



FLASHJACKS

Manual de usuario

Aquijacks 2018 v1.1

Sumario

Introducción.....	3
Especificaciones técnicas.....	5
Características del Hardware:.....	7
Manual de uso. Configuración.....	9
Introducción.....	9
El sistema operativo propietario.....	11
El expansor de Slots.....	15
La unidad FLASHROM.....	17
Cargar y salvar bloques de memoria.....	20
La unidad RAM MAPPER.....	24
La unidad MEGARAM.....	25
La unidad SINFOX compatible.....	28
La unidad SUNRISE compatible.....	30
El sistema de sonido de FLASHJACKS.....	33
El módulo PSG.....	36
El módulo SCC.....	38
El módulo FM.....	40
El bus de audio.....	42
Otros aspectos configuración general.....	44
Carga de ROMS desde MSXDOS. "FROM.COM".....	47
Un poco de MultiMente.....	51
Otro poco de SDs y archivos.....	52
Epílogo.....	54

Introducción

Creo que desde que tengo recuerdos he sido siempre un entusiasta de la electrónica y de la informática. Allá por los ochentaitantos, me quedaba hipnotizado con los ordenadores y con lo que podían hacer. Por mi vida, se han cruzado varias veces el mundo del MSX y siempre he quedado fascinado por aquel fantástico ordenador que curiosamente no todo el mundo seguía. No sé si era la música, las compañías de software japonesas o qué sé yo pero el MSX tenía algo que enganchaba.

Y os preguntareis, ¿Porqué nos cuenta un relato si esto se trata de un manual de instrucciones? Imaginad que al leer las instrucciones de un microondas el diseñador te comenzase a contar su vida de como llegó a ese trabajo. Pues bien, desde un principio quiero comentar que esto no es un microondas ni yo tengo una empresa, ni me dedico a fabricar en cadena este artilugio ni tengo propósitos comerciales como si cual juego de hoy en día se tratase. Las horas de empeño y sacrificio que se han invertido en esto, si es mirado por el lado comercial, no compensa ni de lejos.

Se trata de un sueño, un deseo, un propósito. Algo que con mis conocimientos y mi ímpetu me he decidido a construir y deseo que su resultado final sea de vuestro agrado al igual que ha sido para mi.

Una vez comentado el concepto de este sueño y dejado bien claro que es “la locura” de un aficionado a los MSX pasemos a comentar el porqué de su existencia.

En este último encuentro con los MSX, allá por el 2013, observé que había mucha electrónica de diseño por aficionados que cubrían una u otra carencia del sistema ya sea por obsolescencia e inexistencia del sistema original o por necesidades de adaptación a nuevos medios de almacenamiento.

Y entonces pensé, ¿Porqué no un cartucho que lo haga todo? ¿Porqué no también algo que suene de verdad a como debería haber salido en su día?

Y aquí tenéis el resultado. Mas de 2.000 horas de estudio y sacrificio, mas de 23.000 bloques LE y un montón de fracasos de código a mis espaldas han llevado a lo que tenéis entre manos.

Algo que personalmente se sale de prestaciones para el ordenador al que va dirigido y en el siglo que nos encontramos donde el mundo 3D y virtual van de la mano. Algo que pasará inadvertido para la inmensa mayoría por ser un objeto que va dedicado a rendir homenaje a algo obsoleto y desfasado. Algo que ninguna empresa actual en su sano juicio invertiría la técnica y el desarrollo que se han sumado para obtener un producto que la “mass media” ni sabe ni sabrá para que sirve.

Sin mas dilación, y porque no quiero que se convierta esto en una novela, damos paso a su descripción técnica y manual de funcionamiento.

Gracias por compartir este momento juntos y deseo que mi sueño pueda ser disfrutados por todos vosotros.

Especificaciones técnicas

Dispositivos sintetizados:

- 1 Expansor de SLOTS. Con posibilidad de anulación o activación de sus cuatro subslots para la inserción interna de cualquier dispositivo sintetizado y mostrado en este manual. Cualquier dispositivo puede usar cualquier subslot que nosotros digamos.
- 4 unidades FlashROM con 14 mappers de carga y varios modelos sintetizados de EPROMS que se autoseleccionan por modelo predictivo de carga (AMD, ATMEL, MICROCHIP, etc...). Los mappers disponen de un sistema predictivo que mezcla la heurística con una base de datos. Cada unidad FlashROM tiene capacidad de carga de hasta 4 Megabytes cada una. Todas las unidades tienen la posibilidad de usar códigos de borrado y escritura, etc.. permitiendo leer y grabar infinitas veces hasta su apagado. No se deteriora por el número de grabaciones. Velocidad de acceso inferior a 40nS e incluso picos de 2nS.
- 4 unidades RAM MAPPER compatible incluso con los MSX1. Configurable desde los nativos 64kB (para algunos primigenios MSX) hasta los 4 Megabytes cada una. Velocidades de acceso inferior a los 40nS e incluso picos de 2nS. Autorefresco del propio cartucho sin necesidad de uso del MSX.
- 4 unidades MEGARAM compatible. Configurable hasta los 2 Megabytes cada una. Velocidades de acceso inferior a los 40nS e incluso picos de 2nS. Autorefresco del propio cartucho sin necesidad de uso del MSX.

- 1 unidad SINFOX compatible hasta ver 4.04 con acceso a SDCARD. Acceso a su EPROM de origen con su mapper y con posibilidad de escritura tanto en tarjeta como en EPROM.
- 1 unidad SUNRISE compatible IDE. Autoconversión y operaciones al vuelo a tarjeta SDCARD. Los datos se transforman IDE<-->SD de forma transparente al usuario. Compatible con drivers NEXTOR SUNRISE IDE Ver 2.1 Alpha2. Solo compatible en modo LBA. No compatible en modo CHS u otros. Compatibilidad total de lectura y escritura y pestaña SD contra escritura accidental. Acceso a SDCARD compatible SDSC y SDHC hasta 4GB que es lo que permite el driver NEXTOR. Futura posibilidad de aumentar capacidades con modificación de los drivers. Compatible extracción e inserción tarjeta SD en caliente. Todo ellos gestionado con un driver adaptado para la FlashJacks.
- 1 unidad sonido PSG Stereo seleccionable. Tasa de muestreo 48khz 16bits por canal. Sin pérdida de ranura.
- 2 unidades sonido SCC Stereo seleccionable. Tasa de muestreo 48khz 16bits por canal. Seleccionable el Slot o subslot ocupado. Este dispositivo permite la doble ocupación de ranura.
- 1 unidad sonido FM Stereo seleccionable. Tasa de muestreo 48khz 16bits por canal. Sin pérdida de ranura para la captación del audio. Además, se puede añadir una unidad FMPAC con sintetización completa donde se incluye el soft interno con su mapper original e incluso una SRAM de 8kbytes con autosalvado predictivo.

Características del Hardware:

- Varios mappers sintetizados (Auto, KONAMI4, KONAMI5, ASCII8K, ASCII16K, SUNRISE, SINFOX, ROM16K, ROM32K, ROM64K, RTYPE, ZEMINA6480, ZEMINA126, FMPAC, CROSSBLAIM, SLODERUN)
- 32 Mbytes de RAM DDR con autogestión de los refrescos y de la capacidad según las diferentes configuraciones. Acceso de 2nS
- Múltiples buses de datos a diferentes frecuencias. Pasarelas entre ellos e interdependientes. (300Mhz, 140Mhz, 50Mhz, 3.58Mhz, Reloj MSX, 24Mhz entre otros).
- Sistema operativo propio compatible FAT12, 16 y 32 realizado íntegramente con sistemas lógicos de sintetizado (No hay una sintetización de una CPU que gestione el SO). Dicho sistema es interdependiente del MSX por lo que permite la carga y gestión de todos los componentes, ROMS, etc... antes incluso de que inicie el MSX. Dicho SO es probablemente único en el mundo ya que no he encontrado ninguna sintetización de un SO con puertas lógicas.
- Sin CPUs para la programación secuencial. Cada modelo secuencial sigue su propio patrón de Turing independientemente por lo que la velocidad es máxima y simultanea en todos los módulos. No hay retrasos ni afección intermodular.
- Sistema de sonido bidireccional de hasta 96khz a 24 bits en Stereo para una máxima experiencia.

- Bus de audio Stereo independiente del MSX por lo que el PSG+SCC+FM es mezclado y tratado internamente. Con ello se obtiene un muy bajo nivel de ruido y una máxima pureza de sonido. (Se elimina el problema de volumen y mezclado con cartuchos externos).
- Filtrado del audio por sintetización y por hardware físico. Seleccionable.
- Salida y entrada de audio por Jack con capacidad de potencia para manejar auriculares.
- Independencia del bus de salida por Jack y del bus de audio mono interno del MSX. Conversión a Mono para bus interno y amplificación por hardware independiente para máxima compatibilidad con la norma MSX y mínima afección al Jack exterior.
- Conector de expansión para futuras ampliaciones de Hardware.
- Sistema de encriptación anticopia recurrente y vigilado. El Hardware sintetizado es abierto al uso y experimentación para la creación del software que se quiera y las configuraciones experimentales que se desee pero la copia de FLASHJACKS está protegida mediante sistemas criptográficos redundantes. (Como ya he comentado en la introducción, no voy a hacer una explotación comercial del mismo pero si que me voy a reservar el secreto de protección por el momento).

Manual de uso. Configuración

Introducción.

Vamos a explicar la filosofía interior de FLASHJACKS.

Este cartucho es un todo y un nada a la vez. Me explico. Todo parte de una configuración inicial que es donde seleccionamos el hardware y a partir de ahí y de modo instantáneo el hardware queda visible y presente para nuestro MSX hasta su apagado. Por eso es un todo, y es un nada porque si no configuramos nada, esa opción también es contemplada y nada queda sintetizado por lo que el MSX no encontrará nada en ese SLOT.

Bien, todo es muy sencillo. Podemos dejar la configuración standard sin tocar eternamente y no habrá problemas pero también podremos toquetear todo lo que queramos como si del hardware real tuviéramos. Para adelantaros curiosidades, podremos cargar 16MB de RAM en los cuatro subslots disponibles o incluso cargar dos NEMESIS en dos FLASHROMS independientes para jugar a las pantallas extras sin necesidad de parchear nada y como si de verdad lo tuviéramos (En el caso de que fuese compatible ya que tienen la costumbre de bloquear donde poner cada juego para el modo extra) .

La configuración se carga en el directorio raíz de la SD mediante el archivo de texto CONFIG.CFG. El Cartucho al encenderse lo primero que hace es bloquear el MSX y leer este archivo para saber en que se tiene que sintetizar. Como hemos dicho somos independientes del Z80 y con una velocidad de procesamiento abismal en comparación al equipo al que estamos conectados. Ya veréis que todo el proceso de sintetizado, inicializado de la tarjeta SD y carga de ROMS suele ser inferior a un par de segundos.

Una vez inicializado el MSX, nuestro sistema se vuelve transparente y hace exactamente lo que en la configuración le hemos dicho hasta su próximo apagado. Con el reset revive de nuevo para ejecutar ciertas operaciones especiales que de otro modo no podría como por ejemplo la escritura de ROMs a archivos o el salvado de la SRAM. Si, lo habéis leído bien. Existe la posibilidad de obtener copias de las ROM cargadas pues por ejemplo para salvar una carga que hemos hecho desde nuestro MSX dentro de su sistema operativo.

Y esto es todo. Parece complicado pero ya veréis que es muy sencillo, tanto para la persona que no quiere tocar nada como el que tiene curiosidad en probar configuraciones de lo mas inverosímiles. Tened en cuenta que los MSX siguen una norma pero internamente ninguno es igual a otro. Es por ello que con estas configuraciones pueden saltarnos fallos que no sabíais de su existencia y que en otros MSXs no seáis capaces de reproducirlos.

Vayamos pues a explicar su uso.

El sistema operativo propietario.

FLASHJACKS cuenta, como ya hemos dicho, con un sistema operativo propietario. Para nosotros externamente se comunica mediante unas instrucciones muy sencillas que lee del archivo CONFIG.CFG en el directorio raíz.

Este SO internamente tiene integrado el sistema FAT 12,16 y 32, lectura del directorio raíz, gestión de archivos tanto lecturas como escrituras y un driver que gestiona la tarjeta SD de forma nativa tanto las versiones SDSC como las SDHC hasta una capacidad ilimitada siempre que cumplan con la norma de alta capacidad SDHC.

Dicho SO tiene una velocidad de procesamiento muy alto ya que no lo rige una CPU si no uno o varios punteros dedicados y exclusivos para esta tarea. De hecho he conseguido transferencias de datos en modo serie de 1 bit a la tarjeta SD superior a los 4MBytes por segundo. Obviamente en su conjunto, comunicándonos con todos los módulos, no conseguiremos esas tasas de transferencia pero si que será habitual cargar y grabar ROMS a tasas de 500kBytes por segundo, dependiendo del fragmentado de nuestra tarjeta. Comentar que se ha impuesto la seguridad a la velocidad y que para que os hagáis una idea, cada byte leído se lee 8 veces y se le realiza un método estadístico para evitar errores de lecturas por culpa de malos contactos o similares (de hecho, el nivel estadístico se hace a un estado aún mas bajo que el byte de interpretación).

Sigamos con la configuración y sus comandos.

Disponemos de un archivo de texto tipo CONFIG.CFG donde el sistema nos va a leer nuestra propuesta. Comentar que este sistema va a intentar leerlo ignorando espacios adicionales y otros errores menores por lo que en el peor de los casos nos ignorará el comando con el error y en el mejor de los casos lo interpretará adecuadamente pero nunca nos dejará el sistema bloqueado. Este archivo es editable al mismo estilo que un AUTOEXEC.BAT por lo que podemos usar el editor al que estamos acostumbrados normalmente tanto en MSX como en un PC.

El archivo tiene una capacidad máxima de 8192 bytes por lo que no podemos superar en texto esa capacidad. Tampoco es necesario tanto.

El sistema internamente interpreta solo mayúsculas pero no pasa nada que escribamos por error minúsculas. Él lo postprocesa a mayúsculas internamente.

Vayamos a los comandos.

Siempre hay un comando que le precede un símbolo igual "=" y una serie de configuraciones, terminado con un enter que le dice que es un fin de operación.

Como hemos dicho espacios entre comandos, instrucciones, etc.. se intentan ignorar.

Añadir que, en el caso de haber editado desde el MSX y habernos cargado la configuración, puede darse el caso de no disponer un PC a mano para restaurarla. Existe un modo de que nos arranque en un modo básico con acceso a la unidad.

Pulsando mas de 5 segundos la tecla HOME, en el BASIC o MSXDOS, hasta apagar la luz roja intermitente, al resetear obtenemos un modo de carga segura. Esta nos permitirá editar el CONFIG.CFG y arreglarlo. Al volver a resetear, nos cargará de nuevo el CONFIG.CFG.

En esta carga segura, nos cargará RESCUESU.ROM para un modo Sunrise o RESCUESI.ROM para un modo SINFOX. Todo depende de la ROM que encuentre se cambiará a un modo u otro.

Si todo esto falla por borrado completo de la tarjeta, recordad que siempre tendréis dos ZIPS con todo lo necesario para restaurar a un sistema u otro. Aquí si que ya es necesario un PC para copiar los archivos. En el modo rescate podremos desde BASIC realizar un _FDISK para SUNRISE o usar las aplicaciones de SINFOX para realizar el correspondiente formateo de la tarjeta SD.

Ahora vamos a aprender dos instrucciones que sirven para la carga y salvado de configuraciones:

`LOADCONFIGFILE = ARCHIVO.CFG`

Tal cual lo hemos puesto, nuestro sistema interpretaría que queremos cargar un archivo de configuración. Esto es para no usar el archivo CONFIG.CFG como configuración si no como puntero o indicador de donde se encuentra nuestra configuración. Imaginemos que tenemos ocho configuraciones distintas y que queremos mantenerlas todas. Para no estar renombrado archivos todo el rato, podemos llamarlos como se nos antoje, para luego entrar en el CONFIG.CFG y llamar directamente a nuestro archivo. Por cierto, el formato de nombre del archivo debe cumplir con el standard MSDOS 8.3, o sea, 8 caracteres para el nombre, un punto y tres caracteres para la extensión. Para nuestras configuraciones no hace falta que sean extensiones .CFG pero si recomendable.

El resto de comandos se ignorarán después de este ya que hace ejecución secuencial de comandos por lo que al ejecutar este, nuestro siguiente comando a ejecutar se encontrará en el nuevo archivo leído.

`SAVECONFIGFILE = ARCHIVO.CFG`

Este de momento no tiene mucha utilidad. Nos salva el archivo actual en un nuevo o existente archivo con la configuración presente. Aquí lo único que advertiremos es que el archivo destino tendrá algunos errores corregidos como pueden ser los caracteres en minúsculas que aparecerán en mayúsculas.

Como veis es fácil. Nuestra configuración será escrita en una secuencia de instrucciones.

La estructura es sencilla. Acordaros del return que le indica fin de la instrucción. Cualquier omisión tomará un valor por defecto y si no puede ejecutarlo, simplemente no lo ejecutará.

Una última cosa. De momento toda la gestión de archivos solo la puede hacer desde el directorio raíz. Por ahora no gestiona directorios aunque como veremos tampoco es muy necesario ya que una vez que el sistema arranca, pasa el control al modulo sintetizado y estos ya cargan el MSXDOS donde si permite gestión de subdirectorios y demás.

A partir de aquí vamos a explicar módulo a módulo y su correspondiente juego de instrucciones.

El expansor de Slots.

El expansor de Slots se comporta igual que uno real. Podemos desactivarlo si no lo queremos y enchufar uno de nuestros dispositivos en el Slot principal. Y podemos activarlo e insertar todo lo que queramos en sus subslots.

Acordarse que no deja de ser hardware y que si el software no acepta subslots, este no va a funcionar correctamente. Muchas veces se puede engañar al software colocando el hardware que no funciona en el subslot cero y los demás dispositivos en el resto de ranuras.

El expansor sigue el siguiente tipo de esquema de configuración en el archivo CONFIG.CFG:

```
RANURA_MASTERSLOT = Ranura_Vacia  
RANURA_SUBSLOTO = Ranura_Vacia  
RANURA_SUBSLOT1 = Ranura_Vacia  
RANURA_SUBSLOT2 = Ranura_Vacia  
RANURA_SUBSLOT3 = Ranura_Vacia
```

Podéis hacerlo como queráis pero yo lo declaro todo de golpe, al inicio de la configuración, tal como lo veis. Recomendable dejar la estructura del CONFIG.CFG inicial.

Por defecto el sistema tiene configurado todos sus estados en ranura vacía. Si en RANURA_MASTERSLOT le cargamos algo, este anulará el expansor de Slots ya que entiende que deseamos esa unidad cargada en el slot principal, desactivando incluso el propio expansor de slots e ignorando el resto de configuración.

Para utilizar los subslots por tanto, debemos dejar como ranura vacía el Masterslot y colocar nuestros dispositivos en el resto de subslots.

Y poco mas del expansor. Mas adelante explicaremos cada uno de los dispositivos que podemos enchufar en estas ranuras.

Por supuesto soporta cuatro subslots tal y como veis en la configuración por defecto. No es posible mas debido a que la BIOS del MSX está programada para esto y no más, ya que por hardware no hubiera habido problema añadir mas subslots.

Comentar que podemos dejar la estructura inicial del CONFIG.CFG e ignorar comandos poniendo “//” doble barra. Esto sirve para ignorar lo que viene a continuación. También sirve para añadir comentarios, pero recordar el tope de capacidad de este archivo.

La unidad FLASHROM.

Como hemos detallado en las especificaciones, la unidad FLASHROM se comporta como una EPROM donde en su interior cargaremos y salvaremos las ROMs a ejecutar con los mappers que queramos.

La carga de ROMs la podemos hacer al arranque, mediante instrucciones en la configuración o con un software de carga desde el SO del MSXDOS (FROM.COM).

Para la carga desde la configuración mas adelante explicaremos como cargar y salvar bloques de RAM a ficheros.

Esta unidad sintetiza varias marcas de EPROMS. Depende del software que utilicemos, mediante identificación de acceso, el sistema arrojará un ID de la EPROM u otro.

Esto es automático y transparente para nosotros por lo que no debemos de preocuparnos de ello.

Para hacer una llamada a esta unidad desde el expansor de Slots sería:

`RANURA_SUBSLOTO = Cartucho_FlashROM`

Idem para el resto de subslots o ranura principal.

Recordad que podemos cargar hasta un total de cuatro unidades, una por cada subslot.

Automáticamente nos dará disponible un total de 4MBytes EPROM para el subslot seleccionado.

En este módulo también debemos de decirle el tipo de MAPPER a asignar a esa EPROM.

La instrucción sería:

```
MAPPER_SELECTO = KONAMI5
```

Donde el número del MAPPER_SELECT puede ser 0,1,2 y 3. El 0 es el primer subslot o slot principal depende de lo que hayamos configurado en el expansor. Si no le decimos que mapper, por defecto coge KONAMI5.

Después del “=” colocaremos el tipo de mapper que queramos. Estos son los disponibles:

Auto, KONAMI4, KONAMI5, ASCII8K, ASCII16K, SUNRISE, SINFOX,
ROM16K, ROM32K, ROM64K, RTYPE, ZEMINA6480, ZEMINA126, FMPAC,
CROSSBLAIM, SLODERUN

El modo SUNRISE no hace falta indicarlo como veremos a posteriori. Se selecciona automáticamente si seleccionamos ese módulo al igual que el SINFOX.

El modo Auto detectará el tipo de mapper a cargar según el archivo cargado por lo que es indispensable cargar primero el archivo y luego precederle por esta instrucción para que pueda evaluar que hemos cargado.

A excepción de lo anterior, las instrucciones podemos dárselas con el orden que queramos aunque aconsejo agruparlas por tipos. Todas las ranuras juntas, luego todos los mappers juntos, etc...

No hay mas comandos para este dispositivo. Ahora vamos a aprender como cargar y salvar bloques de memoria.

Cargar y salvar bloques de memoria.

Para completar el anterior dispositivo podemos solicitar una carga o salvado de un bloque de datos.

Para hacernos una idea, FLASHJACKS tiene reservado 4 bloques de 4Mbytes de memoria cada uno. Estos se llaman:

BLOQUE_RAM0, BLOQUE_RAM1, BLOQUE_RAM2 y BLOQUE_RAM3.

Cada bloque pertenece al subslot correspondiente.

Ademas dispone de una carga o salvado de unidad SRAM para el FMPAC (BLOQUE_SRAM0,1,2 y 3), estos fijos a 8kbytes.

Para hacer una carga de una ROM deberíamos colocar una instrucción tipo:

LOADFILE=SUNRISE.ROM T512KB BLOQUE_RAM0

Esto nos cargaría el archivo SUNRISE.ROM de un tamaño de 512KB en el bloque RAM N° 0.

Si tuviéramos configurado en esa ranura un Cartucho Flashrom lo que estamos haciendo es cargar el archivo en la Flashrom para su ejecución al inicio del arranque.

Los tamaños disponibles son:

T64KB, T128KB, T256KB, T512KB, T1MB, T2MB, T4MB.

Por defecto el tamaño preseleccionado es de 512Kbytes.

Para la carga aconsejo que coloquéis el de 4Mbytes ya que el sistema dejará de cargar en cuanto se finalice el archivo de carga. O sea, una ROM de 8Kbytes va a leer 8Kbytes y no 4Mbytes ya que no hay nada mas que leer en el archivo.

Con esta instrucción cada vez que inicie el MSX ya sea por el botón de power o el de Reset, va a hacer la carga del archivo en el bloque seleccionado. El orden de la instrucción debe ser el indicado en el ejemplo y no otro, separado por espacios.

Para salvar un bloque colocaremos una instrucción tipo:

```
SAVEFILE=SUNRISE.ROM T512KB BLOQUE_RAMO
```

Esta instrucción tiene múltiples utilidades.

Lo que va hacer nuestro MSX es que en el primer encendido va a interpretarlo como un LOADFILE. Nos va a cargar el archivo seleccionado en nuestro bloque de memoria.

Si este archivo no existe, no cargará nada. (lo mismo pasaría en el anterior caso del LOADFILE).

Una vez iniciado el MSX y después de haber realizado lo que queríamos en nuestro ordenador, si pulsamos el botón de Reset lo que realizará el cartucho es salvar el bloque seleccionado con la capacidad seleccionada en el nombre del archivo seleccionado. Si este archivo no existiera, lo crearía. Todo esto siempre en el directorio raíz, como hemos comentado en otros apartados.

Esto sucederá tantas veces pulsemos el botón de Reset.

Cuidado con pulsar el Reset varias veces antes de que finalice el proceso de salvado ya que interrumpiremos dicho proceso pudiendo dejar datos corruptos en nuestra tarjeta.

Aquí si importa la capacidad seleccionada porque nos va a grabar esa capacidad, ni mas ni menos.

¿Y que utilidad tiene esto?. Pues muchas.

Imaginemos que tenemos una FLASHROM vacía y que por software, desde el MSXDOS, le cargamos una ROM. Al resetear el MSX nos hará dos cosas. Primero nos salvará el bloque RAM con las modificaciones realizadas en la FLASHROM y segundo nos cargará la ROM en el arranque.

Cuando apaguemos el MSX, la RAM volátil desaparecerá pero al volver a encender el ordenador, al tener esta instrucción volcará la RAM salvada del archivo a la RAM, recuperando los datos nuevamente y haciendo la ilusión de tener una EPROM real no volátil con los datos intactos en el arranque de nuestro MSX.

Recordar que FLASHJACKS usa una RAM volátil y que al apagar el equipo todo se pierde. Con esta instrucción y con la rápida carga que es capaz de realizar, la ilusión de que parece haber una EPROM parecerá real. Si queremos borrar esta EPROM en el próximo arranque tan solo debemos borrar esta línea de configuración o el archivo salvado.

Esta instrucción también puede servir para hacer un volcado de la RAM_MAPPER de nuestro MSX a un archivo físico como veremos mas adelante.

Comentar también que para la SRAM se comporta de igual modo por lo que podremos renombrar el archivo para salvar partidas de distintos juegos sin machacar los escasos 8 bloques de salvado que nos permite esas 8Kbytes.

Tenemos una instrucción en la que evita la doble carga de ROM. Esto sirve para evitar estar cargando la misma ROM una y otra vez después de un Reset. Acordaros que la RAM mantiene los datos hasta que apaguemos el MSX.

Esta es la instrucción:

RELOADROM = OFF

La unidad RAM MAPPER.

Este dispositivo se encargará ni mas ni menos que de eso, de proporcionarnos memoria RAM extra siguiendo los standards de MSX por lo que para el usuario será transparente sin necesidad de instalar ningún controlador adicional.

Para su invocación sería una instrucción del tipo:

`RANURA_SUBSLOTO = Cartucho_RAM_Mapper`

Por defecto nos cargaría el máximo del bloque en el subslot, que serían los 4Mbytes de RAM.

Si queremos otra cosa por el motivo que sea cargaríamos también los siguiente:

`Capacidad_4MBRAMO = T4MB`

Donde el número de 4MBRAM indica el subslot y T es el tamaño de la RAM. Hay los siguientes tamaños seleccionables:

`T64KB,T128KB,T256KB,T512KB,T1MB,T2MB,T4MB`

Como veis todo sigue la misma filosofía de configuración. Podemos indicarlo todo según el orden que queramos pero aconsejo agruparlas por tipos. Todas las capacidades juntas, todas las ranuras juntas, etc... De este modo nos quedará todo mas ordenador y fácil de identificar y modificar.

Disponemos de un total de cuatro dispositivos como este por lo que una vez configurado uno en cada subslot, podríamos tener un total de 16 Mbytes de RAM para nuestro MSX. Algo poco útil ya que lo que si consume son subslots y nos quedaríamos sin ellos para poder añadir dispositivos.

La unidad MEGARAM.

Este dispositivo sintetiza una MEGARAM a todos sus efectos.

Para su invocación sería una instrucción del tipo:

`RANURA_SUBSLOTO = Cartucho_MEGARAM`

Por defecto nos cargaría el máximo que permite un dispositivo MEGARAM, que serían los 2Mbytes de RAM.

Si queremos otra cosa por el motivo que sea cargaríamos también los siguiente:

`Capacidad_4MBRAMO = T2MB`

Donde el número de 4MBRAM indica el subslot y T es el tamaño de la RAM. Hay los siguientes tamaños seleccionables:

`T64KB,T128KB,T256KB,T512KB,T1MB,T2MB`

Si seleccionamos 4MB van a aparecer 2Mbytes ya que el sistema MEGARAM permite como máximo eso.

Disponemos de un total de cuatro dispositivos como este por lo que una vez configurado uno en cada subslot, podríamos tener un total de 4 x 2Mbytes de MEGARAM para nuestro MSX. Aquí si que el soft necesario para acceder al MEGARAM debe ser compatible con los subslots y con múltiples dispositivos.

Ahora vamos a explicar la utilidad del SAVEFILE en dispositivos RAM y MEGARAM. Si hacemos un:

SAVEFILE=BACKUP.RAM T4MB BLOQUE_RAMO

Nos va a hacer una carga inicial del archivo BACKUP.RAM si este existiera en el subslot 0 o slot principal. Al pulsar Reset nos va a hacer un volcado de toda la RAM Mapper o MEGARAM que tengamos de ese subslot al archivo BACKUP.RAM

Esto sirve, en el caso de una RAM Mapper para capturar la RAM para su posterior análisis y saber que ha hecho un determinado programa, un juego, etc... Básicamente sirve para depurar un proceso ejecutado en nuestro MSX.

En el caso de la MEGARAM serviría para mantener intacta la información almacenada en la misma, teniendo disponible siempre dicha información y salvada en nuestra tarjeta.

Como hemos explicado en las EPROMS, ajustar el tamaño de la carga a nuestras necesidades porque el va hacer lo que le digamos aunque sea ilógico. Podemos cargar 64Kbytes en una RAM de 4Mbytes pero seremos conscientes que estamos en pérdida de datos.

Si solo quisiéramos tener la función de carga de archivo pero no de salvar, haríamos un LOADFILE.

Por último comentar que la pestaña de Write Protect de la tarjeta SD es funcional en todo momento por lo que si la tenemos activada, toda función de escritura sería ignorada. En MSXDOS el sistema ya nos indicaría que no podemos grabar pero en el sistema de FLASHJACKS no hay indicación alguna.

Comentar que tenemos la siguiente instrucción para ignorar la pestaña de escritura:

`IGNORA_WP = ON`

La unidad SINFOX compatible

Este dispositivo sintetiza el standard de las antiguas unidades SINFOX. Comentar que solo disponemos de un dispositivo SDCARD por lo que solo podremos usar un tipo de unidad sintetizada simultáneamente pese a que FLASHJACKS disponga de otros tipos. Siempre va a dar preferencia a la unidad SUNRISE ya que una vez activado este módulo, elimina el acceso de otro módulo.

Esta unidad se invocaría de la siguiente manera:

```
RANURA_SUBSLOTO = CARTUCHO_SINFOX
```

Automáticamente el mapper de la EPROM de ese subslot queda configurada para aceptar la ROM de la SINFOX.

Tan solo nos quedaría hacer un LOADFILE o SAVEFILE de la ROM en ese subslot seleccionado.

Por supuesto podríamos hacer una carga de la ROM desde el MSXDOS. Todas las opciones son permitidas.

No voy a explicar que podemos hacer con el soft para esta unidad ya que yo no soy propietario de su software pero contad que tiene sus limitaciones y que hay que considerarlas, sobretodo que hay que formatear la tarjeta con su formato. Precaución porque perderemos el CONFIG.CFG. Una vez formateada sugiero volcar en la tarjeta el CONFIG.CFG para continuar con la copia de archivos, discos, etc...o mejor aún, volcar el archivo ZIP a tal efecto. Como ya hemos dicho FLASHJACKS es compatible en el arranque con FAT12 por lo que él no le hará ascos al nuevo formateo (de hecho es compatible con FAT32 pese a que ninguna unidad de MSX lo soporta).

En el caso de SINFOX, acordaros de que estaremos limitados a esos 512MB de tarjeta en SDSC y es incompatible con las SDHC.

Comento que este tipo de unidades, como la de SUNRISE yo no soy propietario y esto es simplemente una compatibilidad con el standard que ellos crearon. Pueden usar su software según el criterio de los propietarios pero declino cualquier incompatibilidad futura o prohibición de uso. La compatibilidad la tenéis, pero el uso que hagáis es de vuestra responsabilidad.

Lo mismo para el resto del software y sistemas operativos. Al final del manual me extenderé un poco mas en este punto.

La unidad SUNRISE compatible

Este dispositivo sintetiza el standard de las unidades SUNRISE IDE en modo LBA hasta lo máximo permitido por el driver de NEXTOR SUNRISE IDE Ver 2.1 Alpha2, que serían 4GB y compatibilidad con SDSC y SDHC.

Incido como en el apartado anterior, que yo no soy propietario del software y en este caso todo queda limitado a las imposiciones de sus propietarios. Esta unidad internamente es una unidad con doble conversión transparente para nosotros y con acceso al vuelo. Esto significa que tenemos una comunicación bidireccional de los siguientes módulos:

SUNRISE ↔ IDE ↔ SDCARD ↔ NEXTOR

Cada módulo es independiente y tiene sus propias limitaciones del diseño original.

Todo esto lo hace al vuelo y del modo mas eficiente posible pero tiene sus pegas ya que este sistema ha sido pensado para utilizar una COMPACT FLASH y lamentablemente no hay retroalimentación de que todo ha ido bien. Dan un tiempo de acceso constante y no te puedes salir de esos parámetros la cual cosa eso a veces es imposible. Está pensado para una memoria Flash, con un bus paralelo y de tiempos constantes, y estamos accediendo a un dispositivo serie, con otro protocolo y con tiempos de esperas variables.

Es por ello que se ha creado un driver para FlashJacks donde se solventan todas estas cuestiones.

El driver es el FLASHIDE.ROM.

Básicamente es el driver de NEXTOR para Sunrise pero adaptado a nuestra unidad.

A parte de lo anterior lleva algunas vitaminas como la siguiente:

`FREQVDP = 60HZ`

Si colocamos esto en el `CONFIG.CFG`, cuando llegue a ejecutarse Nextor, este pasará de inmediato al forzado a 60HZ. Ídem para 50HZ y para OFF. Muy útil para aquellos juegos que estaban pensados para 60HZ.

Y ahora comentemos la configuración del cartucho SUNRISE o FLASHIDE: Bueno, vayamos a su invocación:

`RANURA_SUBSLOTO = CARTUCHO_SUNRISE`

Ídem que en la SINFOX. El mapper de la EPROM entra en modo SUNRISE y nos permite la carga de la ROM en el mismo subslot seleccionado mediante los parámetros que ya conocemos. Con `LOADFILE` cargaríamos la correspondiente ROM NEXTOR para unidades SUNRISE IDE.

Decir que la compatibilidad SUNRISE no es total si no se carga el driver específico `FLASHIDE.ROM`. El acceso de escritura de la ROM no me ha quedado claro y aunque algún software original pueda funcionar para escribir en la EPROM, no os lo puedo garantizar. De todos modos, el propósito de esta unidad es el acceso a los datos de la tarjeta SD, usando el driver mencionado eso queda cubierto.

Un último consejo es que si no te carga un disco puede ser por varias cosas. Prueba de hacer un apagado del MSX en lugar de un reset. Parece una tontería pero a veces quedan programas residentes que no permiten una carga del juego o programa. El pulsar `CTRL` al arranque también es útil.

Por cierto. El driver interno de la tarjeta SD permite inserción y extracción en caliente. Esto significa que con el MSX encendido podemos extraer la tarjeta, modificar, añadir archivos externamente, etc... y luego insertarla de nuevo. El lector la reiniciará sin problemas pero hay que tener cuidado con el MSXDOS porque este no sabe que hemos hecho esto. No cambiar a tarjetas diferentes ya que el MSXDOS puede dar serios fallos.

El sistema de sonido de FLASHJACKS

Antes de entrar en los módulos de sonido del MSX, voy a comentar un poco la filosofía de audio del sistema FLASHJACKS.

Siempre hemos visto que el sonido en nuestros MSX no ha sido muy cuidado y el tema de ruidos, compatibilidades con los niveles de sonido en determinados modelos de MSX, etc... han sido de lo más variopinto.

En este apartado he querido superar a los sistemas originales, rediseñando todo el concepto a un nivel superior. Para empezar, la mezcla de sistemas de audio se hace internamente en el cartucho, por lo que todos los sistemas tendrán un nivel sonoro adecuado y mezclable entre ellos. Además, se separa el audio interno del MSX con el externo, evitando ruidos, niveles inadecuados, etc...

En segunda instancia, he añadido un procesador de audio que hace las delicias de cualquier oyente. He visto muchos proyectos, tipo ONECHIP, ZEMMIX NEO o similares pero la conversión digital – analógica no me parece adecuada, de hecho ni el tratamiento de la señal, ni otras cosas donde no voy a entrar en polémica con ellos ya que obviamente hay que reconocer la proeza de tales proyectos.

Nuestro procesador sería capaz de entregar sonido stereo a 96Khz, con sampleos de 24bits por canal, con un ratio de ruido SNR de hasta 100dB. A parte incluye de serie varios sistemas de postprocesado, filtros, mezcla de canales, etc.. y todo en stereo y bidireccional (Si, también puede capturar audio) .Vamos, una locura.

También tiene la capacidad de manejar unos auriculares en su salida de línea de 16 o 32 Ohms con una potencia de 30 o 50mW respectivamente. Aconsejo unos SONY MDR ZX300. Son baratos y se escuchan bien.

Para conectar nuestro sistema de audio al MSX podemos usar un jack de 3,5mm y utilizar la salida de audio como auriculares o salida de línea.

Paralelamente, dentro del cartucho tiene un amplificador de dedicación exclusiva a la entrada de audio del Slot por lo que garantiza una máxima compatibilidad con el audio interno del MSX, si es que quisiéramos extraer el audio de ahí. Decir que obviamente se pierde el stereo ya que el hardware del MSX solo soporta Mono pero pese a esto, la conversión a Mono es adecuada y balanceada.

El sistema interno de FLASHJACKS utiliza especificaciones mas comedidas. El bus de datos de audio trata señales de 48Khz a 16bits por canal en PCM. O sea, el PSG, SCC y FM extrae su audio en este formato para su posterior mezclado y tratado. No hay SCC de 1 bit ni PSG de 4 bits. Todo es sintetizado en 16 bits. Comentar que las formas de ondas y densidad de bits de los instrumentos son los originales ya que no se ha querido alterar el sonido original, pero la mezcla de instrumentos, ondulados, etc... si que se trata a 16 bits.

Pero para mi lo que le da un toque superior al sistema original es el Stereo. No PseudoStereo ni bicanal adaptado. Es Stereo real y lo mejor de todo es que lo extrae del audio original. Podremos escuchar todos nuestros juegos habituales en Stereo, siendo pensados para un sistema Mono y la verdad, no lo hace nada mal.

El modelo sintetizado ha sido pensado en la síntesis de un espacio acústico de 180 grados. O sea, de nuestro oído izquierdo a nuestro oído derecho se abre un abanico de 180 grados y en esa ubicación se ordenan los instrumentos según su origen para lo que fueron concebidos.

O sea, los sistemas de percusión, baterías, etc... se ubican a 90° en el canal central mientras canales para instrumentos de acompañamiento como trompetas, violines, etc... se localizan a 45° y a 135° dando una sensación envolvente única. Su conjunto hace que verdaderamente sintamos los instrumentos desplegados en abanico delante de nosotros pese a que fueron compuestos para una audición en mono.

Espero que este sistema creado haga un precedente para futuros compositores y de este modo usar un standard y concebir partituras que exploten mejor estos canales de audio.

En los módulos siguientes detallaremos que canales están desviados y en que lado para tomarlo como futuras referencias.

El módulo PSG

Este módulo básicamente cuando se activa entra en modo esclavo. Es decir, captura los datos del bus del MSX y reproduce el sistema PSG en el audio de FLASHJACKS. No necesita configurar slot ni subslot ya que no lo necesita. Se invoca de la siguiente manera:

`SOUNDPSG = ON`

Por defecto se encuentra en estado apagado. No es necesario decírselo. Si tenemos el modo Stereo activado, los instrumentos se reparten en el entorno 180° de la siguiente manera:

- Canal B, solo tono por el lado izquierdo a 45°.
- Canal C, solo tono por el lado derecho a 135°
- Resto de canales y ruido por el lado central a 90°.

Esta configuración la he extraído de algún sitio donde he visto esta distinción para canales stereos PSG. En el PSG y el resto de sistema he intentado que la separación de canales no sea arbitraria y que si alguien ya separó el stereo, respetar lo máximo posible lo existente.

El PSG de serie emite a 112Khz pero se ha recortado a 20Khz ya que la banda alta es muy molesta.

Además, si se quiere un filtro extra, se puede añadir un filtro global, como veremos mas adelante al igual que el resto de parámetros.

Una última cosa. El PSG ya lo emite nuestro MSX, si usamos el nuestro y lo escuchamos a través del MSX, este quedará reforzado por lo que puede que notes una potenciación del PSG al resto de sistemas de audio. Si percibes esto y te parece molesto, cuando reproduzcas el audio por el MSX puede que te interese desactivar el PSG y dejar activado el resto de módulos para que la mezcla dentro del MSX sea mas lineal. Mencionar que el PSG del MSX no atraviesa nuestro sistema y no podrás oírlo por auriculares si desactivas este módulo.

El módulo SCC

Este módulo activa el sonido SCC. Dicho sistema si requiere que le digamos por donde escuchará las instrucciones. Aquí su llamada:

`SOUNDSCC = MASTERSLOT`
`DUALSCC = OFF`

Donde en MASTERSLOT podemos poner lo siguiente: (OFF, MASTERSLOT, SUBSLOTO, SUBSLOT1, SUBSLOT2 o SUBSLOT3).

No ocupa ranura por lo que permite duplicidad de recursos pero si entra en conflicto para que el software lo detecte correctamente. Lo lógico sería ponerlo en un subslot ocupado por un cartucho FlashROM. Esta es su configuración natural. Si lo dejamos suelto puede que el juego no nos detecte el sistema SCC al no haber la unidad FLASH.

También se puede dar el caso que el software solo mire en el slot principal y necesitemos tener cosas en los subslots. Podemos poner el SCC en MASTERSLOT con el expansor de Slots activo pero es probable que tengamos interferencias acústicas con instrucciones que no van dirigidas al SCC ya que estas van a módulos ubicados en los subslots. Esto es una solución de compromiso y que normalmente funciona pero con la salvedad de que a veces se escuchan ruidos los cuales son la comunicación al resto de componentes. Si hacemos esto, colocad una FLASHROM en el SUBSLOTO para que el programa que no acepta expansor de slots lo tenga lo mas fácil posible de ubicar.

El dualSCC está apagado por defecto. Este sirve para colocar dos SCC en dos subslots distintos y que un programa pueda acceder a un doble SCC simultaneo.

Si tenemos el modo Stereo activado, los instrumentos se reparten en el entorno 180° de la siguiente manera:

- Canal A, C y E por el lado izquierdo a 45° .
- Canal B y D, por el lado derecho a 135°

Es el modo mas natural que he visto y se adapta muy bien a la música existente.

El módulo FM

Este módulo activa el FM audio. No ocupa ranura pero si que necesita que por algún lado le carguemos el controlador FMPAC.ROM. Esto ya es a decisión de cada uno. Podemos crear una FLASHROM en un subslot y hacer una carga en el arranque del FMPAC.ROM o podemos cargarla a posteriori de cualquier otro modo. Algunos juegos no necesitan la ROM. Esto ya es la decisión de cada uno. Lo mejor es cargar una FlashROM con mapper FMPAC y cargar el FMPAC de Panasoft.

Con esto tendremos una reproducción fiel de ese cartucho, incluido su CALL FMPAC o su SRAM interna, como hemos hablado antes.

Para llamar al módulo para su activación:

`SOUNDFM = ON`

Por defecto estará desactivado.

Si tenemos el modo Stereo activado, los instrumentos se reparten en el entorno 180° de la siguiente manera:

- Canal 1, 3 y 5 por el lado izquierdo a 45°.
- Canal 0, 2 y 4, por el lado derecho a 135°
- Resto de canales y percusión por el canal central a 90°

Al alternar canales pares e impares lo que hacemos es que normalmente las partituras están escritas en canales consecutivos y si algún instrumento se quiere reforzar, suelen duplicarlo en el canal consecutivo a este. Con este sistema obtenemos que ambos instrumentos quedan separados produciendo una sensación espacial única.

Personalmente es la configuración que más se adapta al mayor repertorio disponible. Obviamente habrán canciones que esta configuración no será la mas adecuada.

Para partituras nuevas, si se tiene en cuenta este esquema, se podrá componer a antojo la desviación acústica del stereo.

El bus de audio

Aquí vamos a explicar el resto de parámetros que afectará de forma genérica al audio.

Empecemos por el Stereo:

SONIDO = STEREO

Tiene dos opciones, *STEREO* o *MONO*. En mono lo escucharemos en su composición original. Por defecto estará en *STEREO*.

Ahora le indicaremos el volumen de auricular o salida de línea:

VOLUMEN = 5

Las opciones disponibles son 0,1,2,3,4 y 5. Cero es sin sonido y 5 máxima amplitud. Los valores intermedios dan atenuaciones de mayor a menor. Esto no afecta al volumen del bus interno del MSX. La amplificación interna del MSX está calibrada a un nivel específico para mayor compatibilidad con el standard. Si optamos por un cero, se seguirá escuchando a través del MSX interno. Por defecto el valor es 5.

El filtro por hardware:

AUDIOFILTRO = ON

Por defecto estará apagado. Si lo seleccionamos, atenuaremos los “glitches” y otros ruidos pero también perderemos respuesta en frecuencia, escuchando los tonos mas graves.

Si queremos una configuración que respete al máximo el sistema original se debería activar el SCC, el FM, el sonido en MONO y el AUDIOFILTRO activado. Si deseas profundidad y pureza pues ya sabes, en STEREO y sin filtro.

A continuación, el mezclador de sonidos:

Bus de audio interno	Jack salida de audio
PSGVOLIN = 0	PSGVOLOUT = 3
SCCVOLIN = 2	SCCVOLOUT = 3
FMVOLIN = 2	FMVOLOUT = 3
EXTVOLIN = 2	EXTVOLOUT = 2

Las opciones son de 0 a 3, siendo el 0 silencio o bajo nivel y el 3 máximo. Cada chip de audio lleva su propia ganancia.

El sistema detecta si nos conectamos al jack de salida y permuta de la configuración "IN" a la "OUT". De este modo podemos escuchar con una configuración sin PSG y niveles adecuados para una reproducción desde el bus de audio de nuestro propio MSX y su PSG interno, con la particularidad que si conectamos al Jack un conector, el sistema permuta automáticamente a la configuración "OUT", ajustándose al nuevo sistema de audición.

El EXTVOL está pensado para que nosotros aportemos el sistema de audio externo que queramos para su mezclado y nivelado totalmente digital, sin pérdida de calidad e incorporado al resto de chips de audio.

Esto está pensado para añadir una Moonsound, Moonblaster o similar.

Incluso puedes enchufar tu teléfono y poner tus MP3 favoritos. En algunos juegos diseñados para SCC, si quitas el volumen de este, puedes jugar con tu música favorita y el sistema mezcla los disparos y efectos del PSG.

Otros aspectos configuración general

Y por último un comando para apagar el estado de transferencia RAM del LED ROJO:

LEDROJO = ON;

Solo admite dos estados, encendido(ON) y apagado (OFF). Por defecto estará encendido.

Cuando está encendido veréis un parpadeo del led mientras trabaja. Este nos indica dos cosas.

Que el sistema está funcionando correctamente y la disponibilidad de la RAM para otros usos.

A parte de toda la gestión de los módulos, la RAM de FLASHJACKS tiene el privilegio de tener multiacceso y acceso compartido simultaneo. Esto quiere decir que a parte de atender las necesidades del MSX también estamos accediendo simultáneamente para otros menesteres.

Internamente tenemos un pequeño bucle de lectura y escritura en la RAM. Va leyendo y escribiendo posiciones incrementales y va comparando el resultado. Un solo byte erróneo y el led quedaría congelado. Cada parpadeo completo significa que ha hecho una comprobación completa de 256Kbytes de memoria RAM. La velocidad del parpadeo nos indica la disponibilidad de acceso a esta RAM ya que el MSX tiene prioridad siempre. Si la vemos ir mas lenta significa que el MSX la está ocupando más y viceversa. Podréis observar la cantidad de parpadeos que hace para imaginad la tasa de transferencia que tiene nuestro sistema ya que estos accesos solo son una parte, el MSX hace mucho acceso recurrente.

Ahora indicaremos una configuración ejemplo para nuestro archivo CONFIG.CFG:

```
RANURA_MASTERSLOT = Ranura_Vacia
RANURA_SUBSLOTO = CARTUCHO_SUNRISE
RANURA_SUBSLOT1 = Cartucho_RAM Mapper
RANURA_SUBSLOT2 = Cartucho_FlashROM
RANURA_SUBSLOT3 = Cartucho_FlashROM
RELOADROM = OFF
LOADFILE = FLASHIDE.ROM T512KB BLOQUE_RAM0
LOADFILE = FMPAC.ROM T512KB BLOQUE_RAM3
MAPPER_SELECT2 = KONAMI5
MAPPER_SELECT3 = FMPAC
IGNORA_WP = ON
SAVEFILE = SAVESRAM.RAM BLOQUE_SRAM3
Capacidad_4MBRAM1 = T4MB
FREQVDP = 60HZ
SOUNDSCC = SUBSLOT2
SOUNDPSG = ON
SOUNDFM = ON
SONIDO = STEREO
VOLUMEN = 5
AUDIOFILTRO = OFF
LEDROJO = ON
PSGVOLIN = 0
SCCVOLIN = 2
FMVOLIN = 2
EXTVOLIN = 2
PSGVOLOUT = 3
SCCVOLOUT = 3
FMVOLOUT = 3
EXTVOLOUT = 2
```

Este sería un ejemplo donde tenemos una configuración muy completa.

Dispondríamos de un cartucho SUNRISE FLASHIDE, de una memoria RAM de 4Mb, de dos Cartuchos FLASHROMs donde el SCC iría conectado al primer FlashROM disponible, que es lo mas compatible. En el segundo FLASHROM lo tendríamos para cargar el FMPAC.ROM y en la SUNRISE también su correspondiente ROM. El primer FLASHROM quedaría libre para usos. Todo en Mapper KONAMI5, que es el mapper defecto en la mayoría de aplicaciones.

También tendríamos todos los dispositivos de audio activados, en stereo y sin filtros. Y por último el sistema de comprobación de la RAM activado también.

Carga de ROMS desde MSXDOS. "FROM.COM"

A continuación describiremos una alternativa a la carga a través del sistema de configuración de FLASHJACKS.

La idea del sistema se basa en dos modos. El modo manual donde configuras a antojo y el modo automático que con una configuración standard puedas acceder a la mayor parte del catálogo de programas y juegos que se hicieron para MSX.

Con FROM.COM se pretende abordar ese sistema automático de pulsar y listos. La configuración standard está optimizada para tener un poco de todo. En el subslot 0 tenemos la unidad de disco, en el subslot 1 RAM, en el 2 una FlashROM libre y en el 3 un FMPAC completo el cual podemos tirarlo para que conviva una segunda FlashROM.

Una vez dicho esto comentemos sus opciones:

FROM NOMBRE1.ROM NOMBRE2.ROM

En este modo, si no le colocamos opciones, aceptará la carga de dos ROMS simultaneas. Estas las colocará en los dos primeros FlashROMs que encuentre. En nuestro caso se sitúan en el subslot2 y 3. La unidad de disco tiene prohibida convertirse en FlashROM.

Si solo ponemos una ROM(lo habitual), nos cargará en el subslot 2 dejándonos el FMPAC en el subslot 3.

Al no haber parámetros, cargará el mapper en modo AUTO.

El modo AUTO elige por ti el mapper que mas se adapta a esa ROM. Incluso se autoconfigura a Slot primario si observa en el código de la ROM alguna incompatibilidad con los subslot.

No hemos hablado todavía del sistema AUTO de los mappers.

Cuando se elige este modo, internamente se analiza las ROMs con un doble método. Por un lado existe un método tipo heurístico, donde busca marcas en la ROM tipo salto para averiguar como se comporta y adaptar el mapper mas acertado a este. Este sistema hace un cruzamiento múltiple de estas marcas y la ganadora es la seleccionada por el sistema. Además, se cruzan con una pequeña base de datos donde termina de rellenar esos huecos donde se tienen dudas.

Realmente se comporta parecido a lo que hace un antivirus.

El proceso se realiza en el Hardware de Flashjacks a velocidad de vértigo durante la carga de la ROM sin que esta se vea afectada en velocidad.

Además de la búsqueda anterior, el sistema intenta buscar marcas de incompatibilidad ante subslots y en el caso de encontrarlos, tira la configuración del expansor de Slots, dejando como primario la primera ROM de carga.

Por supuesto el sistema no es infalible y es por ello que FROM.COM tiene un sistema manual. Vayamos a por este segundo modo de carga:

FROM /Sxx /My NOMBRE.ROM

Aquí solo nos permite la carga de una ROM con parámetros.

En /Sxx podremos forzar Slot y Subslot, siempre y cuando exista una FlashROM de FlashJacks en esa ranura.

Si colocamos solo un dígito en lugar de dos, le estamos diciendo al sistema que queremos un forzado de la ROM al Slot principal por lo que anulará el expansor de Slots.

En /My nos permite el formado del mapper donde “y” puede valer:

0: AUTO	1:KONAMI5	2:ASCII8K
3: KONAMI4	4:ASCII16K	5:SUNRISE
6: SINFOX	7:ROM16K	8:ROM32K
9: ROM64K	A:RTYPE	B:ZEMINA6480
C: ZEMINA126	D:FMPAC	E:CROSSBLAIM
F: SLODERUN		

Cada nombre intuye el tipo de mapper que cargará.

Todos estos mappers, están expandidos a lo máximo que puedan direccionar ese formato, superando el direccionado y capacidad para que fueron pensados. El mapper que tiene capacidad para direccionar los 4Mb (o 32Mbits) de Flashjacks es el tipo ASCII16K.

Una vez pulsemos ENTER ocurre la magia. Aunque aparentemente te da un resultado y el MSX sigue funcionando como si nada, el dispositivo ya tiene esas instrucciones en la recámara para el próximo reset.

FROM.COM no funciona como otros cargadores de EPROM donde realiza el volcado a través de la CPU del MSX.

FROM.COM es un programa que le manda instrucciones a FLASHJACKS, donde después de un reset, la bestia del hardware interno, procesará a una velocidad infinitamente superior a lo que podría hacer el Z80.

Este modo de funcionamiento seguirá así hasta que no realicemos un Power OFF.

Indicaros que se pueden usar otros cargadores de EPROM ya que nuestra FlashROM se va a comportar como si de una EPROM original se tratase. El sistema FlashJacks dispone internamente de un sistema adaptativo que hace dar la referencia adecuada de EPROM dependiendo como se lo pida el software.

Este proceso es lento ya que es el Z80 el que se encarga de transferir e incluso parchear nuestra ROM para que funcione en la EPROM(en nuestro caso una RAM simula esta EPROM).

Por cierto, nuestro sistema es puro hardware y se comporta como tal. Nunca nos va a parchear una ROM para que cargue emulando un mapper, que es lo que hacen los sistemas tradicionales. FlashJacks aporta ese mapper como si fuese real (bueno, de hecho lo es).

Y poco mas que comentar de FROM.COM. Acordaros de hacer Reset para ejecutar lo solicitado y Power para volver a empezar.

Acordarse que en las configuraciones de autos salvado de ROMs o SRAM, ese guardado lo hace al siguiente reset. Si queremos salvar una SRAM a archivo, debemos hacer un reset a posteriori de haber jugado y salvado.

Un poco de MultiMente

Aunque este programa se sale un poco de FlashJacks, quiero comentarlo ya que está muy extendido su uso, por lo que se ha añadido a los archivos de la tarjeta SD.

Comentar que cuando carguemos la SD, esta vendrá con un mix de archivos diversos configurados para su uso directo desde MSXDOS, o a través de MultiMente.

El arranque es automático, a no ser que estemos en un MSX1, donde el cual en la carga final nos lanzará un comando inválido. En este caso, no os queda mas remedio que usar el teclado y teclear comandos de MSXDOS como toda la vida.

Comentar que aunque no funcione el MultiMente en un MSX1, podremos acceder a todas las características de FlashJacks sin mayor problema.

Bueno. Dentro de Multimente, todo es fácil. Pulsamos ENTER en la ROM que deseemos y esta nos la cargará.

Comentar que los DSK serán cargados a través de Nextor mediante el EMUFILE. Para este caso ídem que lo anterior, ENTER y a disfrutar.

Si seleccionamos varios archivos y ENTER, nos cargará múltiples archivos.

En el caso de las ROMs acordaros que hay un máximo de dos archivos.

Y poco mas. Del resto seguramente ya estaréis acostumbrados.

Otro poco de SDs y archivos

Os comento un poco lo que os vais a encontrar en la SD que viene incluida. La tarjeta ya viene formateada en FAT16 desde el BASIC de un MSX mediante el comando CALL FDISK.

Las instrucciones de esto podéis encontrarlos en los leeme.txt de la propia tarjeta.

Dentro viene una estructura configurada para un funcionamiento automático. En el directorio raíz aparecen las ROMs y configuraciones para la carga de FLASHJACKS.

También aparece directorios para colocar las ROMS, DSK, etc. Además aparece un directorio de Backup donde hay dos ZIPs. Uno de ellos es para restaurar el contenido de la tarjeta para un funcionamiento SUNRISE. Y el otro ZIP es para restaurar en un modo funcionamiento SINFOX.

En el leeme.txt de la tarjeta SD explica con detalle cómo llevar a buen puerto ese formateo a esos sistemas.

Comentar que SINFOX es un sistema cerrado y que no ha sido alterado de ningún modo por lo que irá exactamente como uno original. Esto quiere decir que ciertos parámetros de carga como el cambio a 60Hz no funcionará ya que no hay un driver de FLASHJACKS que se lo diga.

Y el sistema SUNRISE IDE de NEXTOR pues nos brindará todas las disposiciones del Hardware ya que se ha adaptado el driver original a este dispositivo (FLASHIDE).

Una última cosa. Acordaros de dejar los RESCUESU.ROM y RESCUESI.ROM. Estos sirven para cuando pulses la tecla HOME (tecla de pánico) durante mas de cinco segundos, en la próxima carga, se autoseleccione una configuración de rescate y, entre otras cosas, cargue las ROMs mencionadas. Esta cualidad es útil para tocar un CONFIG.CFG alterado y no queramos usar un PC para arreglarlo.

Epílogo

Y llegamos al final. Podréis observar la flexibilidad del cartucho donde cada uno puede adaptarlo a sus necesidades ocupando solo un SLOT. Como curiosidad, podéis conectarlo en un expansor de SLOTS pero entonces solo podréis configurar la ranura MASTERSLOT por razones obvias.

Agradecer y pedir disculpas a la vez, a toda la comunidad MSX. Agradecer por toda la información disponible que hay en la red y me ha permitido realizar este trabajo. Y pedir disculpas a todo aquel que se pueda haber sentido ofendido por ver que este aparato hace parte de lo que él creó. No ha sido mi intención plagiar ni nada por el estilo. Solo he cogido de la red el standard de funcionamiento de dispositivos electrónicos, los cuales si observáis no están en producción. Me hubiera gustado obviamente añadir ciertos dispositivos que actualmente están a la venta, y que además, ciertos emuladores de MSX ya tienen emulados, pero por respeto a sus propietarios no he realizado.

Todo el software, nombres, etc... que aquí se mencionan pertenecen a sus respectivos propietarios y su compatibilidad con este dispositivo se basa en el protocolo de funcionamiento por hardware y no por parches de sus programas. Su trabajo queda intacto y su uso y distribución depende del usuario final de este dispositivo.

También quiero comentar la continuidad y soporte de este aparato. Por supuesto que me gustaría dar soporte y continuar expandiendo las posibilidades del mismo pero desgraciadamente no os lo puedo garantizar. Esto está realizado íntegramente en mi tiempo libre y de ocio, de forma altruista. Desconozco qué usos y disponibilidad voy a tener en un futuro, y como he dicho, no soy empresa, no tengo un comercio ni esto es un negocio.

Las unidades que circularán seguramente serán muy limitadas debido a varias restricciones. El coste de producción es elevado tanto económico como en tiempo. Hay piezas que no están disponibles todo el año debido a su enorme fluctuación de precios.

Se trata de un aparato de un aficionado y el que lo compra debe saber lo que conlleva eso. Si quieres seguridad, soporte, continuidad, etc... mejor compra otros equipos del mercado que te van a dar esa estabilidad que buscas. Esto mas bien está dirigido a personas inquietas, que le gusta el cacharreo, experimentar, etc... y quieren un plus en sus sistemas pero sabiendo sus limitaciones.

Yo intentaré añadir mas sintetizaciones y actualizaciones de bugs que se vayan encontrando pero no os puedo garantizar un soporte continuado. Las actualizaciones se tienen que hacer presencialmente ya que necesito conectarme al dispositivo físicamente. Intentaré estar en las reuniones de usuarios de MSX pero es algo que tampoco puedo garantizar.

Como veis, he sido muy estricto en la responsabilidad del mismo. Pongo por delante de todo la verdad y no quiero que nadie tenga desengaños ni ideas equivocadas. Seguramente suplirá sobradamente sus expectativas pero prefiero ser estricto y comentar todos los peros de este sistema.

Y poco mas que decir. A disfrutarlo y a sacarle partido.

Saludos.

AQUIJACKS 2018.