados unos ajustes en la sincronización del emulador. El sonido sigue estando estabilizado, pero ciertos juegos (Manic Miner,...) han recuperado su vieja velocidad ;-D
- Arreglado un error en el transfer. Con este error, solo la primera transferencia funcionaba correctamente. La segunda vez el código Z80 del transfer resultaba corrupto.
- Arreglado el problema del Kempston con el Panama Joe y el Commando.
- El acceso al borde es un poco más rápido.

(distribuida como ZXAM Spectrum Emulator AGA 1.3b ß)

1.149 1.4 ß

1.4 ß 39.12 (2-SEPTIEMBRE-94)

- Primeros ensayos de puerto ARexx. Implementadas funciones GetReg, SetReg, Peek, Poke y Putmem. HA FUNCIONADO A LA PRIMERA!!!!

39.13 (10-SEPTIEMBRE-94)

- Ahora los menús son tipo NewLook (blancos con letras negras).
- Añadidos mas funciones ARexx: DPeek, DPoke, FindByte, y el acceso a los registros Z80 en orientación Z80 y 68000. Posibilidad de recibir comandos y funciones.
- Añadido el código para arrancar scripts ARexx desde el emulador. Ahora tengo que modificar la ventana para el botón de ejecutar script ARexx.
- Escritos ya algunos scripts de ejemplo.

39.14 (16-SEPTIEMBRE-94)

- Añadida la versión interna al About...
- Añadida ToolType ROMFILE para cargar ROMs externas. La ROM puede estar comprimida con powerpacker.
- El nombre del último programa cargado/grabado aparece en el requester de SAVE, pero con la extensión adecuada al formato.
- Añadidas funciones ARexx: Run, Quit, NameFormat, GetMem, ppLoadFile, ppSaveFile, LoadRequester, SaveRequester, ActName,

- ActLoadPath, ActSavePath y ActFormat.
- Modificados los scripts de ejemplo para uso de las rutinas SAVE y LOAD a través de powerpacker.
- Añadido el gadget para ejecutar scripts ARexx (de forma un poco chapucerilla...). Abre un requester de ficheros para elegir el script a ejecutar.
- Añadida ToolType REXXPATH para indicar el path de los scripts ARexx. La extensión es SIEMPRE '.zxm'.
- Ahora al arrancar el programa, éste hace CD al directorio donde está el ejecutable, aunque nos arranquen desde shell con CD en otro directorio. Esto significa que TODOS los paths entregados al emulador son relativos al directorio del ejecutable, o deben contener un path completo.

(no distribuida)

1.150 1.5

- 1.5 39.15 (28-SEPTIEMBRE-94)
- Añadida la posibilidad de grabar los programas comprimidos con la powerpacker.library.
 - Reorganizados los botones de selección de formato para SAVE. Ahora hay dos botones para cada formato (comprimido y sin comprimir).
 - Añadida dirección FidoNet al About ;-)
 - Añadidas funciones ARexx: ProgVersion, ActSaveFormat, SaveFormat, ParseToSave, ParseLoaded, JoinPathName, ClearNameFormat, GetScr, FilePart, PathPart, LoadPath, SavePath, PreRun, ActPreRun, PostRun, ActPostRun, Pattern, ActPattern y BASICToken.
 - Si hay algún problema inicializando el puerto ARexx, el gadget de ejecución de scripts ARexx queda desactivado.
 - Añadida la posibilidad de poner un titulo a los requesters de SAVE y LOAD.
 - Ahora el emulador devuelve correctamente la memoria cuando detecta que falta el WB 3.0+, el 68020+ o los chips AGA.
 - La ventana es una AppWindow, o sea, puedes tirar dentro los iconos de los ficheros a cargar. Si lo que metes es un cajón o un disco, se abrirá el requester de ficheros en dicho cajón/disco.
 - Añadidas ToolTypes PRERUN y POSTRUN para correr comandos antes de arrancar la emulación y después de pararla.
 - Antes de PRERUN se almacena el estado de los cachés, y se recupera al volver, antes de POSTRUN.
 - El emulador ha dejado de ser BETA. Esto significa que la ventana está completa (o sea, que todo su contenido funciona) y que se puede arrancar sin peligro de cuelgues salvajes ni fuegos artificiales (a mi no se me cuelga desde hace meses). Los defectos de emulación no se incluyen en la consideración de BETA, ya que si no el emulador podría ser BETA eternamente. Esto NO significa que renuncie a mejorar la emulación (¡NI MUCHO MENOS!) ;-)
- 39.16 (12-OCTUBRE-94)
- Añadidas funciones ARexx: ClearAbort, ReadAbort y NoReload.
-

-Añadido el gadget 'Abort ARexx' para poder parar los scripts ARexx, aunque es responsabilidad del script el leer dicho gadget.

39.17 (27-OCTUBRE-94)

-Modificadas algunas instrucciones que acceden al flag P/V.
-Creada una versión OCS/ECS del emulador. Esta versión no tiene FLASH y la emulación de color es un poco más lenta que la AGA. Todas las demás posibilidades del emulador AGA están incorporadas a esta versión (ARexx, AppWindow, carga de cinta, etc...)

(distribuida como ZXAM Spectrum Emulator 1.5)

1.151 1.6

1.6 39.18 (14-NOVIEMBRE-94)

-Ahora el emulador aloja legalmente el canal 0 de audio antes de entrar en la emulación, y lo devuelve al salir. En caso de no poder alojarlo no empieza la emulación y muestra un requester.

39.19 (6-DICIEMBRE-94)

-Integrado un desensamblador en el emulador. Hay un nuevo comando ARexx (ZXAMDisassemble(dirección)) que desensambla la instrucción de la dirección indicada.
-Arreglado el problema con el teclado de algunos juegos de DINAMIC (debido a sus técnicas chapuceras de programación).
-Añadido menú para activar/desactivar la sincronización del audio con las CIAs.
-Añadida tooltype CIASync.
-Añadido menú para seleccionar el modo de SAVE de la rutina de la ROM (Cinta/Disco/Disco comprimido). Al salvar un bloque (ya sea cabecera o datos) salimos al sistema con un requester de ficheros, que contendrá ya la extensión adecuada. Si pulsamos CANCEL se realiza un BREAK en el emulador.
-Añadido menú para seleccionar el modo de LOAD (Cinta/Disco)
-Arreglado un error en la gestión de los flags del Z80 al modificarlos desde ARexx.
-Mejorada la lectura de la CIA.
-Ahora el emulador aloja cualquier timer de las CIAs (ya no tiene que ser específicamente el timer-b de la cia-b).
-El timer de la CIA se aloja durante la emulación.
-Arreglado problema con el flash en sistemas con caché de datos (030 y 040).
-Añadida posibilidad de cargar snapshots en formato Z80.

39.20 (22-DICIEMBRE-94)

-Arreglado el problema con scripts arexx situados en un path que contenga espacios. Aún así, el nombre del script no puede contener espacios (es una limitación del ARexx).
-Añadidas tooltypes LOADMODE y SAVEMODE para controlar los modos de carga/grabación.
-Arreglado un problema al substituir la extensión en el nombre de un fichero al hacer 'save program'. Cuando el nombre

- contenia varios puntos, se ponía la extensión después del primero, cuando debía ser después del último.
- Se pueden cargar juegos con iconos cuya default tool es el emulador, o también por multiselección
- Añadido al comando Disassemble la posibilidad de mostrar los datos y direcciones en decimal o hexadecimal
- Ahora al desensamblar una instrucción de salto relativo se imprime como comentario la dirección absoluta a donde va dicho salto.
- El transfer ya siempre va a cinta (arreglado el bug por el cual el transfer era afectado por el menu "Save mode").
- Añadida posibilidad de cargar snapshots Z80 de formato moderno (2.01). Anteriormente solo se podían cargar los de formato antiguo (1.45).
- Modificada función ARexx ParseLoaded() para que reconozca también el formato Z80.
- Eliminado el menu que permite activar/desactivar la sincronización del audio con las CIAs. Eliminada también la ToolType CIASYNC.
- Arreglado un bug inofensivo en la rutina de grabar el formato PC_PP.
- Ahora el emulador aloja TODOS los canales de audio, para una próxima emulación del chip AY-3-8912 del +2/+2A/+3.
- El ejecutable ya no está comprimido con powerpacker por que parece que la rutina descompresora causaba gurús algunas veces.

(distribuida como ZXAM Spectrum Emulator 1.6)

1.152 1.6b

- 1.6b 39.21 (19-ENERO-95)
- Arreglado un error en la carga de snapshots en formato Z80. En muchos casos el emulador no reconocía correctamente los programas en formato Z80 (versión 1.45) comprimidos. Esto es debido a información errónea en el manual del "Z80" 1.45 y a la falta de programas en formato Z80 para probar.
- Añadida la posibilidad de cargar (aunque no grabar) programas en formato KGB.
- Modificadas algunas instrucciones para que incrementen correctamente el registro R.
- Ligeras mejoras en la emulación de los flags del Z80.
- Añadido script ARexx para desensamblar, buscar pokes, y trastear por los programas, como si de un monitor se tratara (Hecho por Leo Cocaña. ¡Gracias Leo! :-)
- Arreglados dos errores en el desensamblador (en las instrucciones 'ld hl,(nn)' y 'ld (nn),de').
- Añadidas funciones ARexx LoadMode(), SaveMode(), ActLoadMode() y ActSaveMode(), para controlar así desde ARexx el menu de opciones LOAD/SAVE.
- Añadida función ARexx About()
- Modificada la función ARexx ParseLoaded() para que también reconozca los snapshots en formato KGB.
- Ahora 'Reload' funciona correctamente tras 'Save program'.
- Tras 'Reset & Run' ya no funciona 'Reload' desde teclado.

- integrada una pausa de 1 segundo al final de la rutina SAVE de la ROM, para programas que salvan bloques demasiado juntos (como el ensamblador GENS).
- Añadidas funciones ARexx FindBlock() y DoChecksum().
- Añadidas funciones ARexx BreakPoint(), Trace(), DoInt() y BlockOfRegs(). Estas funciones permiten ejecutar código Z80 paso a paso, y poner puntos de ruptura.
- Modificada la función ARexx GetReg(). Ahora tiene un segundo parámetro (opcional) con el que se puede indicar el formato del resultado (decimal o hexadecimal).
- Ahora cuando hay en marcha un script ARexx ya no se acumulan los eventos 'cerrar ventana', y los menús son desactivados.
- Arreglado un error en 'Reload' tras grabar un programa en formato PC_PP.

(distribuida como ZXAM Spectrum Emulator 1.6b)

1.153 2.0

- 2.0 39.22 (6-febrero-95)
- Tras varias semanas de modificaciones y reestructuraciones en los sources, parece que el emulador vuelve a funcionar. ¡Menos mal! Seguro que con tantas modificaciones se habrán añadido al emulador algunos bugs nuevos...para variar...
 - Encontrado un bug...en el DEVPAC!!!!
 - Añadido soporte para pantallas 'legales' en la versión ECS.
 - Añadida una MULTITAREA básica a la versión ECS
 - Añadida estabilización de velocidad para ordenadores demasiado rápidos (AL FIN!)
 - Arreglada incompatibilidad con 3D-Tennis, Amaurote y Aquaplane.
 - Añadido handler al input.device para leer las teclas en multitarea.
 - Añadido soporte para multitarea en pantalla independiente o sobre una VENTANA en una pantalla pública, solo versión ECS.
 - Código de dibujado de ventana totalmente legal, para futuras versiones del hardware, o emulaciones del sistema tipo RTG.
 - Añadido código de adaptación de colores para ventana en pantallas públicas (solo kickstart 3.0).
 - La ventana de emulación es una appwindow, donde se pueden lanzar los iconos de los juegos para que sean cargados.
 - Las teclas se leen 'a través' de la ventana de visualización (o sea, solo son leídas cuando dicha ventana está activa).
 - Añadido soporte para desactivar el audio del emulador.
 - Los menús ya no usan la horrible topaz 8. Ahora usan la fuente por defecto de la pantalla actual. La ventana sigue siendo topaz :-)
 - Ahora el nombre del fichero cargado aparece también en la barra de título de la ventana de emulación.
 - El tamaño vertical de la ventana de control cuando es 'encogida' se ajusta a la fuente usada por el sistema para el título.
 - ahora la tecla TAB cumple la función de la tecla 'Graf' (shift+9) del spectrum.
 - Los programas pueden ser cargados mientras el emulador está en marcha. Serán cargados y arrancados automáticamente.
 - 'Reforzado' el RESET para que NUNCA se pueda quedar el emulador congelado sin posibilidad de salir...al menos, eso creo...

- Arreglado un bug en la simulación de BREAK cuando se CANCELa un requester de LOAD/SAVE desde BASIC.
- Arreglado un problema con el reconocimiento de algunos snapshots formato MIRAGE, que contenian extraños numeros de color de borde...
- Incorporada emulación de joystick SINCLAIR (también conocido como INTERFACE II) y CURSOR (también llamado AGF o PROTEK).
- El joystick ahora es selectivo como el teclado, o sea, solo es leído cuando la ventana/pantalla de emulación es seleccionada.
- Mejorada copperlist OCS/ECS para que en maquinas AGA se aproveche mejor el ancho de banda adicional del sistema gráfico.
- Implementada una emulación (muy) básica del chip de sonido 128k (el AY-3-8912).
- Añadidas tooltypes AUDIO, AYEMULATION, SPEED, JOYSTICK, TASKPRI, DISPLAY y REFRESH, para controlar las nuevas posibilidades del emulador.
- Añadidos los registros del AY-3-8912 a la lista de registros manipulables desde ARexx. Son AY0, AY1,.....AYE, AYF, y el especial AYX que es el ultimo byte enviado al puerto \$FFFD.
- Añadida posibilidad de cargar snapshots Z80 V3.0.
- LOAD/SAVE a CINTA solo están disponibles en modo NO multitask.
- Ahora la prioridad de la TASK de la ventana de control es SIEMPRE un punto superior a la prioridad de la task de emulación. De esta forma el propio emulador no ralentizará el requester de ficheros ni cualquier otra operación de la ventana.
- Incorporado control dinámico de hardware. La CIA y el audio son alojados SOLO cuando son necesarios.
- Añadida tooltype SCREENMODE para especificar el modo de pantalla en que queremos que se muestre el emulador.
- Eliminada toda sincronización con el VBlank, ya que en modos multisync/VGA no tiene sentido.

39.23

- Añadido el centrado automático de la pantalla respecto a las preferencias de overscan del sistema.
- Añadido menu para modo NO MULTITASK, pero sobre una pantalla legal del sistema (es el modo SCREEN pero desactivando el sistema).
- Añadido al freno automático un sistema de descarga de la CPU en modo multitarea. Cuando el procesador es muy rápido, la potencia que 'sobra' es entregada al sistema para las demás tareas.
- Ahora cuando corremos sobre una ventana, solo se procesan los bitplanos realmente usados en la pantalla. Esto acelera la emulación sobre pantallas de menos de 16 colores :-)
- El sistema de aproximación de colores ha sido simplificado. La aproximación intenta ser lo mejor posible, pero si los colores son demasiado diferentes a los del spectrum, los resultados pueden ser bastante sicodélicos....
- Ahora ya pueden 'convivir' perfectamente varios ZXAM en marcha al mismo tiempo, sin que se interfieran.
- Ahora ya no es necesario alojar una CIA para el estabilizador de velocidad en multitarea. En su lugar se usa el timer.device. De esta forma se pueden arrancar todas las copias que se quiera del emulador en multitarea, distribuyendose entre si el tiempo de CPU disponible, y devolviendo el sobrante.
- Probada con exito la emulación sobre ventana en workbench en un workbench emulado con una tarjeta Retina Z2 :-)
- Añadido tooltype PUBSCREENNAME para especificar el nombre de la

- pantalla publica donde abrir las ventanas.
- Añadido un requester de aviso para indicar cuando el ejecutable de emulación esté en memoria CHIP, avisando así de la seria ralentización resultante...
- En modos no multitarea el audio es alojado SIEMPRE. Así nos aseguramos de que no hay ningún player en marcha al desactivar la multitarea.
- Ahora, al abrirse sobre una pantalla publica, dicha pantalla se lleva al frente.
- Ahora en el título de la ventana de control se especifica si la emulación está en marcha (running) o parada (stopped).
- Bajo WB 3.0, se usa BestModeID para conseguir el monitor del workbench si no se nos especifica ninguno en concreto.
- La barra de menú ahora también está 'pegada' a la ventana de emulación. Mas cómodo ¿no? :-)
- La emulación se para cuando se abre el requester para load/save.
- Ahora se le puede suministrar al emulador un parámetro desde CLI, que es el path+nombre del juego a cargar en el momento de arrancar.
- Ahora la task de la ventana de control SIEMPRE tiene la prioridad +2, en previsión de que haya otras emulaciones en marcha con diferentes prioridades a las de la emulación actual.
- Mejorado ligeramente el aprovechamiento del caché de instrucciones del 020+
- Añadida a la emulación del chip AY el canal de ruido blanco.

39.24 (y 37.24)

- AL FIN! Terminada la emulación completa del chip AY. Incorpora emulación de canal de ruido blanco y de envolventes de volumen. Incluye un sistema de mezcla dinámica de ruido para que los instrumentos compuestos por tono+ruido suenen correctamente mezclados, o al menos esa es la intención.
- Ahora al arrancar varias copias del emulador, SOLO la primera tiene el puerto ARexx activo. Esto evita conflictos en ARexx (comando ADDRESS , y duplicidad de funciones en el sistema).
- Añadidas funciones ARexx Audio(), AYEmulation(), Speed(), Joystick(), TaskPri(), Display(), Refresh(), ActAudio(), ActAYEmulation(), ActSpeed(), ActJoystick(), ActTaskPri(), ActDisplay(), ActRefresh(), WriteROM(), AYBlockOfRegs(), ActRun() y CheckSumString().
- Ahora se redibuja el contenido de la ventana cuando se hace un POKE desde la ventana de control.
- Ahora cuando hay problemas con la creación del puerto ARexx ya queda totalmente desactivado toda posibilidad de ARexx. Antes solo quedaba parcialmente desactivado.
- Añadido el control de load/save desde ARexx. Ahora se puede desviar la carga/grabación a un script ARexx (el script ZXAM_TAPE.zxam) para gestionar ficheros compuestos como los .TAP o comandos externos como los de carga de discos +D. Para tal fin se han añadido las funciones ARexx: TapeAction(), TapeLoadName(), TapeSaveName(), TapeAddress(), TapeSize(), TapeFlag(), TapeLoadOffset(), TapeSaveOffset(), TapeSetResult(), TapeSetLoadOffset(), TapeSetSaveOffset(), TapeSetLoadName() y TapeSetSaveName().
- Añadida la ejecución del script ARexx ZXAM_INIT.zxam en el momento de cargar el emulador.
- Añadida la ejecución del script ARexx ZXAM_EXIT.zxam en el

- momento de quitar el emulador.
- Docs convertidos a AmigaGuide
- Añadido menu AY-3-8912 -> Smart, y las tooltypes/ARexx adecuadas.
- Las versiones ECS y AGA se han separado como 37.24 y 39.24, a efectos de poder diferenciarlas desde ARexx con la función IntVersion().
- Añadida la tooltype OUTPUT para poder suministrar a los scripts ARexx una ventana de salida. Esto facilita inmensamente el debug de los scripts ARexx :-)
- Añadidas funciones ARexx CompareString(), ControlToFront() y EmulToFront().
- Añadidas funciones TapeMark() y TapeSetMark()
- Ahora al hacer RESET del spectrum, los nombres TAPE (tapeloadname y tapesavename) son borrados, y los offsets puestos a 0.
- Ahora si falla la conversión de la ventana de control en appwindow el emulador arranca de todas formas. Asi no es obligado que el WB esté en marcha para arrancar el emulador.
- 'Resucitado' el transfer. Ahora vuelve a funcionar en ambas versiones (Genérica y AGA). SOLO modos NO multitarea.
- El modo de pantalla por defecto es el de la pantalla donde está la ventana de control, en vez de ser el del workbench.
- No se pueden seleccionar los modos LOAD/SAVE ->ARexx cuando el puerto ARexx no ha podido ser creado.
- El transfer desactiva todo el sonido 128k cuando tiene que grabar el programa actual hacia la cinta.
- Ahora el transfer ya no estropea la ROM tras la transferencia, y puede interrumpirse tanto con HELP como con la tecla F6.
- Perdido totalmente el control del contador de líneas del Devpac. Ahora ya no sé cuantas 'vueltas' ha dado, por lo que ignoro el numero de líneas de ASM resultantes en el momento de ensamblar.
- Añadidos los menus 'Project -> Load Program' y 'Project -> Save Program', con su correspondiente atajo de teclado. De esta forma se pueden cargar/grabar snapshots desde la misma ventana de emulación (muy práctico para probar montones de juegos).
- Las versiones FASTER vuelven a funcionar. Con todas las modificaciones realizadas habian quedado 'fuera de combate'.
- Ajustado el modo 'Speed -> Fast' para que no se descontrole en algunos casos (como la musica del Silk Worm).
- Arreglado un error en la rutina de carga de ficheros .Z80 v3.0.
- Ahora ya se leen los registros del AY-3-8912 de los snapshots .Z80 (2.0 y 3.0).
- Arreglado un bug en la rutina de generación de ruido blanco.
- Arreglado un bug en la rutina de alojado de audio. En ocasiones, cuando el audio no podia ser alojado, el emulador aún seguia intentando generar sonido.
- Arreglado un error en el puerto ARexx. Al arrancar una segunda copia del ZXAM ésta no tenia puerto ARexx, pero al salir desactivaba el puerto de la primera copia.
- Reajustada la tabla de colores del emulador. Algunos colores oscuros lo eran demasiado.
- Arreglado un bug en el transfer. Cuando el nombre del programa tenia varios puntos, tomaba como separador de extensión el primero, cuando debia tomar el último.
- Ahora la emulación del chip AY ya soporta los 3 canales de samples de 4 bits que usaban algunas demos de Spectrum.
- Añadida función ARexx Stop() (un olvido increíble!!).
- Ahora el joystick kempston está siempre activo. Lo que se

- elige en el menu 'Joystick' son emulaciones adicionales, pero que no desactivan la emulación kempston.
- Ahora hay dos modos Sinclair, el 'Sinclair Right' que emula el joystick derecho (teclas 6, 7, 8, 9 y 0), y el 'Sinclair Left' emula el izquierdo (teclas 1, 2, 3, 4 y 5).
 - Modificadas las funciones ARExx Joystick() y ActJoystick(), y la ToolType JOYSTICK, para aceptar la emulación 'Sinclair Left'.
 - Arregladas las funciones ARExx BasicToken() y Trace().
 - Ahora, cuando la emulación está en marcha en multitarea, las siguientes funciones devuelven un error: GetReg(), SetReg(), ParseLoaded(), ParseToSave(), BreakPoint(), Trace(), DoInt(), BlockOfRegs(), AYBlockOfRegs(), WriteROM(), PutMem(), Poke() y DPoke().
 - Eliminadas varias funciones y tooltypes en versiones FASTER y AGA, por ser irrelevantes dichas versiones.
 - Realizados unos reajustes definitivos en los cálculos de comparación de colores. Parece que la aproximación de colores es bastante mejor ahora.
 - Añadido un sistema de 'seguridad' al remapeado de colores sobre pantalla pública, para evitar que 'desaparezcan' gráficos debido a tener el mismo color de papel y tinta una vez remapeados. Ahora aparecen TODOS los gráficos incluso sobre pantallas de 2 colores.
 - Incluido el script ARExx que uso para cargar juegos desde el DirOpus.
 - Añadido menu 'RUN' para poder usarlo desde teclado en la ventana de emulación.

(distribuida como ZXAM Spectrum Emulator 2.0)

1.154 Futuro del emulador

¡ Definitivamente, el futuro es el emulador de 128k !

1.155 ¡Gracias a toda esta gente!

Mis más sinceros agradecimientos (sin un orden específico) para:

- Fco Javier Cocaña Galán, Leonardo Cocaña Galán y Juan A. Estela Valín por su amistad, y por sus constantes sugerencias (y feroces críticas).
- Mi hermana, que sigue resistiendo denodadamente la tentación de degenerar en una pija perdida como sus compañeras de clase. Ah, y por grabarme los capítulos de Dr Slump, Dragon Ball y Ranma ½ :-)
- Commodore-Amiga Inc por crear el mejor ordenador del mundo.
- Motorola por sus excelentes microprocesadores, fáciles de programar y extremadamente flexibles.
- Nico François por crear las fantásticas librerías reqtools.library y powerpacker.library.
- Jan van den Baard por el utilísimo GadToolsBox.
- IBM y Atari por crear unos ordenadores tan malos y facilitar así mi elección por el Amiga (muchiiiiisimas gracias).
- Miguel Barnosi, Sysop de TANIT-BBS, por haber distribuido el emulador por otras BBS, y por iniciarme en lo de FidoNet. Ah, y ahora, por ser el

- mejor BOSS amiguero de Baleares ;-)
- Raúl Ureña Sánchez, Arturo Rubio Pavón, Javier López Cosialls y Sergi Martínez Colldeforn por sus ánimos y sus sugerencias.
 - Andrew Pointon (Inglaterra) por mandarme unos snapshots en formato Z80 (sin los cuales no podía implementar el formato) y ficheros en formato .TAP (que tampoco tenía).
 - Flávio Massao Matsumoto (Brasil) por mandarme el "ZX Spectrum FAQ" de Internet. Sin él, no habría podido añadir el formato KGB.
 - Jordy Mejias por pasarme el disco del KBG, que yo había borrado hacia tiempo.
 - Richard Harris (Sudáfrica) por los iconos MagicWB que me ha mandado para usarlos con el ZXAM (¡los míos eran horribles!).
 - Juan Gómez y Juan Jordana por subir la versión 1.6b del emulador a Aminet.
 - Todos los usuarios registrados, por considerar que mi trabajo vale la pena pagarlo. MUCHÍSIMAS GRACIAS.
 - Rafat Krawczyk (MuabDib/ESI, Polonia) por mandarme los módulos de música 128k y el player para el Delitracker II. Ah, y sobre todo por las DEMOS de Spectrum, que fueron la causa de que emulara el chip AY mucho antes de lo previsto, y me han servido MUCHO para probar y mejorar la emulación de ese chip.
 - Alberto M. Ordoñez Tellez (Namek), por ser un probador de juegos incansable.
 - Toda la gente de las áreas Amigueras de FidoNet R34, donde uno se encuentra como en casa ;-)
 - Toda la gente que me ha escrito desde dentro y fuera de España, por sus ánimos, sugerencias y críticas.

1.156 ¿ Quien ? ¿ Donde ?

Te agradecería que me enviases toda información que me pueda ser útil para ampliar la potencia del emulador, como por ejemplo, información sobre otros formatos de fichero. En el caso de querer informar sobre algún error o incompatibilidad lo mejor es que me mandes el/los programas de Spectrum que producen dicho problema, junto con toda la información posible sobre el equipo que usas (procesador, sistema operativo, tamaño y tipo de memoria, etc...).

En estos momentos estoy buscando varias cosas para mejorar la emulación, que son:

- Programas 48k con sonido 128k, para probar si la emulación del chip AY está completa y libre de errores.
- Información sobre el joystick tipo Fuller, para emularlo.
- Información sobre los ratones de Spectrum (AMX, Star, etc...) para emularlos.
- Demos de Spectrum, y el tracker para el chip AY.
- Un algoritmo de comparación de colores (como el usado por ImageFX o ADPro) para mejorar el ajuste de colores de la ventana sobre pantalla pública.
- El formato de los módulos AY usado por el AYPlayer (el player de músicas 128k para Delitracker II).

Para enviar sugerencias, informar de errores, intercambiar información, etc, contacta conmigo en esta dirección:

Antonio J. Pomar Rosselló
C/ Alférez Cerdá nº 13-bajos
Palma de Mallorca 07014
Baleares (España)

o, si tienes modem, puedes contactar conmigo a través de TANIT BBS-IBIZA (971-392829, USRobotics 28.800 baudios, nodo de AmigaNet, FidoNet e InterRed). Deja un mensaje a nombre de TONIACE (llamo muy poco por terminal, o sea que la respuesta puede retrasarse mucho).

En esta misma BBS puedes conseguir la versión mas reciente del emulador, haciendo FileRequest con el Magic Name 'ZXAM', aunque la siguiente versión tras la actual 2.0 tardará varios meses en estar lista.

A través de FidoNet puedes mandarme mensajes a: Toni Pomar (2:347/11.3 y 2:343/119.80).

Por AmigaNet también puedes mandarme mensajes a: Toni Pomar (39:195/1.3 y 39:190/1.80).

(si tienes ambas redes disponibles, es preferible que me mandes los mensajes por AmigaNet)

En InterNet mi dirección es: tpomar@penedes.mazanet.es

(mi antigua dirección tpomar@penedes.mazanet.encomix.com ya no es válida)

NOTA: mi acceso a InterNet es limitado y lento, o sea que no te desesperes si tardas en recibir respuesta. Puedes contar con, al menos, un retraso de 15 días.

Palma de Mallorca 8-Junio-1995

1.157 Mensaje para los usuarios registrados

Antes que nada quiero pedir mis mas encarecidas disculpas a todos los usuarios registrados por el gran retraso con el que les he contestado. He tenido serios problemas de tiempo, y sobre todo de programación, que han motivado este retraso.

Tras esto también querria explicar que esta no es una versión 'definitiva' en el sentido que es simplemente una versión 'a medio camino' hacia la definitiva 128k (en realidad es 'casi B'), pero que he querido distribuir por que me ha parecido que valia la pena, y para mostrar que realmente estoy mejorando el emulador.

Es por esta condición de 'provisionalidad' que esta versión aún NO lleva una llave para los usuarios registrados.

TODOS los usuarios registrados recibirán, además de esta versión (que pueden considerar 'gratuita'), la próxima versión 128k, tan pronto como la tenga lista, junto con la llave correspondiente.

Definitivamente, no puedo aventurar una fecha concreta para la distribución

de la versión 128k. Aparte de la emulación de la paginación de memoria tendré que modificar montones de comandos ARexx, la ventana del emulador, implementar formatos 128k, reescribir partes enteras del emulador (como la ejecución paso a paso, los parches de la ROM o el transfer), añadir nuevos parches a las ROMs del 128k.... y, por supuesto, una vez hecho esto, el emulador estará al menos dos o tres semanas de intensivas pruebas para comprobar que está libre de errores, por que con lo que van a tener que 'sufrir' los listados me extrañaría que funcionase todo a la primera...

Simplemente confio en tener el emulador listo para otoño...