

ZX80S+

Manual



Índice

Introducción	1
Agradecimientos	1
Puertos y Conectores	2
ZXDOS+	2
gomaDOS+	3
Descripción	3
Configuración Inicial	4
Formato de la tarjeta microSD	5
Windows	7
macOS	8
Linux	10
Teclado	11
Teclado de membrana en gomaDOS+	11
Teclado PS/2	14
Teclas especiales y botones	16
esxdos	17
BIOS	19
Main	20
ROMs	21
Upgrade	21
Boot	22
Advanced	23
Exit	24
ZX Spectrum	25
ROMs	27
DerbyPro	27
CargandoLeches	29
POKEs	29
Preparación de cintas de carga ultrarrápida	30
SE Basic IV	31
Otras ROMs	33
Formato avanzado de la tarjeta microSD (+3e)	34
Windows	34
macOS	35
Linux	37
+3e	40
Comandos de esxdos	41
Guía básica	41

Comandos para ZXDOS+	43
Wi-Fi	45
Network tools for ZX-Uno	46
FTP-Uno	47
UART Terminal	48
Creación de vídeos RDM (RaDastan Movie)	49
FUZIX.....	50
Cómo compilar.....	50
Cómo utilizarlo	51
Actualizaciones	55
BIOS.....	55
ROMs.....	55
Cores	56
esxdos.....	57
Memoria Flash	57
Otros cores	58
ZX Spectrum 48K (Kyp).....	58
Formato de Tarjeta microSD.....	58
Teclado	58
Teclas especiales y botones.....	58
ZX Spectrum 128K (Kyp).....	59
Formato de Tarjeta microSD.....	59
Teclado	59
Teclas especiales y botones.....	59
ZX Spectrum Next	60
Formato de Tarjeta microSD.....	61
Teclado	61
Teclas especiales y botones.....	61
Guía básica	62
MSX	64
Formato de Tarjeta microSD.....	64
Teclado	66
Teclas especiales y botones.....	66
Guía básica	67
MSXCTRL.....	68
Otros.....	68
Amstrad CPC 6128	69
Formato de Tarjeta microSD.....	69
Teclado	69
Teclas especiales y botones.....	69
Guía básica	70

Acorn Atom	71
Formato de Tarjeta microSD.....	71
Teclado	72
Teclas especiales y botones.....	72
Guía básica	73
Commodore 64	74
Formato de Tarjeta microSD.....	74
Teclado	74
Teclas especiales y botones.....	74
Guía básica	75
Phoenix.....	77
Formato de Tarjeta microSD.....	77
Teclado	77
Teclas especiales y botones.....	77
Guía básica	78
Pong.....	79
Formato de Tarjeta microSD.....	79
Teclado	79
Teclas especiales y botones.....	79
Guía básica	80
NES	81
Formato de Tarjeta microSD.....	81
Teclado	81
Teclas especiales y botones.....	81
Guía básica	82
Computers Lynx.....	83
Formato de tarjeta microSD.....	83
Teclado	83
Teclas especiales y botones.....	83
Guía básica	84
Colecovision	85
Formato de Tarjeta microSD.....	85
Teclado	85
Teclas especiales y botones.....	85
Guía básica	86
Atari 2600	87
Formato de Tarjeta microSD.....	87
Teclado	87
Teclas especiales y botones.....	87
Guía básica	88
Videopac	89

Formato de Tarjeta microSD	89
Teclado	89
Teclas especiales y botones	89
Guía básica	90
Cambio del charset de la ROM de VDC	91
ZX81	92
Formato de Tarjeta microSD	92
Teclado	93
Teclas especiales y botones	93
Guía básica	94
PC XT	96
Formato de Tarjeta microSD	97
Windows	97
Linux	98
macOS	98
Teclado	99
Teclas especiales y botones	99
Guía Básica	99
Chip-8	100
Formato de Tarjeta microSD	100
Teclado	100
Teclas especiales y botones	100
Guía Básica	101
Oric Atmos (Kyp)	102
Formato de Tarjeta microSD	102
Teclado	102
Teclas especiales y botones	103
Guía básica	104
Otro Hardware	105
Codificadores rotatorios	105
Conexión	105
Configuración del Core de Pong	106
Carga desde cinta	107
Reproductor de cassette	107
Ordenador	107
PlayTZX	107
Teléfono móvil, tableta, reproductor de sonido MP3, etc.	108
Conversión a fichero de audio	109
Miniduino	110
Puertos y Botones	110
Preparación	111

Uso	112
Creación de ficheros TZX o TSX desde otros formatos	114
Actualización de firmware Maxduino	115
Addon RTC+I ² S+PIzero	120
I ² S	120
Cores con soporte	120
Raspberry Pi	121
NextPI	121
Conexión	121
Uso desde NextZXOS	123
Solución de problemas	124
Gestión de imágenes de firmware	124
zx123_tool	124
Recuperación del firmware	128
Preparación del cableado	128
Recuperación usando una Raspberry Pi	129
Recuperación usando macOS y Cable USB-Blaster	135
Recuperación usando Windows y Cable USB-Blaster	140
Referencias	144
Spectrum	144
Scan Codes	144
Registros E/S para control de ZX DOS+	145
\$00 MASTERCONF	146
\$01 MASTERMAPPER	147
\$02 FLASHSPI	147
\$03 FLASHCS	147
\$04 SCANCODE	147
\$05 KEYSTAT	148
\$06 JOYCONF	148
\$07 KEYMAP	148
\$09 MOUSEDATA	149
\$0A MOUSESTATUS	149
\$0B SCANBLCTRL	150
\$0C RASTERLINE	151
\$0D RASTERCTRL	151
\$0E DEVCONTROL	152
\$0F DEVCTRL2	153
\$10 MEMREPORT	153
\$40 RADASCTRL	154
\$41 RADASOFFSET	154
\$42 RADASPADDING	154

\$43 RADASPALBANK	155
\$80 HOFFS48K	155
\$81 VOFFS48K	155
\$82 HOFFS128K	155
\$83 VOFFS128K	156
\$84	156
\$85 VOFFSPEN	156
\$A0 DMACTRL	156
\$A1 DMASRC	156
\$A2 DMADST	156
\$A3 DMAPRE	157
\$A4 DMALEN	157
\$A5 DMAPROB	157
\$A6 DMASTAT	157
\$C6 UARTDATA	157
\$C7 UARTSTAT	157
\$C8 - \$DF RESERVED	157
\$F0 SRAMADDR	158
\$F1 MADDRINC	158
\$F3 VDECKCTRL	158
\$F7 AUDIOMIX	158
\$FB AD724	159
\$FC COREADDR	159
\$FD COREBOOT	159
\$FE SCRATCH	159
Enlaces	161

Introducción

ZXDOS+ y gomaDOS+ son la continuación de [ZX-Uno](#) un proyecto de hardware y software basado en una placa FPGA programada para trabajar como un ordenador ZX Spectrum, y creado por el equipo de ZX-Uno: Superfo, AVillena, McLeod, Quest y Hark0.

Con el paso del tiempo, el proyecto ha ido creciendo, de forma que es posible instalar distintas configuraciones de software (cores) en la memoria flash de la FPGA, y que trabajan como otros sistemas distintos del ZX Spectrum, pudiendo elegir al arrancar el ZXDOS+ la configuración que se desee de entre todas las instaladas.

La página oficial de ZXDOS+ y gomaDOS+ es <https://zxdos.forofpga.es>.

La mayoría de las funciones y características de ZXDOS+ y gomaDOS+ son las mismas, así que, en este documento, se hablará, en general, de ZXDOS+, indicando las diferencias con gomaDOS+ donde sea necesario.

Agradecimientos

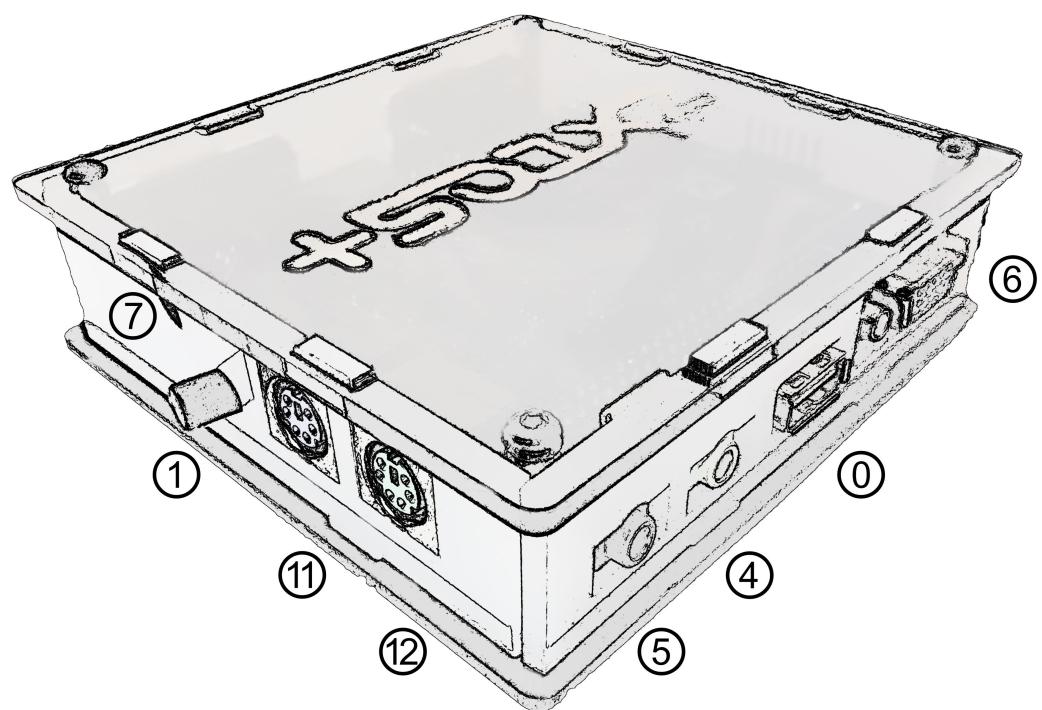
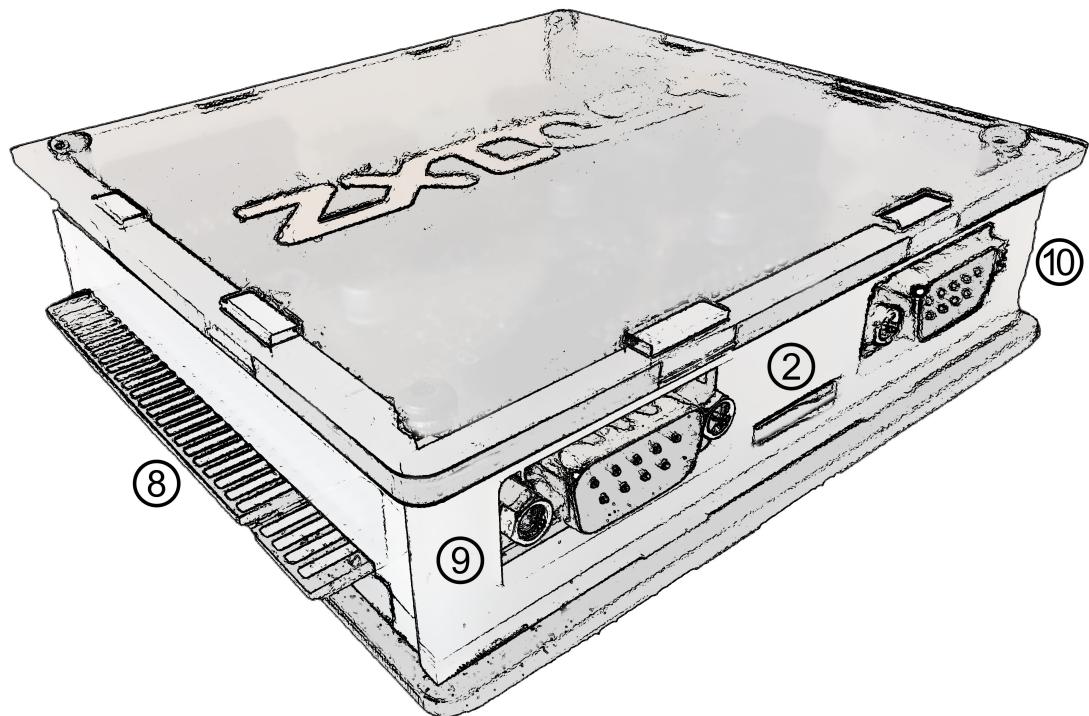
Gran parte del contenido de este documento se basa en información compartida anteriormente:

- En [foroFPGA](#)
- En el [foro de ZX-Uno](#)
- En los varios FAQ existentes, principalmente la versión original [de @uto_dev](#), y la versión más moderna [de @desUBIKado](#)
- En los canales oficiales de Telegram de [ZX-Uno](#) y [ZXDOS](#)

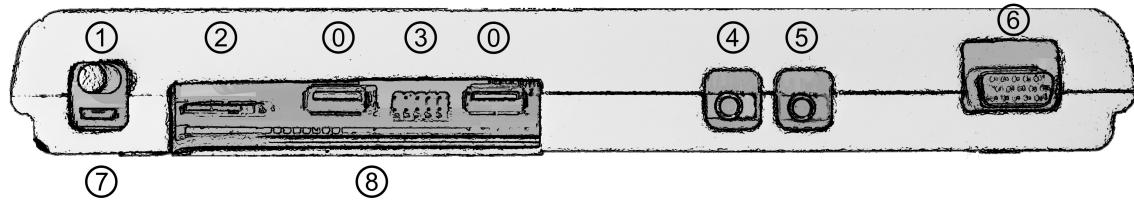
Sin el trabajo previo de todas estas personas (y más), ese manual no podría existir.

Puertos y Conectores

ZXDOS+



gomadOS+



Descripción

1	Interruptor
2	Ranura microSD
3	JTAG y Joystick
4	Salida de Sonido
5	Entrada de Sonido
6	Salida RGB/VGA
7	Enchufe de Alimentación
8	Puerto de expansión
9	Puerto de Joystick Izquierdo
10	Puerto de Joystick Derecho
11	Puerto de Teclado PS/2
12	Puerto de Ratón PS/2
0	Puerto USB (PS/2)

Configuración Inicial

Para poder poner en marcha un ZX DOS+ o gomaDOS+ hace falta, al menos, lo siguiente:

- Un cargador USB, una TV u otro dispositivo que ofrezca alimentación USB
- Un cable y un monitor VGA
- Un teclado PS/2 (en el caso de ZX DOS+)

Para poder aprovechar todo su potencial, es útil tener también:

- Una tarjeta microSD, de 32GB de capacidad o menos
- Unos altavoces de PC para conectar a la salida de audio, o un cable jack-stereo a dos conectores RCA rojo/blanco para conectar a la TV (opcional en gomaDOS+, ya que tiene beeper incorporado)
- Un joystick norma Atari, como por ejemplo, un gamepad DB9 de Megadrive (se necesita el adaptador de joystick en el caso de gomaDOS+)
- Un ratón PS/2 (se necesita adaptador PS/2 a USB en el caso de gomaDOS+)
- Un cable con un jack estéreo de 3,5 mm en un extremo y los dos canales de sonido divididos en dos salidas mono en el otro, si se quiere usar algún dispositivo de reproducción y/o grabación de audio, como por ejemplo un Miniduino ([ver la sección correspondiente más adelante](#)), un PC/Mac/Raspberry PI, etc. o un reproductor/grabador de [cassette](#). El canal derecho se utiliza como entrada (EAR) y el canal izquierdo se puede usar como salida de grabación (MIC).

Si se utiliza un adaptador VGA a [Euroconector o SCART](#) activo, este emplea pines no utilizados del puerto VGA para transmitir el audio, pero sólo si no hay nada conectado al puerto jack de 3,5 mm de salida de sonido.

Formato de la tarjeta microSD

La siguiente tabla resume los requisitos particulares de los distintos cores que utilizan la tarjeta microSD.

Core	FAT 16	FAT 32	+3e	Tipo de Partición Primaria	Particion es Extra	Tipo de Acceso	Notas
ZX Spectrum EXP	Sí	Sí	Sí	Cualquiera	Sí	Completo	Utiliza esxdos de SPI Flash
ZX Spectrum Kyp 48K	Sí	Sí	No	Cualquiera	Sí	Completo	Utiliza esxdos embebido
ZX Spectrum Kyp 128K	Sí	Sí	No	Cualquiera	Sí	Completo	Utiliza esxdos embebido
ZX Spectrum Next	Sí	Sí	No	Cualquiera	Sí	Completo	Puede utilizar esxdos en la microSD
MSX	Sí	No	No	0x06 (16-bit FAT)	Sí (FAT16)	Completo	No compatible con PC XT
Amstrad CPC 6128	No	Sí	No	0x0b (Win95 FAT-32)	No	Sólo imágenes .DSK de disco en la raíz de la SD	Partición de 4G o menos y cluster de 4096
Acorn Atom	Sí	No	No	Cualquiera	No	Atom software archive	
Commodore 64	Sí	Sí	No	Cualquiera	No	Sólo imágenes (.D64 y .TAP)	
Phoenix							No utiliza la MicroSD
Pong							No utiliza la MicroSD
NES	Sí	Sí	No	Cualquiera	No	Sólo ROMs (.NES)	
Computers Lynx							No utiliza la MicroSD
Colecovision	Sí	No	No	Cualquiera	No	Sólo ROMs (.ROM)	
Atari 2600	Sí	No	No	Cualquiera	No	Sólo ROMs (.BIN)	
Videopac	Sí	No	No	Cualquiera	No	Sólo ROMs (.BIN)	
ZX81	Sí	Sí	No	Cualquiera	No	Sólo imágenes (.P)	
PC XT	Sí	No	No	0x06 (16-bit FAT)	No	Completo. Necesita tener instalado DOS	No compatible con MSX
CHIP-8	Sí	Sí	No	Cualquiera	No	Sólo ROMs (.BIN o .CH8)	

Por tanto, en este momento, para poder utilizar todos los cores, sería necesario tener, al menos, tres tarjetas distintas:

- Tarjeta con una partición primaria FAT32. Requerida por el core de [Amstrad CPC 6128](#)
- Tarjeta con una partición primaria FAT16 y MSX-DOS instalado. Requerida por el core de [MSX](#) (incompatible con [PC XT](#))
- Tarjeta con una partición primaria FAT16 y DOS instalado. Requerida por el core de [PC XT](#)

(incompatible con [MSX](#))



El tamaño máximo de una partición FAT16 son 4GB



A la hora de poner el nombre a una partición que se vaya a utilizar con esxdos, es importante no utilizar el mismo que el de cualquiera de los directorios dentro, o se producirá un error de acceso a ese directorio. (Ej: No llamar a la partición **BIN**, **SYS** o **TMP**).



Para el core de Spectrum, también es posible tener [una primera partición en formato +3DOS y luego otra\(s\) en formato FAT16 o FAT32](#), para su uso con una ROM de +3e.

Windows

Para configuraciones sencillas, y tarjetas del tamaño adecuado (2GB o menos para FAT16 y 32GB o menos para FAT32), se puede utilizar [la herramienta de formato oficial de la SD Association](#).

Para otras configuraciones, y según la versión de sistema operativo de que se disponga, se podrá utilizar la herramienta de línea de comandos `diskpart` o bien la interfaz gráfica de administración de discos del sistema.

Por ejemplo, en Windows, para formatear una tarjeta con una única partición FAT16 (si la tarjeta es de 4GB o menos de tamaño), que figura como disco 6 al ejecutar `list disk` desde `diskpart`:

```
select disk 6
clean
create part primary
active
format FS=FAT label=ZXDOSPLUS
exit
```

Para crear dos primeras particiones FAT16 de 4GB (por ejemplo, para usar con el core de MSX) y usar el resto del espacio con otra más en formato FAT32 (para tarjetas de más de 8GB):

```
select disk 6
clean
create part primary size=4000
set id=06
active
format fs=FAT label=ZXDOSPLUS quick
create part primary size=4000
format fs=FAT label=EXTRA quick
create part primary
format fs=FAT32 label=DATA quick
exit
```

Para crear una partición FAT32 de 4GB (por ejemplo, para usar con el core de Amstrad CPC 6128) y usar el resto del espacio con otra más en formato FAT32 (para tarjetas de más de 4GB de tamaño):

```
select disk 6
clean
create part primary size=4000
set id=0b
active
format fs=FAT32 label=ZXDOSPLUS unit=4k quick
create part primary
format fs=FAT32 label=EXTRA quick
exit
```

macOS

Para configuraciones sencillas, y tarjetas del tamaño adecuado (2GB o menos para FAT16 y 32GB o menos para FAT32), se puede utilizar [la herramienta de formato oficial de la SD Association](#) o la Utilidad de Discos incluida con el sistema operativo.

Para configuraciones más complejas, será necesario utilizar la línea de comandos.

Por ejemplo, en macOS, para formatear una tarjeta con una única partición FAT16 (si la tarjeta es de 2GB o menos de tamaño), que figura como `disk6` en la lista de dispositivos:

```
diskutil unmountDisk /dev/disk6
diskutil partitionDisk /dev/disk6 MBR "MS-DOS FAT16" ZX DOSPLUS R
```

Para dividirla en dos particiones iguales (si la tarjeta es de 4GB o menos de tamaño):

```
diskutil unmountDisk /dev/disk6
diskutil partitionDisk /dev/disk6 MBR "MS-DOS FAT16" ZX DOSPLUS 50% "MS-DOS FAT16"
EXTRA 50%
```

Para crear dos primeras particiones FAT16 de 4GB (por ejemplo, para usar con el core de MSX) y usar el resto del espacio con otra más en formato FAT32 (para tarjetas de más de 8GB):

```
diskutil unmountDisk /dev/disk6
diskutil partitionDisk /dev/disk6 MBR %DOS_FAT_16% ZX DOSPLUS 4G %DOS_FAT_16% EXTRA 4G
"MS-DOS FAT32" DATA R
sudo newfs_msdos -F 16 -v ZX DOSPLUS -c 128 /dev/rdisk6s1
sudo newfs_msdos -F 16 -v EXTRA -c 128 /dev/rdisk6s2
```



El comando `diskutil` no permite crear particiones FAT16 de más de 2G de tamaño y formatearlas a la vez. Por eso, en el último caso, se crean primero las particiones y luego se formatean en FAT16.

Para crear una partición FAT32 de 4GB (por ejemplo, para usar con el core de Amstrad CPC 6128) y usar el resto del espacio con otra más en formato FAT32 (para tarjetas de más de 4GB de tamaño):

```
diskutil unmountDisk /dev/disk6
diskutil partitionDisk /dev/disk6 MBR "MS-DOS FAT32" ZXDOSPLUS 4G "MS-DOS FAT32" EXTRA
R
```

Para el core de Amstrad CPC, en este caso, al elegir exactamente un tamaño de partición de 4G, macOS elegirá por defecto un tamaño de cluster de 4096. Para un tamaño inferior, podría ser necesario volver a formatear la primera partición con unos comandos similares a estos:



```
diskutil unmountDisk /dev/disk6
newfs_msdos -F 32 -v ZXDOSPLUS -b 4096 /dev/rdisk6s1
```

Por defecto, macOS indexa y guarda información extra en archivos de los discos externos. Esto se puede minimizar usando estos comandos (suponiendo que la partición de la microSD se llama **ZXDOSPLUS**):



```
mdutil -i off /Volumes/ZXDOSPLUS
cd /Volumes/ZXDOSPLUS
rm -rf .{,.}{{fsevents,Spotlight-V*,Trashes}
mkdir .fsevents
touch .fsevents/no_log .metadata_never_index .Trashes
cd -
```

Linux

Existen multitud de herramientas en Linux que permiten formatear y particionar el contenido de una tarjeta microSD (como **fdisk**, **parted**, **cfdisk**, **sfdisk** o **GParted**). Sólo se ha de tener en cuenta que el esquema de particiones a utilizar siempre ha de ser MBR, y la primera partición (la que se utilizará para esxdos) ha de ser primaria.

Por ejemplo, para formatear una tarjeta con una única partición FAT16 (si la tarjeta es de 4GB o menos de tamaño), que figura como **sdc** en la lista de dispositivos:

```
sudo fdisk --compatibility=dos /dev/sdc
```

```
(...)
Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (62-31116288, default 62):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (128-31116288, default 31116288):
Created a new partition 1 of type 'Linux'

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 6
Changed type of partition 'Linux' to 'FAT16'.

Command (m for help): a
Partition number (1, default 1): 1
The bootable flag on partition 1 is enabled now.

Command (m for help): p
Disk /dev/sdc
Disklabel type: dos
Disk identifier

Device      Boot   Start     End   Sectors   Size Id Type
/dev/sdc1            62 31116288 31116288 984,9M 6  FAT16
```

Formatear la partición FAT (requiere permisos de root)

```
sudo mkfs.fat -F 16 -n ZXDOSPLUS -s 128 /dev/sdc1
```

Teclado

Teclado de membrana en gomaDOS+

El teclado de gomaDOS+ al ser similar al teclado del ZX Spectrum original, carece de algunas de teclas existentes en un teclado moderno de PC. Internamente, el teclado de membrana está conectado a una placa Arduino, que se encarga de transformar las pulsaciones al protocolo PS/2 de teclado. Esta placa está programada de manera que pueda comportarse de distintas maneras según nos interese.

Por defecto, está configurado en modo ZX Spectrum. Para cambiar a otro modo, se debe pulsar **Caps Shift+Symbol Shift+U** y luego la tecla correspondiente. Al hacerlo, se tecleará automáticamente un texto indicando el modo seleccionado (por ejemplo: **.zx** si se pulsa **Caps Shift+Symbol Shift+U** y luego **0**).

La siguiente tabla indica los distintos modos y la tecla de activación asociada:

Modo	Tecla
ZX Spectrum	0
Amstrad CPC	1
MSX	2
Commodore 64	3
Atari 800XL	4
BBC Micro	5
Acorn Electron	6
Apple (I y II)	7
Commodore VIC 20	8
PC XT	9
Oric Atmos	A
SAM Coupé	B
Jupiter ACE	C

La distribución del teclado en modo ZX Spectrum, con la pulsación asociada al combinar junto con **Caps Shift+Symbol Shift**, se puede resumir según el siguiente esquema:

1	2	3	4	5	6	7	8	9	0
F1	F2	F3	F4	F5	F6	F7	F8	F9	F1
Q	W	E	R	T	Y	U	I	O	P
F11	F12					Modo			
A	S	D	F	G	H	J	K	L	Enter
				BlqDs					
CShift	Z	X	C	V	B	N	M	SShift	Space
		Guarda		Vers	hRes	sRes			

Donde:

- **BlqDs**: Bloq. Despl. cambia de modo vídeo RGB a VGA y viceversa (en el core de Next, se debe usar **Caps Shift+Symbol Shift+2** o **F2**)
- **Guarda**: Define el modo actual como el modo por defecto
- **Vers**: Muestra (teclea) la versión actual del firmware
- **hRes**: Hard Reset
- **sRes**: Soft Reset

La lista completa de combinaciones de teclado (y el modo en que se pueden utilizar) es la siguiente:

Caps S.+Symbol S.	Modo	Acción
1	Todos	F1
2	Todos	F2
3	Todos	F3
4	Todos	F4
5	Todos	F5
6	Todos	F6
7	Todos	F7
8	Todos	F8
9	Todos	F9
0	Todos	F10
Q	Todos	F11
W	Todos	F12
S	C64	Ctrl+F12
E	Acorn/CPC	Re Pág
R	Acorn	PgAbajo
U	Todos	Modo
G	ZX/MSX/C64	B1qDs
X	Todos	Guarda
C	PC	OPQA
V	Todos	Versión
B	ZX	Ctrl+Alt+Bcksp
N	ZX	Ctrl+Alt+Supr

Teclado PS/2

El mapa de un teclado conectado al puerto PS/2 (asignación de las teclas físicas del teclado con las pulsaciones que se presentan a los distintos cores) se cambia desde el menú **Advanced** de la BIOS. Existen tres mapas distintos a elegir: Español (por defecto), inglés, y Spectrum (avanzado).

También se puede cambiar con la utilidad **keymap**. Dentro de **/bin** hay que crear un directorio llamado **keymaps** y ahí copiar los mapas de teclado se desee usar. Por ejemplo, para cambiar al mapa US hay que escribir **.keymap us** desde esxdos.

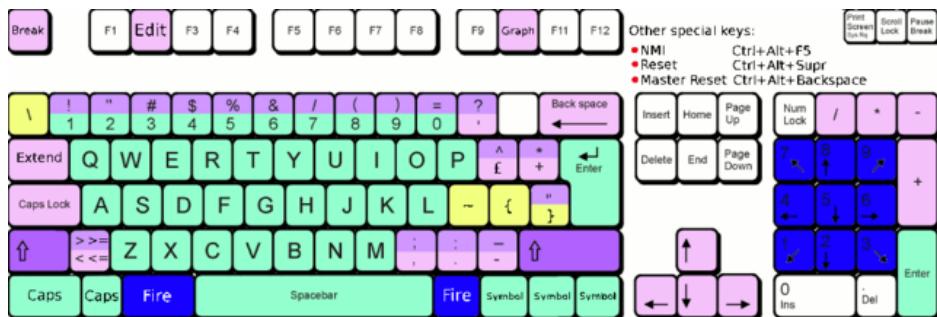
Para que el mapa se conserve después de un master reset, hay que tener seleccionado **Default** en la configuración de BIOS.

Para más información, consultar [este mensaje en el foro de ZX-Uno](#).

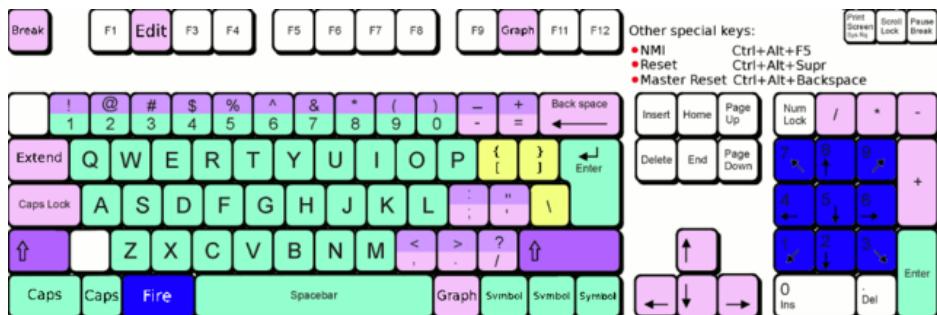


SE Basic IV utiliza su propio sistema nativo para configurar el mapa de teclado y también tiene [mapas propios](#).

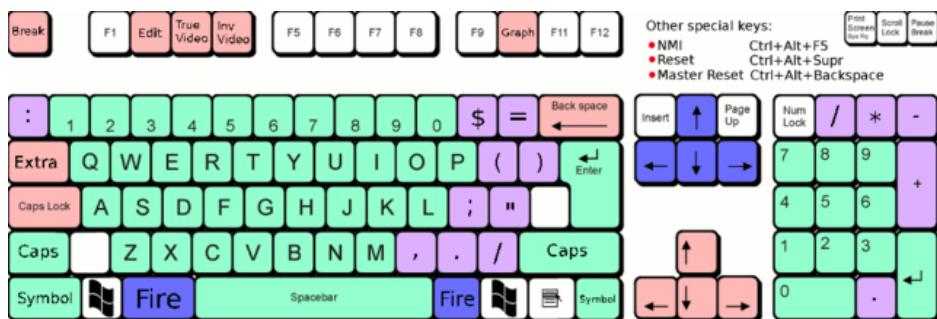
Español



Inglés



Spectrum



Teclas especiales y botones

Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado **ZX**. Véase el [apartado dedicado a los modos de teclado](#) de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado **PC XT** (Por ejemplo, **Caps Shift+Symbol Shift+2** en vez de **Caps Shift+1**).

Teclas especiales durante el arranque:

- **F2** (**Caps Shift+1** en gomaDOS+) Entrar en la BIOS
- **Bloq. Mayús o Cursor abajo** (**Caps Shift+2** en gomaDOS+) o, si hay un joystick conectado, pulsar la dirección hacia abajo: Menú de selección de cores
- **Esc** (**Caps Shift+Espacio** en gomaDOS+) o, si hay un joystick de dos o más botones conectado, pulsar el botón de disparo 2: Menú de selección de ROMS del core de ZX Spectrum
- **R**: Carga la rom del core de ZX Spectrum en modo "real" deshabilitando esxdos, nuevos modos gráficos, etc.
- **/** (del teclado numérico, **Symbol Shift+V** en gomaDOS+): Carga la ROM por defecto del core de ZX Spectrum en modo "root"
- Número del **1** al **9**: Cargar el core en la ubicación de la Flash correspondiente a dicho número

Teclas especiales que se pueden utilizar durante la ejecución del core principal (ZX Spectrum):

- **Esc** (**Caps Shift+Espacio** en gomaDOS+): BREAK
- **F1** (**Caps Shift+Symbol Shift+1** en gomaDOS+): En las versiones más recientes del core, activar los distintos modos de color monocromo
- **F2** (**Caps Shift+1** en gomaDOS+): Edit
- **F5** (**Caps Shift+Symbol Shift+5** en gomaDOS+): NMI
- **F7** (**Caps Shift+Symbol Shift+7** en gomaDOS+): Reproducir o Pausa en la reproducción de archivos .PZX
- **F8** (**Caps Shift+Symbol Shift+8** en gomaDOS+): Rebobinar el archivo .PZX hasta la marca anterior
- **F10** (**Caps Shift+9** en gomaDOS+): Graph
- **F12** (**Caps Shift+Symbol Shift+W** en gomaDOS+): Turbo Boost. Pone a la CPU a 28MHz mientras se mantenga pulsada (a partir del core EXP27).
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+B** en gomaDOS+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**.
- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+N** en gomaDOS+): Soft reset.
- **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+): cambia de modo vídeo RGB a VGA y viceversa.

esxdos

esxdos es un firmware para la interfaz the DivIDE/DivMMC, que el ZX DOS+ implementa, y que permite el acceso a dispositivos de almacenamiento como la tarjeta microSD. Incluye comandos similares a los de UNIX, aunque para usarlos hay que precederlos con un punto, por ejemplo `.ls`, `.cd`, `.mv`, etc.

Para poder utilizarlo es necesario incluir los ficheros correspondientes en la primera partición de la tarjeta microSD.

En el momento de escribir este documento, la versión incluida con ZX DOS+ es la 0.8.6, y se puede descargar desde la página oficial [en este enlace](#).

Una vez descargado y descomprimido, se han de copiar, a la raíz de la tarjeta, los directorios **BIN**, **SYS** y **TMP** con todo su contenido.

Si todo se ha hecho correctamente, al encender el core Spectrum de ZX DOS+ se verá cómo esxdos detecta la tarjeta y carga los componentes necesarios para funcionar.



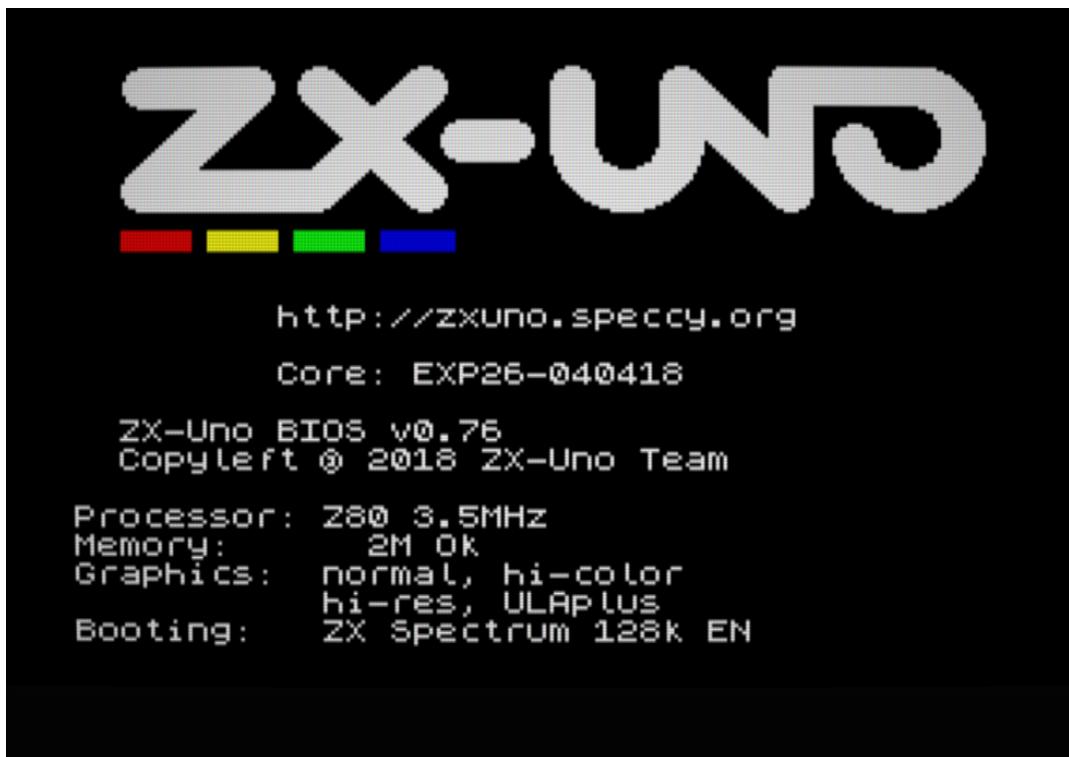
The screenshot shows the ZX DOS+ boot process. It starts with the logo 'ZX DOS+' and the text 'v0.8.6-DivMMC'. Below that, it says '© 2005-2018 Papaya Dezign'. The process continues with 'Detecting Devices...', showing 'sda: card' connected. Then it moves to 'Mounting drives...', showing 'hd0: ZX , FAT16, 128M'. Finally, it shows the loading of several system files: 'Loading ESXDOS.SYS... [OK]', 'Loading RTC.SYS... [OK]', 'Loading NMI.SYS... [OK]', and 'Loading BETADISK:SYS... [OK]'. The background is black with white text.

Es recomendable, además, añadir los comandos esxdos específicos para ZX DOS+. Estos se pueden obtener en la página con el código fuente del proyecto ([aquí](#), [aquí](#) y [aquí](#)), y son los siguientes:

```
back16m  
backzx2  
backzxd  
core  
corebios  
dmaplayw  
esprst  
iwconfig  
joyconf  
keymap  
loadpzx  
playmid  
playrmov  
romsback  
romsupgr  
upgr16m  
upgrzx2  
upgrzxd  
zxuc  
zxunocfg
```

[Más adelante](#) se explica lo que hace cada uno de ellos.

BIOS



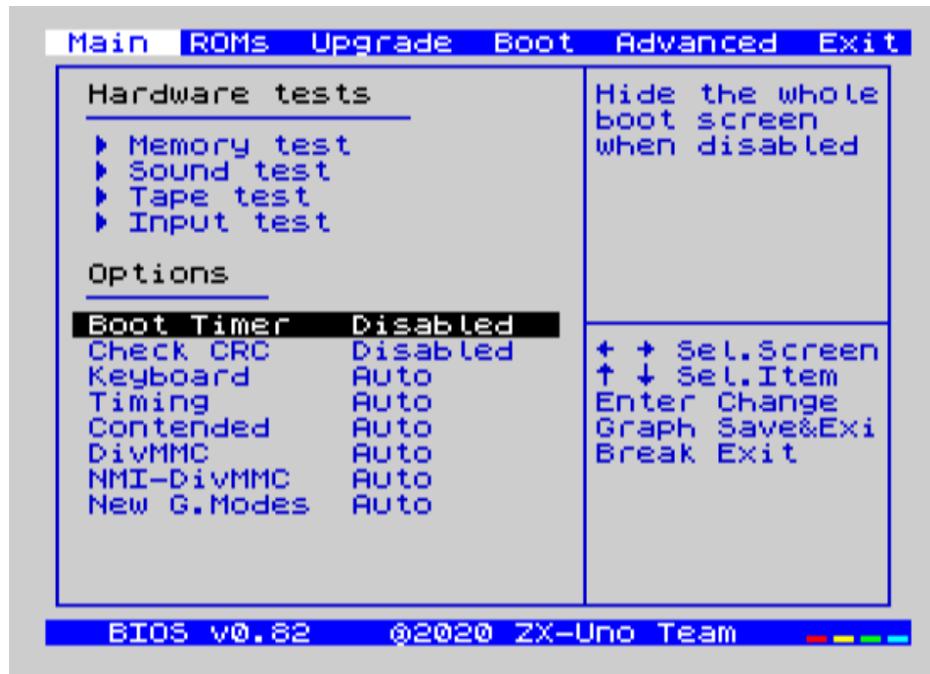
Si se pulsa la tecla **F2** (**Caps Shift+1** en gomaDOS+) durante el arranque, se tendrá acceso a la configuración de BIOS. El firmware de BIOS es el primer programa que se ejecuta cuando se enciende el ZX DOS+. El propósito fundamental del software de BIOS es iniciar y probar el hardware y cargar uno de los cores instalados.

Usando las teclas de cursor izquierda y derecha (**Caps Shift+5** y **Caps Shift+8** en gomaDOS+), se puede navegar por las pantallas de configuración de la BIOS. Con las teclas arriba y abajo (**Caps Shift+7** y **Caps Shift+6** en gomaDOS+) se pueden elegir los distintos elementos de cada pantalla y, con la tecla **Enter**, es posible activar y elegir las opciones de cada una de estas. La tecla **Esc** (**Caps Shift+Espacio** en gomaDOS+) sirve para cerrar las ventanas de opciones abiertas sin aplicar ninguna acción.

Otras teclas que se pueden pulsar durante el arranque de la BIOS son:

- **Bloq. Mayús** o **Cursor abajo**, o **Caps Shift+6** o **Caps Shift+2** o, si hay un joystick conectado, pulsar la dirección hacia abajo: Menú de selección de cores
- **Esc** o **Caps Shift+Espacio**, o si hay un joystick de dos o más botones conectado, pulsar el botón de disparo 2: Menú de selección de ROMS del core de ZX Spectrum
- **R**: Carga la rom del core de ZX Spectrum en modo "real" deshabilitando esxdos, nuevos modos gráficos, etc. (combinado con **Esc** a continuación permite elegir otra ROM distinta)
- **/** (del teclado numérico) o **Symbol Shift+V**: Carga la ROM por defecto del core de ZX Spectrum en modo "root" (combinado con **Esc** a continuación permite elegir otra ROM distinta)
- Número del **1** al **9**: Cargar el core en la ubicación de la Flash correspondiente a dicho número

Main

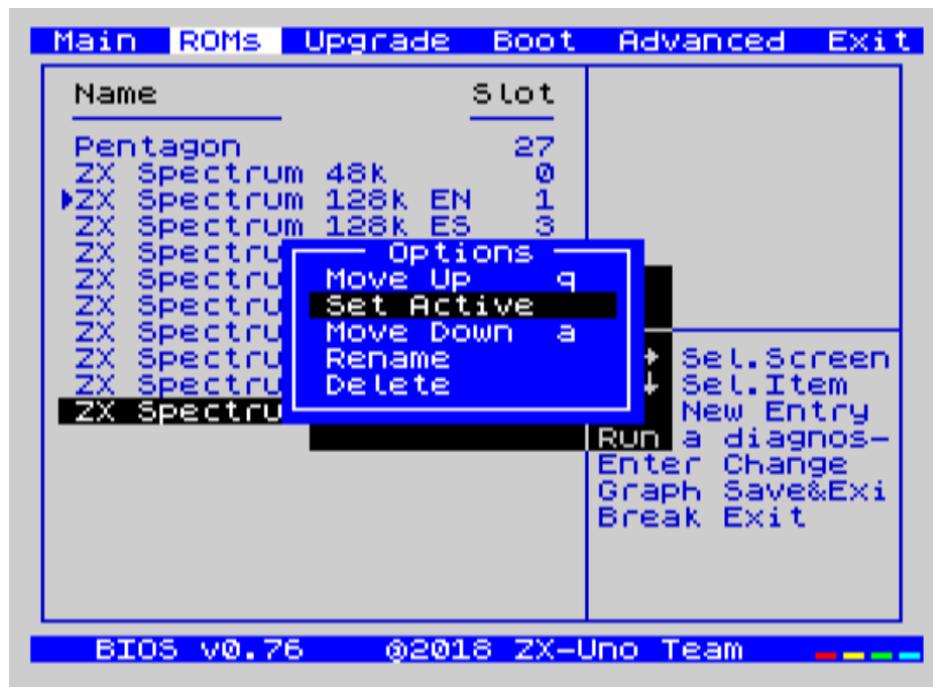


En la primera pantalla de configuración, además de poder ejecutar distintas pruebas, se puede definir el comportamiento por defecto para lo siguiente:

- Espera en el arranque (Boot Timer): Indica el tiempo que está la pantalla de arranque disponible (o la oculta por completo)
- Comprobar CRC de las ROMs (Check CRC): Para comprobar la integridad de las ROMs al cargarlas (más seguro) u omitirla (más rápido)
- Tipo de teclado (Keyboard)
- Timing: Para definir el comportamiento de la ULA (Modo 48K, Modo 128K, Modo Pentagon)
- Contención de memoria (Contended)
- DivMMC
- Soporte NMI para DivMMC
- Soporte para nuevos modos gráficos (ULAPlus, Timex, Radastan)

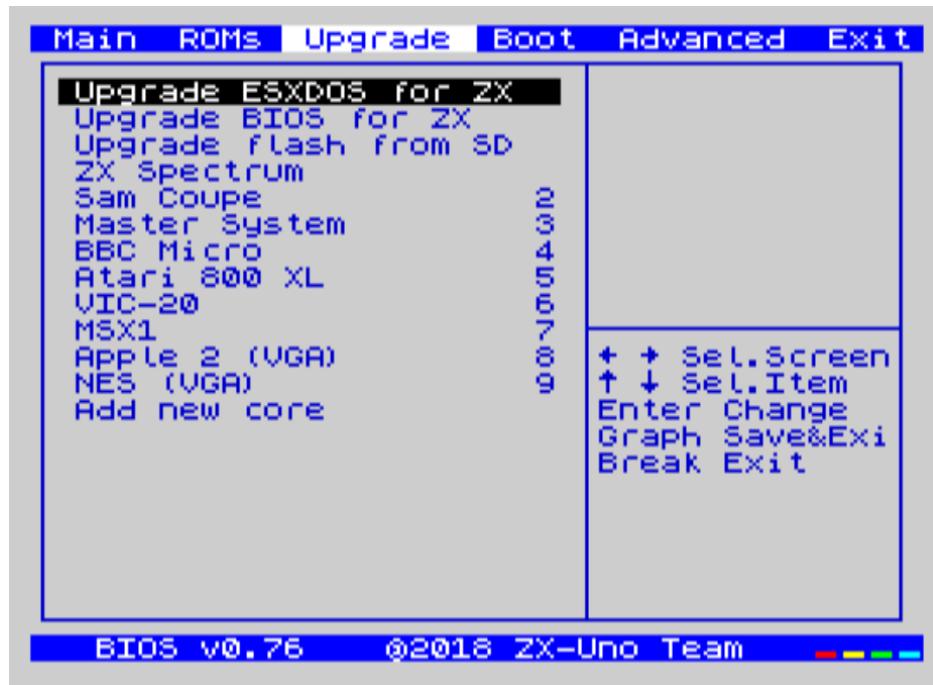
Se puede consultar información más técnica en [la Wiki de ZX-Uno](#).

ROMs



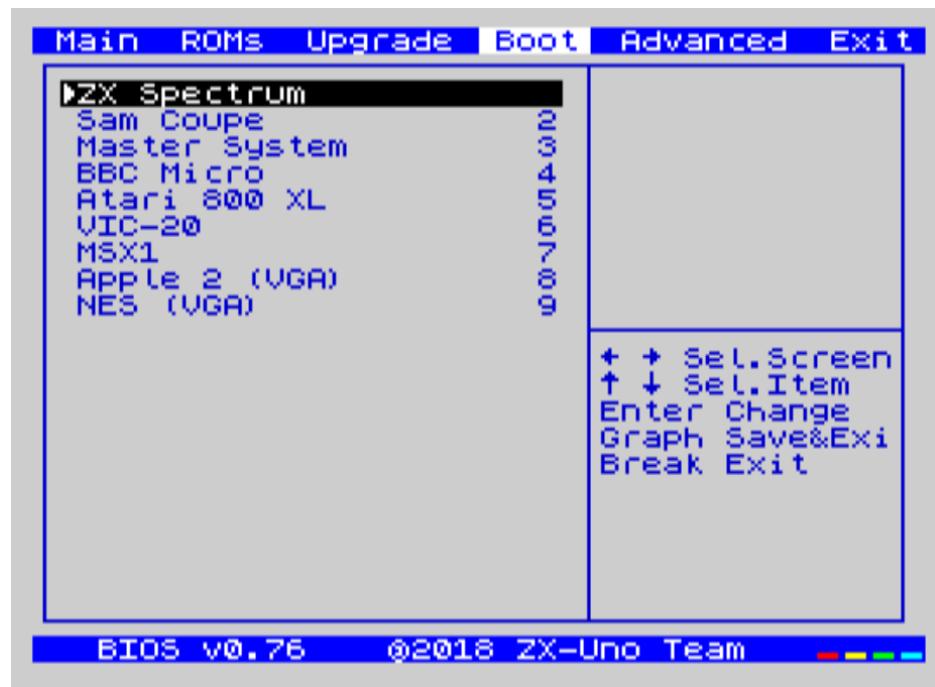
La segunda pantalla muestra las ROMs de ZX Spectrum instaladas y permite reordenar (Move Up, Move Down), renombrar (Rename) o borrar (Delete) cada una de ellas, así como elegir la que se cargará por defecto en el arranque (Set Active).

Upgrade



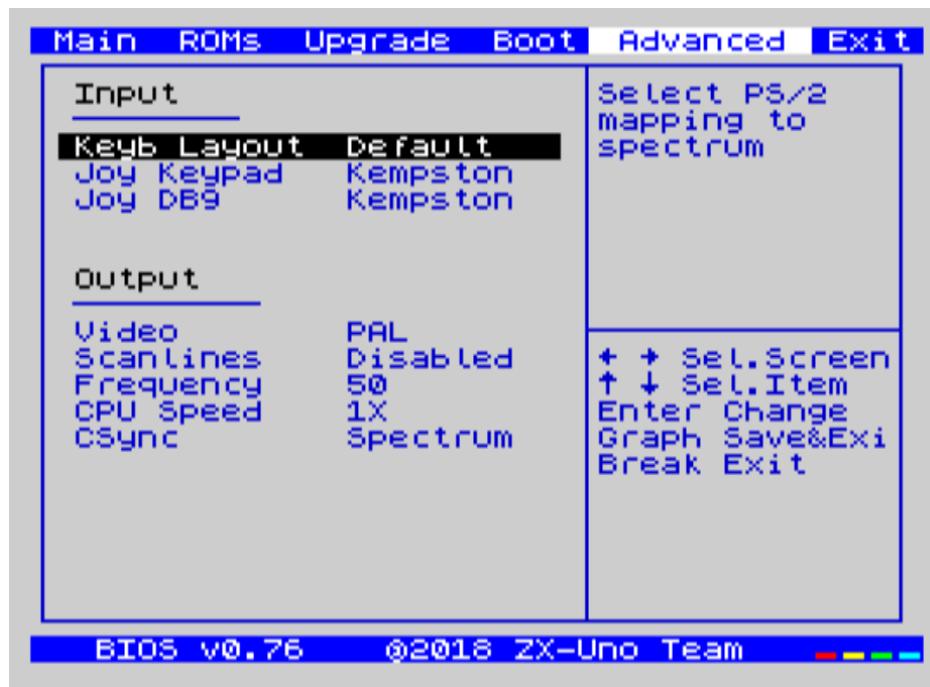
La pantalla *Upgrade* se utiliza para realizar las distintas actualizaciones del contenido de la memoria Flash: esxdos, BIOS, Cores, etc. (véase [el apartado correspondiente a actualizaciones](#) para más información).

Boot



En la pantalla *Boot* se puede elegir qué core de los instalados se desea que cargue por defecto en el arranque.

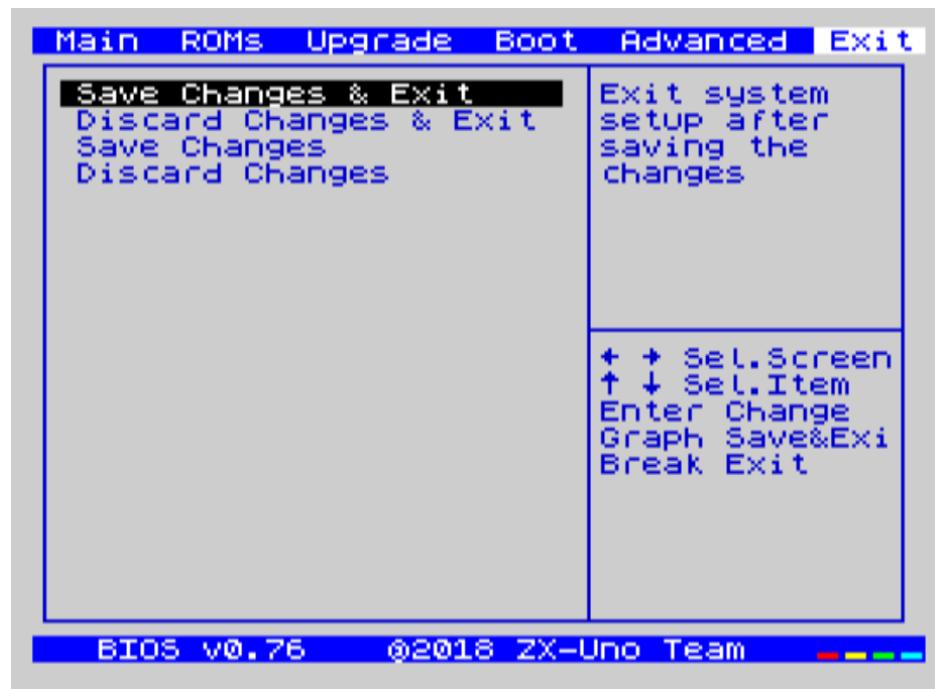
Advanced



La pantalla de configuración avanzada sirve para modificar los siguientes ajustes:

- Distribución del teclado (Keyb Layout): Ver [el apartado correspondiente](#) para más información)
- Comportamiento del joystick emulado con el teclado numérico (Joy Keypad): Kempston, Sinclair Joystick 1, Sinclair Joystick 2, Protek o Fuller
- Comportamiento de un joystick conectado al puerto (Joy DB9): Kempston, Sinclair Joystick 1, Sinclair Joystick 2, Protek, Fuller o simular las teclas **Q, A, O, P, Espacio** y **M**
- Salida de vídeo (Video): PAL, NTSC o VGA
- Simulación de línea de exploración (Scanlines): Activas (Enabled) o inactivas (Disabled)
- Frecuencia horizontal de VGA (Frequency): 50, 51, etc.
- Velocidad de la CPU: Normal (1x) o acelerada (2X, 3X, etc.)
- Csync: Spectrum o PAL

Exit



Finalmente, desde la última pantalla se puede:

- Salir de la configuración de BIOS guardando los cambios (Save Changes & Exit)
- Descartar los cambios y salir (Discard Changes & Exit)
- Guardar los cambios sin salir (Save Changes)
- Descartar los cambios (Discard Changes)

ZX Spectrum

El core principal es el que implementa un ordenador ZX Spectrum. Este core es especial, y no se puede sustituir por otro que no sea de ZX Spectrum, ya que el ZX DOS+ lo utiliza para su funcionamiento.

Estas son algunas de sus principales características:

- Implementación ZX Spectrum 48K, 128K, Pentagon y Chloe 280SE
- ULA con modos ULAPLUS, Timex y modo Radastan (incluyendo scroll por hardware y grupo de paleta seleccionable)
- Posibilidad de desactivar la contención de memoria (para compatibilidad con Pentagon 128)
- Posibilidad de elegir el comportamiento del teclado (issue 2 o issue 3)
- Posibilidad de elegir el timing de la ULA (48K, 128K o Pentagon)
- Control del encuadre de pantalla configurable para tipo de timing, y posibilidad de elegir entre sincronismos originales de Spectrum o sincronismos estándar PAL progresivo.
- Soporte de la MMU horizontal del Timex con bancos HOME, DOC y EXT en RAM.
- Interrupción ráster programable en número de línea, para cualquier linea de TV.
- Posibilidad de activar/desactivar los registros de manejo de bancos de memoria, para mejor compatibilidad con cada modelo implementado
- Posibilidad de activar/desactivar los dispositivos incorporados al core para mejorar la compatibilidad con ciertos programas
- Soporte ZXMMC y DIVMMC para para +3e, esxdos y firmwares compatibles
- Soporte Turbo Sound
- Soporte de SpecDrum
- Cada canal A,B,C de los dos chips AY-3-8912, beeper y SpecDrum pueden dirigirse a las salidas izquierda, derecha, ambas o ninguna, permitiendo la implementación de configuraciones tales como ACB, ABC, etc.
- Soporte de joystick real y joystick en teclado con protocolo Kempston, Sinclair 1 y 2, Cursor, Fuller y QAOOPSCM.
- Soporte de modo turbo a 7MHz, 14MHz, 28MHz
- Soporte de teclado con protocolo PS/2 y mapeado configurable desde el propio Spectrum.
- Soporte de ratón PS/2 emulando el protocolo Kempston Mouse.
- Posibilidad de salida de vídeo en modo RGB 15kHz, o VGA
- Frecuencia de refresco vertical seleccionable para la compatibilidad con monitores VGA
- Soporte de arranque multicore: desde el Spectrum se puede seleccionar una dirección de la SPI Flash y la FPGA cargará un core desde ahí
- Modos de color incluyendo monocromo o fósforo verde/naranja
- Salida de audio I²S (con el [addon RTC+I²S+Pizero](#))

- Soporte Wi-Fi (UART)
- Soporte MIDI (General MIDI) vía addon
- Carga de ficheros de ficheros **PZX** desde la SD

ROMs

El core de ZX Spectrum tiene la capacidad de inicializar utilizando diferentes versiones de ROM (48K, 128K, Plus 2, etc.). Estas se almacenan en la memoria flash del ZX DOS+, y se puede elegir cuál cargar, pulsando la tecla **Esc** durante el arranque. También es posible definir desde la configuración de BIOS, cuál es la ROM que se desea que se cargue por defecto.

Véase el [apartado de actualizaciones](#) para más información sobre cómo ampliar o modificar las ROMs almacenadas en la memoria flash.

DerbyPro

[DerbyPro](#) o [Derby++](#) es una ROM mejorada para el ZX Spectrum, basada en la versión 1.4 de la ROM de desarrollo Derby. El Spectrum 128 (nombre en código "Derby") fue una máquina española, encargada por Investronica y lanzada en 1985. Incluía un teclado aparte que añadía varias teclas de edición extra. En 1986 se lanzó la versión para Reino Unido con una versión simplificada de 128 BASIC y sin teclado extra. Derby++ se basa en la ROM española para incluir lo mejor de las dos versiones, sin sus inconvenientes, y con soporte para nuevo hardware.



Algunas de sus características incluyen:

- Modo 48K con compatibilidad binaria al 100%
- Comando PLAY con soporte para 6 canales
- Acceso al navegador NMI de esxDOS desde el menú de arranque
- 128 BASIC con correcciones, comandos adicionales y editor de texto a pantalla completa
- Soporte para esxDOS en 128 BASIC
- Acceso vía menú a TR-DOS
- Comando PALETTE para ULAplus
- Ejecución de la mayor parte del software para Spectrum sin tener que hacer un cambio de configuración en la BIOS

Se puede descargar la ROM, un manual de usuario y otros ficheros del [grupo público oficial de Facebook](#).

Como es una ROM de 64K con soporte para nuevo hardware, se pueden usar estas opciones al [añadirla a la SPI flash](#):

Ajuste	Significado
d	Habilitar DivMMC
n	Habilitar NMI DivMMC (menú de esxdos)
t	Usar timings de 128K

CargandoLeches

[CargandoLeches](#) es un conjunto de ROMs de ZX Spectrum originalmente pensadas para cargar juegos a una velocidad de 15 a 20 veces superior a lo normal. En lugar de una cinta se requiere una fuente de audio digital como un ordenador, un dispositivo móvil, un reproductor MP3, etc. También tiene una rutina que detecta el método de carga y si no se trata de una carga ultrarrápida, ejecuta el código de la ROM original. En cargas que no sean CargandoLeches no se nota por tanto la diferencia entre usar esta ROM y la ROM original.

Desde la versión 2.0, el proyecto pasó de ser una única ROM a varias, cada una con distintas opciones. Así, es posible elegir diferentes combinaciones que pueden incluir:

- Carga ultrarrápida
- Reset & Play (es decir, que al hacer un reset automáticamente se ponga en modo carga de cinta)
- Introducción de POKEs
- Desactivar o activar la expansión de tokens (palabras clave) de Sinclair BASIC

El conjunto completo de ROMs está disponible para descargar desde el repositorio en GitHub [aquí](#).

Dependiendo de la ROM elegida, los ajustes a indicar cuando se [añada a la SPI flash](#) pueden variar. Por ejemplo, para la ROM [48le_ea_re_po](#) (que tiene habilitadas todas las opciones), se pueden usar estos ajustes (no hay que habilitar NMI DivMMC porque el editor de POKEs ya lo utiliza):

Ajuste	Significado
d	Habilitar DivMMC
h	Deshabilitar bit alto de ROM (bitd 2 de 1FFD)
l	Deshabilitar bit bajo de ROM (bit 4 de 7FFD)
x	Deshabilitar modo Timex

POKEs

En el caso de usar una ROM con la opción de introducir POKEs, se hace de la siguiente manera:

1. Una vez el juego ha cargado, pulsando NMI ([F5](#)), aparecerá un campo en la parte superior izquierda de la pantalla
2. Escribir la dirección del POKE a introducir y pulsar [Enter](#)
3. Escribir el valor del POKE y pulsar [Enter](#)
4. Repetir los pasos 2. y 3. todas las veces que se desee. Para terminar y volver al juego, pulsar [Enter](#) dos veces seguidas

Preparación de cintas de carga ultrarrápida

Las ROMs con la opción de carga ultrarrápida necesitan archivos de cinta especiales que se generan desde ficheros **TAP** de carga normal, de juegos que no tengan protección de carga o modo turbo.

Para crear una cinta de carga ultrarrápida se necesitan las utilidades de línea de comandos **leches** y **CgLeches**. Estas se pueden conseguir, para Windows, [en el repositorio oficial](#). Para macOS es posible descargar una versión no oficial [en este otro repositorio](#).

En otro caso, es posible compilar desde [el código fuente disponible en el repositorio oficial](#). Por ejemplo, para compilar en Linux usando **gcc** basta con usar estos comandos:

```
gcc leches.c -o leches  
gcc CgLeches.c -o CgLeches
```

Para generar una cinta de carga ultrarrápida se ha de invocar desde una consola al comando **CgLeches** indicando, al menos, el fichero **TAP** de origen, y el fichero (**WAV** o **TZX**) de destino. Existen otros parámetros como el nivel de velocidad de la carga, entre 0 y 7 (donde 0 es la más rápida pero posiblemente más incompatible), si se desea un fichero mono, estéreo, etc. (en el caso de **WAV**) y más.

Así, para producir un fichero **WAV** de audio con una cinta de carga ultrarrápida desde el fichero de cinta **Valley.tap** con velocidad de carga 5, se haría así:

```
(...) CgLeches Valley.tap Valley.wav 5
```

Ahora el fichero **Valley.wav** se puede reproducir desde un ordenador u otro dispositivo y cargarlo usando la ROM (véase la sección dedicada a la [carga desde cinta](#) para más detalles).



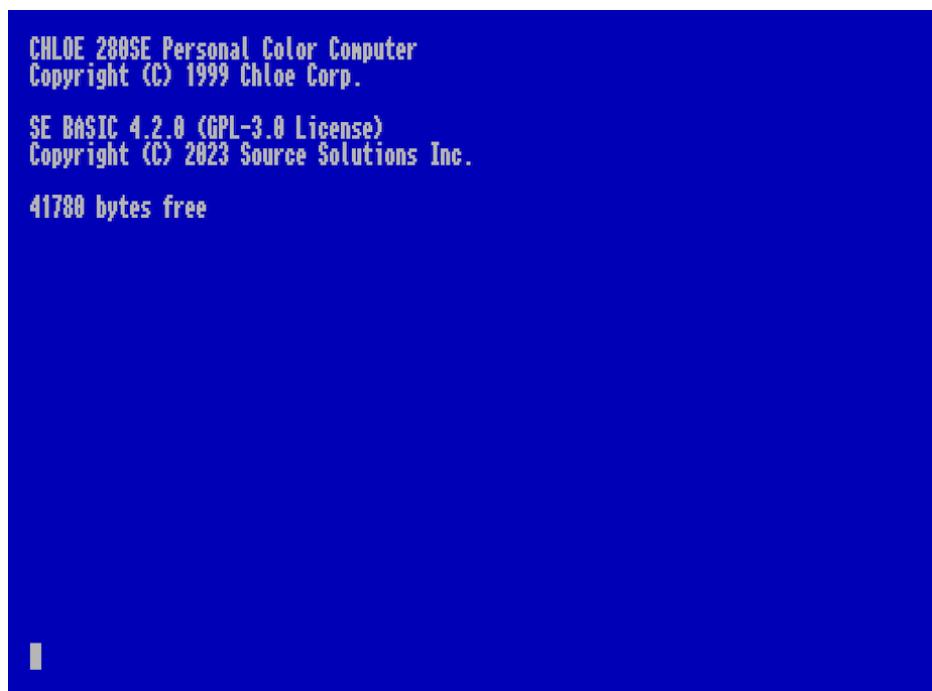
Debido a limitaciones en el hardware, los ficheros **TZX** generados con **CgLeches** no funcionan correctamente con **Miniduino**, aunque sí que suelen funcionar con **PlayTZX**.

SE Basic IV

SE Basic IV es un intérprete de Microsoft BASIC, gratuito y de código abierto para la arquitectura Z80. Aunque aspira a tener un alto grado de compatibilidad con Microsoft BASIC, hay algunas diferencias. Se ha diseñado para funcionar en un [Chloe 280SE](#) pero también es compatible con el core de ZX Spectrum del ZX DOS+.

SE Basic comenzó en 1999 como firmware para el [ZX Spectrum SE](#), el antecesor al Chloe 280SE. Las primeras versiones eran modificaciones aplicadas sobre la ROM original del ZX Spectrum. Desde la versión 1 utiliza su propio fichero ensamblador. Desde la versión 2, soporta ULAPlus.

La versión 3, también conocida como [OpenSE BASIC](#) reemplazó el código ROM original con una versión de código abierto derivada de las ROMs del [ZX81](#) y el [SAM Coupé](#). Todavía se mantiene como un firmware de código abierto para el Spectrum y se incluye, por ejemplo, en el [repositorio principal de Debian](#) para su uso en emuladores.



La versión 4.0 añadió soporte para un modo de 80 columnas. La versión 4.1 fue un intento fallido de refactorizar el código. A partir de 2019, la última versión (4.2 Cordelia) se reconstruyó desde cero para aprovechar al completo el core de ZX Spectrum del ZX-Uno (y ZX DOS+). Aunque las versiones anteriores mantenían un gran nivel de compatibilidad con Sinclair BASIC y otro software, esta nueva versión ya no tiene soporte para software de Sinclair y es mucho más parecida en sintaxis a Atari BASIC.

La versión 4.2 requiere que divMMC esté activo con esxDOS o UnoDOS 3 instalado. Sin embargo, ni los [comandos "DOT"](#) ni el navegador NMI están soportados.

Algunas de sus características son:

- Modos de vídeo con paleta de 40 columnas (16 colores) y 80 columnas (2 colores)
- Evaluación de expresiones siempre activa (uso de variables como nombres de fichero)
- Formato de empaquetado de aplicaciones con soporte para convertir programas BASIC en apps

- Tipos automáticos de datos
- Lógica a nivel de bit (AND, NOT, OR, XOR).
- Sistema de ayuda incorporado
- Troceado de cadenas a elegir entre Microsoft (LEFT\$, MID\$, RIGHT\$) o Sinclair (TO)
- Caracteres compuestos (soporta Vietnamita).
- Sistema de archivos basado en discos (nada de cintas).
- Gestión de errores (ON ERROR..., TRACE).
- Control de flujo (IF...THEN...ELSE, WHILE...WEND).
- Acceso aleatorio completo a ficheros desde BASIC (OPEN, CLOSE, SEEK).
- Soporte para teclado de tamaño completo (Supr, Inicio, Fin y demás).
- Modos gráficos en el modo de 40 columnas (CIRCLE, DRAW, PLOT).
- Traducción de conjuntos de caracteres, mensajes de error y mapas de teclado
- Nombres de variable largos
- Notación de números estilo Motorola (%; binario, @; octal, \$; hexadecimal)
- BREAK NMI
- Validación de sintaxis según se introduce
- Comando PLAY con soporte para PSG de 6 canales y MIDI
- Funciones recursivas definidas por el usuario
- Actualizaciones inteligentes de firmware
- Abreviación de tokens y atajos (&; AND, ~; NOT; |; OR, ?; PRINT, ';' REM')
- Capacidad para deshacer NEW (OLD).
- Canales definidos por el usuario
- Conjuntos de caracteres definidos por el usuario (256 caracteres)
- Macros definidas por el usuario
- Modos de pantalla definidos por el usuario



Para que la opción de actualización de firmware funcione, se ha de instalar SE Basic IV en los slot 2 y 3



El uso de la actualización de firmware reemplaza la versión de esxDOS con la última versión de UnoDOS 3

Otras ROMs

Estos son algunos ajustes válidos para [añadir a la SPI flash](#) algunas otras ROM personalizadas:

Nombre de la ROM	Ajustes
Gosh Wonderful ROM v1.33	dnhl17x
Looking Glass 1.07	dnhl17x
ZX82 by Daniel A. Nagy	dnhl17
ZX85 by Daniel A. Nagy	dntmh1
Arcade Game Designer 0.1	thl17x

Formato avanzado de la tarjeta microSD (+3e)

Una de las ROM que se pueden cargar con el core de ZX Spectrum es la de ZX Spectrum +3e, que es una versión mejorada del Sinclair ZX Spectrum +3, y que soporta el uso de discos duros o tarjetas de memoria.

El +3e usa su propio esquema de particionado (llamado IDEDOS) para dividir el disco duro en diferentes particiones donde se pueden almacenar datos. Se necesita una versión 1.28 o superior de la ROM para poder compartir particiones IDEDOS con particiones MBR. En otro caso, se ha dedicar la tarjeta completa al particionado IDEDOS.



El esquema de particionado que se presentará a continuación sólo se podrá utilizar con el core de Spectrum. Otros cores que necesiten acceso a la tarjeta SD posiblemente fallen o no se inicien correctamente, si se encuentra insertada una SD con este formato.



En IDEDOS, cada partición puede tener un tamaño entre 1 y 16 Megabytes (16 millones de bytes), y cada disco puede tener entre 1 y 65535 particiones. Por tanto, lo máximo que se puede ocupar de una tarjeta será alrededor de 1 TB de espacio.

A continuación se explica una forma de dividir una tarjeta en dos o tres partes, con la primera partición IDEDOS (1GB de tamaño), la segunda FAT16 (4GB) y la tercera FAT32 (resto del espacio de la tarjeta).

En la segunda partición se puede instalar, tal y como se explicó anteriormente [exsdos](#) y otros programas.

Windows

Se puede utilizar el administrador de discos de Windows. Los pasos a seguir serían:

1. Eliminar todas las particiones de la tarjeta
2. Crear una partición extendida, del tamaño que se quiera utilizar para IDEDOS
3. Crear una partición primaria de 4GB y formatear como FAT16
4. Opcionalmente, crear otra partición primaria ocupando el resto del espacio y formatear como FAT32

macOS

Será necesario utilizar la línea de comandos. Lo primero es determinar el disco a formatear:

```
diskutil list
```

En este ejemplo sería el disco 6:

```
(...)
/dev/disk6 (external, physical):
 #:          TYPE NAME      SIZE    IDENTIFIER
 0: FDisk_partition_scheme          *15.9 GB   disk6
 1: DOS_FAT_32 UNKNOWN           15.9 GB   disk6s1
```

Pasos a seguir:

1. Expulsar el disco y editar el esquema de particiones (el segundo paso requiere permisos de administrador):

```
diskutil unmountDisk /dev/disk6
sudo fdisk -e /dev/rdisk6
```

```
fdisk: could not open MBR file /usr/standalone/i386/boot0: No such file or directory
Enter 'help' for information
```

```
fdisk: 1> erase
fdisk:*1> edit 1
Partition id ('0' to disable) [0 - FF]: [0] (? for help) 7F
Do you wish to edit in CHS mode? [n]
Partition offset [0 - 31116288]: [63] 128
Partition size [1 - 31116287]: [31116287] 2017152
```

```
fdisk:*1> edit 2
Partition id ('0' to disable) [0 - FF]: [0] (? for help) 06
Do you wish to edit in CHS mode? [n]
Partition offset [0 - 31116288]: [2017280]
Partition size [1 - 29099135]: [29099135] 7812504
```

```
fdisk:*1> flag 2
```

```
fdisk:*1> edit 3
Partition id ('0' to disable) [0 - FF]: [0] (? for help) 0B
Do you wish to edit in CHS mode? [n]
Partition offset [0 - 31116288]: [9829784]
Partition size [1 - 21286504]: [21286504]
```

```

fdisk:*1> print
      Starting      Ending
#: id cyl hd sec - cyl hd sec [      start -      size]
-----
1: 7F 1023 254 63 - 1023 254 63 [        128 - 2017152] <Unknown ID>
2: 06 1023 254 63 - 1023 254 63 [ 2017280 - 7812504] DOS > 32MB
3: 0B 1023 254 63 - 1023 254 63 [ 9829784 - 21286504] Win95 FAT-32
4: 00    0   0   0 -    0   0   0 [          0 -           0] unused

fdisk:*1> write
fdisk: 1> quit

```

2. Formatear las particiones FAT (requiere permisos de administrador)

```

diskutil unmountDisk /dev/disk6
sudo newfs_msdos -F 16 -v ZX DOSPLUS -c 128 /dev/rdisk6s2
sudo newfs_msdos -F 32 -v EXTRA -b 4096 -c 128 /dev/rdisk6s3

```

3. Comprobar cómo el esquema de particiones ha cambiado y ya es el que se deseaba:

```
diskutil list
```

```

(...)

/dev/disk6 (external, physical):
 #:          TYPE NAME      SIZE      IDENTIFIER
 0: FDisk_partition_scheme          *15.9 GB   disk6
 1:          0x7F                1.0 GB   disk6s1
 2: DOS_FAT_16 ZX DOSPLUS        4.0 GB   disk6s2
 3: DOS_FAT_32 EXTRA            10.9 GB  disk6s3

```

Linux

Será necesario utilizar la línea de comandos. Lo primero es determinar el disco a formatear:

```
lsblk
```

En este ejemplo sería **sdc**:

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
(..)
sdc        179:0   0 15,8G  0 disk
└─sdc1     179:1   0 15,8G  0 part
```

Pasos a seguir:

1. Comprobar que no está montado y editar el esquema de particiones (este paso requiere permisos de root):

```
sudo fdisk --compatibility=dos /dev/sdc
```

Welcome to fdisk

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n

Partition type

 p primary (0 primary, 0 extended, 4 free)
 e extended (container for logical partitions)

Select (default p): p

Partition number (1-4, default 1): 1

First sector (62-31116288, default 62): 128

Last sector, +/-sectors or +/-size{K,M,G,T,P} (128-31116288, default 31116288):
2017152

Created a new partition 1 of type 'Linux'

Command (m for help): t

Selected partition 1

Hex code (type L to list all codes): 7f

Changed type of partition 'Linux' to 'unknown'.

Command (m for help): n

Partition type

 p primary (1 primary, 0 extended, 3 free)
 e extended (container for logical partitions)

Select (default p): p

Partition number (2-4, default 2):

```
First sector (45-31116288, default 45): 2017280 .
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2017153-31116288, default 31116288):
7812504
```

```
Created a new partition 2 of type 'Linux'
```

```
Command (m for help): t
Partition number (1,2, default 2): 2
Hex code (type L to list all codes): 6
```

```
Changed type of partition 'Linux' to 'FAT16'.
```

```
Command (m for help): a
Partition number (1,2, default 2): 2
```

```
The bootable flag on partition 2 is enabled now.
```

```
Command (m for help): n
Partition type
  p  primary (2 primary, 0 extended, 2 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (3-4, default 3): 3
First sector (45-31116288, default 45): 9829784 .
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2017153-31116288, default 31116288):
31116288
```

```
Created a new partition 3 of type 'Linux'
```

```
Command (m for help): t
Partition number (1-4, default 3): 3
Hex code (type L to list all codes): b
```

```
Changed type of partition 'Linux' to 'W95 FAT32'.
```

```
Command (m for help): p
Disk /dev/sdc
Disklabel type: dos
Disk identifier

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdc1          128 2017152  2017025 984,9M 7f unknown
/dev/sdc2    *    2017280 7626751 7812504  2,7G  b FAT16
/dev/sdc3          9829784 7626751 21286504   21G  b W95 FAT32
```

2. Formatear las particiones FAT (requiere permisos de root)

```
sudo mkfs.fat -F 16 -n ZXDOSPLUS -s 128 /dev/sdc2
sudo mkfs.fat -F 32 -n EXTRA -s 128 /dev/sdc3
```

3. Verificar que el esquema de particiones ha cambiado y ya es el que se quería:

```
lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
(...)						
sda	179:0	0	15,8G	0	disk	
└─sda1	179:1	0	1G	0	part	
└─sda2	179:2	0	4G	0	part	
└─sda3	179:3	0	10,8G	0	part	

+3e

Una vez preparada la tarjeta para su uso, se puede arrancar el core de Spectrum con una ROM de +3e, y formatear la parte de IDEOS según se deseé.

El primer paso consiste en determinar la geometría de la microSD. Con la tarjeta insertada en el ZX DOS+, desde el core de Spectrum con la ROM de +3e, ejecutar el comando:

```
CAT TAB
```

Esto devolverá un resultado indicando el número de [cilindros, cabezales y sectores](#).

Teniendo esto en cuenta, calculamos el espacio que ocupa nuestra partición, en cilindros. Por ejemplo, si el número de cilindros obtenido es de 32768, y queremos utilizar 1GB de una tarjeta de 16GB, el número de cilindros que se necesitarían son $32768/16=2048$. Por tanto, podemos formatear la partición IDEOS usando ese número:

```
FORMAT TO 0,100,2048
```

El primer valor (**0**) indica el disco a utilizar (el primero), el segundo valor es el número máximo de particiones IDEOS que se podrán usar, y el tercer valor es el número de cilindros a utilizar.

Una vez hecho el formato, ya será posible crear nuevas particiones. Por ejemplo, para crear una partición llamada "Software" de 16MB, una llamada "Swap1", de 4MB (para usar como espacio swap) y otra llamada "Utils" de 8MB:

```
NEW DATA "Software",16  
NEW EXP "Swap1",4  
NEW DATA "Utils",8
```

Para más información sobre el uso de los distintos comandos de +3e para acceso al disco, se puede visitar [esta página en World of Spectrum](#).

Comandos de esxdos

Guía básica

Existen dos tipos diferentes de comandos de esxdos, los llamados comandos "DOT", que, como su nombre indica, comienzan por un punto, y las extensiones de la funcionalidad de comandos existentes en BASIC.

Los principales comandos "DOT" commands son los siguientes:

- **128**: Para pasar al modo 128K desde el modo 48K.
- **cd**: Cambiar el directorio actual de trabajo.
- **chmod**: Cambiar los atributos de los ficheros de la tarjeta microSD.
- **cp**: Copiar un archivo.
- **divideo**: Reproduce un archivo de vídeo Divideo (.DVO).
- **drives**: Mostrar las unidades disponibles.
- **dskprobe**: Utilidad para ver el contenido a bajo nivel de un dispositivo de almacenamiento.
- **dumpmem**: Permite volcar contenido de la memoria RAM a un fichero.
- **file**: Intenta determinar el tipo de un fichero por su contenido (como el comando de UNIX).
- **gramon**: Monitor para buscar gráficos, sprites, fuentes de texto, etc. en la memoria RAM.
- **hexdump**: Muestra el contenido de un fichero usando notación hexadecimal.
- **hexview**: Permite ver y navegar por el contenido de un fichero usando notación hexadecimal.
- **launcher**: Crea un atajo (launcher) para abrir directamente un fichero TAP.
- **ls**: Ver el contenido de un directorio.
- **lstap**: Ver el contenido de un fichero .TAP
- **mkdir**: Crear un directorio.
- **mktrd**: Crear un fichero imagen de disquete .TRD
- **more**: Ver el contenido de un archivo de texto.
- **mv**: Mover un archivo.
- **partinfo**: Muestra información sobre las particiones de un dispositivo de almacenamiento.
- **playpt3**: Reproducir un archivo musical .PT3.
- **playsqt**: Reproducir un archivo musical .SQT.
- **playstc**: Reproducir un archivo musical .STC.
- **playtfm**: Reproducir un archivo musical .TFC.
- **playwav**: Reproducir un archivo de audio .WAV.
- **rm**: Borrar un archivo o directorio.
- **snapshot**: Carga ficheros snapshot.

- **speakcz**: Reproduce texto usando pronunciación checa.
- **tapein**: Montar un archivo .TAP para poder ser utilizado luego desde BASIC con la sentencia LOAD
- **tapeout**: Montar un archivo .TAP para poder ser utilizado luego desde BASIC con la sentencia SAVE
- **vdisk**: Monta una unidad de disquete .TRD para usar en el entorno TR-DOS (Una vez montadas todas las unidades deseadas, se puede entrar en el emulador de TR-DOS escribiendo: **RANDOMIZE USR 15616**)

Algunos comandos extendidos de BASIC son:

- **GO TO** para cambiar de unidad y/o directorio (ej: **GO TO hd1** o **GO TO hd0"juegos"**)
- **CAT** para mostrar el contenido de una unidad
- **LOAD** para cargar un fichero desde una unidad (programa en BASIC, pantalla, código, etc. por ejemplo **LOAD *"Pantalla.scr" SCREEN\$**)
- **SAVE** para guardar datos en un fichero (Ej: **SAVE *"Programa.bas"**)
- **ERASE** para borrar un fichero

Además, esxdos incluye un gestor NMI, es decir, una aplicación que se carga cuando se pulsa NMI (F5) y que facilita la navegación por la tarjeta microSD y la carga de algunos tipos de archivo (TAP, Z80, TRD, etc.). Pulsando la tecla "H" se accede a una pantalla de ayuda, en la que se indican todas las teclas disponibles.



El gestor de esxdos muestra las entradas de archivos y directorios en el orden de la tabla FAT interna, y no de manera alfabética. Si se desea ver esta información ordenada, se debe reorganizar la estructura de la tarjeta con una utilidad como **FAT Sorter** para Windows, **FATsort** para Linux y macOS, **YAFS**, **SDSorter** u otros.



En el caso de utilizarse también la tarjeta con el core de **PC XT**, **no se debe utilizar ninguna utilidad de reordenación de FAT** ya que esto puede provocar que deje de arrancar correctamente DOS en dicho core.



Existen navegadores alternativos como **Long Filename Browser de Bob Fossil** o **New NMI Handler de Dr. Slump** que incorporan funcionalidades y características que no tiene el gestor de esxdos original

Comandos para ZX DOS+

Tal y como se ha explicado en la parte de instalación, existe una serie de comandos que son exclusivos para ZX DOS+, y que se describen a continuación:

- **back16m**: Copia a un fichero **FLASH.ZX1** en el directorio raíz de la tarjeta microSD el contenido de una memoria SPI Flash de 16 megas. Se debe ejecutar desde una ROM en modo "root". Tras terminar su ejecución hay que ejecutar el comando **.ls** para que se termine de grabar la cache en la tarjeta.
- **backzx2** o **backzxd**: Genera un fichero **FLASH_32.ZX2** o **FLASH_32.ZXD** en el directorio raíz de la tarjeta microSD el contenido de una memoria SPI Flash de 32 megas. Se debe ejecutar desde una ROM en modo "root". Cuando termine hay que ejecutar el comando **.ls** para que se termine de grabar la cache en la tarjeta microSD. Si no se hace, la longitud del archivo se quedará en 0 de forma errónea.
- **corebios**: Para hacer una actualización conjunta del core de ZX Spectrum y de la BIOS.
- **dmaplayw**: Reproduce un archivo de audio .WAV, que debe ser de 8 bits, sin signo y muestrado a 15625 Hz.
- **esprst**: Resetea el módulo WiFi ESP8266(ESP-12).
- **iwconfig**: Configura el módulo WiFi.
- **joyconf**: Configura y prueba los joysticks de teclado y DB9.
- **keymap**: Sirve para cargar una definición de teclado diferente.
- **loadpzx**: Para cargar un archivo de imagen de cinta .PZX.
- **loadtap**: Para cargar un archivo de imagen de cinta usando la integración con .PZX.
- **playmid**: Reproduce archivos musicales .MID en el addon MIDI.
- **playrmov**: Reproduce vídeos en [formato radastaniano \(ficheros .RDM\)](#). Este comando no funciona en modo 48K.
- **romsback**: Copia a un fichero RomPack, llamado **ROMS.ZX1**, en el directorio raíz de la tarjeta microSD todas las ROMS del core ZX Spectrum almacenadas en la memoria SPI Flash. Sólo funciona correctamente en ZX-Uno y ZX DOS (no utilizar en ZX DOS+ o gomaDOS+). Se debe ejecutar desde una ROM en modo "root".
- **romsupgr**: Copia el contenido de un fichero RomPack, llamado **ROMS.ZX1**, en el directorio raíz de la tarjeta microSD con todas las ROMS para el core ZX Spectrum a la memoria SPI Flash. Se debe ejecutar desde una ROM en modo "root".
- **upgr16m**: Copia el contenido de un fichero **FLASH.ZX1** en el directorio raíz de la tarjeta microSD a una memoria SPI Flash de 16 megas. Se debe ejecutar desde una ROM en modo "root".
- **upgrzx2** o **upgrzxd**: Copia el contenido de un fichero **FLASH_32.ZX2** o **FLASH_32.ZXD** a una memoria SPI Flash de 32 megas. Versión del comando upgrade exclusivo para memorias SPI Flash de 32 Megas. Se debe ejecutar desde una ROM en modo "root".
- **zxuc**: Configura todas las opciones de la BIOS, permitiendo grabar en la microSD las opciones seleccionadas en archivos de configuración que pueden posteriormente ser cargados (se puede obtener [en el repositorio de Utodev](#)).
- **zxunocfg**: Configura determinados aspectos del funcionamiento del ZX-Uno como los timings, la

contención, el tipo de teclado, la velocidad de la CPU, el tipo y frecuencia vertical del vídeo.



La mayoría de estos comandos se pueden obtener en el [el repositorio oficial de ZX-Uno](#).

Wi-Fi

Todos los gomaDOS+, y algunos modelos de ZX DOS+, tienen incorporado un módulo ESP-12 con un chip Wi-Fi [ESP8266](#), que se puede utilizar fácilmente con un core de ZX Spectrum (por ejemplo, el core EXP27 160820) que tenga sintetizado un dispositivo [UART](#), que permite la comunicación con el módulo.

Para configurar de forma básica el acceso al módulo, existen dos comandos "DOT" que se pueden obtener desde [el repositorio oficial en GitHub](#):

- [esprst](#), que sirve para reiniciar el módulo
- [iwconfig](#), que se utiliza para indicar el identificador (SSID) y la contraseña de la red Wi-Fi a la que conectarse, que quedarán almacenados en el fichero [/sys/config/iw.cfg](#) para que puedan usarlos otros programas.

Por ejemplo:

```
.iwconfig miwifi miclavedeacceso
```



Debido a que la frecuencia seleccionada para VGA afecta a la frecuencia del reloj maestro, para que el módulo Wi-Fi se comunique correctamente con el core, esta debe estar configurada a 50 (véase el [apartado dedicado a la configuración de la BIOS](#)).



Todo el software de Wi-Fi que se indica a continuación está disponible con [las distribuciones para ZX-Uno de desubikado](#)

Network tools for ZX-Uno

Se trata de un conjunto de programas para ZX Spectrum, desarrollados por Nihirash algunos de los cuales se pueden [ver en su web](#).

- **netman**: Utilidad sencilla para configurar la conexión Wi-Fi para el resto de programas. No funciona en modo 48K. Disponible para descarga [en GitHub](#).
- **Moon Rabbit**: Cliente de [Gopher](#). No funciona en modo 48K. Disponible para descarga [en GitHub](#)
- **irc**: Cliente de [Internet Relay Chat](#). Funciona mejor a 14 MHz.
- **wget**: Utilidad para descargar ficheros vía HTTP (no funciona con HTTPS).
- **platoUNO**: Cliente de [PLATO](#). También funciona mejor a 14 MHz. Para más información sobre el uso moderno de PLATO, es interesante ver los artículos en [el foro de ZX-Uno](#) así como la web de [IRATA.ONLINE](#).

FTP-Uno

Cliente de FTP desarrollado para ZX Spectrum por Yombo, disponible [en GitHub](#).

Para utilizarlo, se deben seguir los siguientes pasos:

1. Editar el archivo **FTP.CFG** con los datos necesarios (Wi-Fi, servidor FTP al que conectar, etc.)
2. Copiar **FTP.CFG** en **/SYS/CONFIG/** en la tarjeta microSD
3. Copiar también **ftpUno.tap** al lugar que desee de la tarjeta
4. Iniciar el ZXDOS+ y cargar el archivo de cinta **ftpUno.tap**

UART Terminal

Se trata de un programa de ejemplo para ZX Spectrum incluido con la biblioteca de funciones C [ZXYLib](#) desarrollada por yombo, y que permite enviar directamente pulsaciones de teclado a través del UART, y ver el resultado. Se puede descargar [en este enlace](#).

Una vez copiado el fichero de cinta [UARTTERM.tap](#) y cargado, se pueden teclear distintos comandos específicos para el chip ESP8266. Por ejemplo:

- **AT**. Para verificar si hay comunicación con el chipo. El resultado normal, si todo está bien, sería **OK**
- **AT+RST**. Para reiniciar el chip. Es exactamente lo mismo que hace el comando **esprst**
- **AT+GMR**. Para ver información relativa al chip, versión de firmware instalado, etc
- **AT+CWMODE_CUR=1**. Para configurar el chip en modo cliente Wi-Fi de forma temporal, hasta el próximo reinicio
- **AT+CWMODE_DEF=1**. Para configurar el chip en modo cliente Wi-Fi y guardar el ajuste como opción por defecto
- **AT+CWJAP_CUR="<RedWiFi>","<ContraseñaWiFi>"**, donde **<RedWiFi>** es el ID de a red Wi-Fi donde conectar, y **<ContraseñaWiFi>** la contraseña de acceso, conecta temporalente a la red indicada
- **AT+CWJAP_DEF="<RedWiFi>","<ContraseñaWiFi>"**, conecta a la red indicada, y la guarda como red por defecto en la memoria del chip
- **AT+CWAUTOCONN=1** configura el chip para conectarse a la red Wi-Fi por defecto al encenderse (**AT+CWAUTOCONN=0** desactiva esta opción)

Se pueden consultar todos los comandos disponibles en [la documentación oficial del fabricante](#).

Creación de vídeos RDM (RaDastan Movie)

El comando `PLAYRMOV` reproduce vídeos en formato radastaniano. Para poder convertir nuestros propios vídeos, se debe obtener la utilidad `makevideoradas` desde el [Repositorio SVN](#).

En el caso de Windows, en el propio repositorio hay un ejecutable (`makevideoradas.exe`) ya preparado. Para Linux o macOS, será necesario tener las herramientas de desarrollo correspondientes y compilarlo.

```
gcc makevideoradas.c -o makevideoradas
```

Una vez dispongamos de `makevideoradas`, necesitaremos otras dos herramientas: `ffmpeg` e `imagemagick`. Estas se pueden instalar con el gestor de paquetes correspondiente (`apt`, `yum`, `pacman`, `brew`, etc.) o descargando el código fuente y compilándolo también.

Ahora, el primer paso para convertir nuestro vídeo (por ejemplo, `mivideo.mp4`), es exportar los fotogramas como imágenes BMP de 128x96 píxeles de tamaño. Crearemos un directorio temporal (`img` en este ejemplo), donde guardar dichas imágenes.

```
mkdir img  
(...)/ffmpeg -i mivideo.mp4 -vf "scale=128:96,fps=25" -sws_flags lanczos -sws_dither  
ed -pix_fmt rgb4 -start_number 0 img/output%05d.bmp
```

Ahora transformaremos los ficheros `BMP` a `BMP` (v3) de 16 colores.

```
(...)/magick mogrify -colors 16 -format bmp -define bmp:format=bmp3 img/*.bmp
```

Finalmente, creamos el fichero `.RDM` (en este ejemplo `mivideo.rdm`) y borramos las imágenes y el directorio temporal.

```
(...)/makevideoradas img/output  
mv img/output.rdm ../mivideo.rdm  
rm -rf img
```

En [este hilo del foro Zona de Pruebas](#) hay más información sobre todo este proceso.

FUZIX

FUZIX es una fusión de varios elementos de [UZI](#) (una implementación del kernel Unix para ordenadores basados en Z80), extendida desde la versión 7 a algo entre SYS3 y SYS5.x con algunas partes de POSIX.

Aún no es algo útil, aunque se puede construir, arrancar y ejecutar aplicaciones. Aún falta mucho trabajo tanto en las utilidades como las bibliotecas.

En el momento de escribir estas líneas, las [imágenes oficiales](#) no funcionan con el ZX DOS+. Sin embargo, otro código más reciente, sí. Las siguientes instrucciones han sido probadas con [el último código disponible en Junio de 2021](#).

Cómo compilar

Estas instrucciones han sido elaboradas usando una instalación limpia de Fedora Workstation Linux (Fedora 34). Aparte de los comandos relativos a la instalación de paquetes, el resto de pasos deberían servir también con muchas otras distribuciones de Linux.

Primero, instalar los paquetes necesarios, si no estuvieran ya:

```
sudo dnf groupinstall -y 'Development Tools'  
sudo dnf install -y gcc-c++ automake boost-devel gutils flex texinfo bison byacc
```

Obtener la versión especial del [compilador SDCC](#) para Fuzix, e instalarla:

```
git clone https://github.com/EtchedPixels/sdcc280.git  
  
cd sdcc280  
cd sdcc  
.configure  
make  
sudo make install  
cd ../../
```

Obtener el código fuente de Fuzix:

```
git clone https://github.com/EtchedPixels/FUZIX.git  
cd FUZIX
```

Modificar el fichero [Makefile](#) y cambiar la línea que pone `TARGET=` para que sea `TARGET=zxdiv`. Compilar:

```
sudo make
```

Cuando termine, obtener el fichero binario de esxdos y la imagen del kernel en estas rutas:

```
./Kernel/platform-zxdiv/FUZIX  
./Kernel/platform-zxdiv/FUZIX.BIN
```

Construir el sistema de archivos raíz:

```
cd ./Standalone/filesystem-src  
./build-filesystem rootfs 256 65535  
cd ../../
```

Obtener el archivo de imagen en esta ubicación:

```
./Standalone/filesystem-src/rootfs
```

Cómo utilizarlo

Hace falta que la tarjeta microSD tenga una tabla de particiones MBR. Se pueden configurar una o dos particiones primarias [de la forma habitual](#) (la primera de ellas con una instalación funcional de esxdos), dejando suficiente espacio libre al final para añadir una partición de 32MB (Tipo [7E](#)) para el sistema de archivos raíz y otra partición de 4MB (Tipo [4E](#)) para el espacio de intercambio (swap).

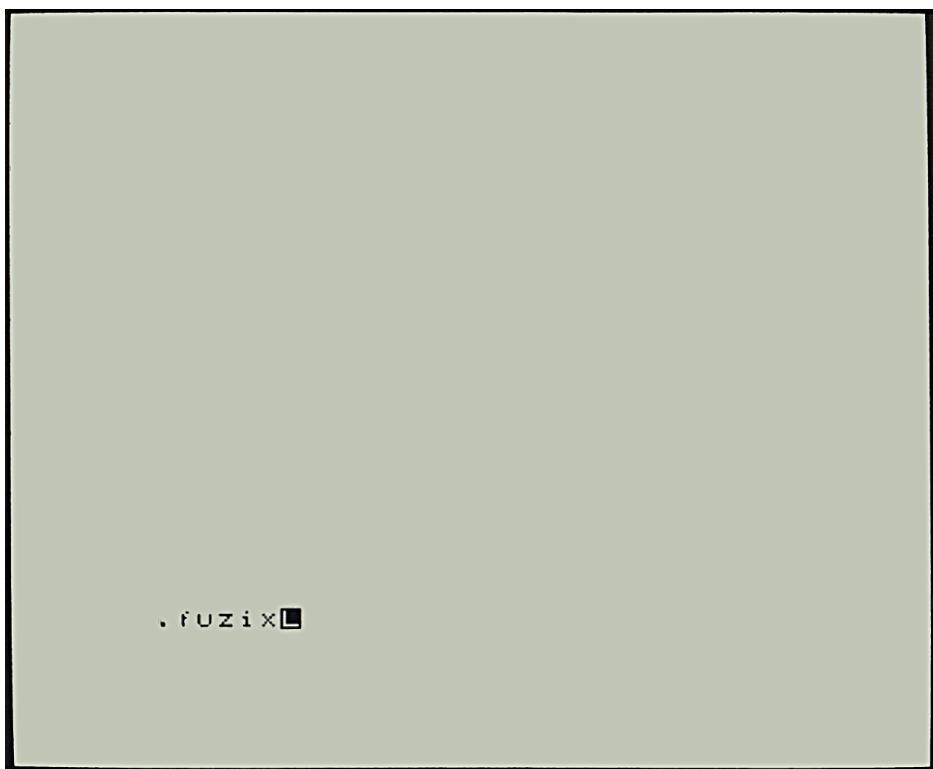
Volcar el contenido del sistema de archivos raíz en la partición [7E](#). Para ello, se puede usar la utilidad [dd](#), incluida en Linux, macOS, etc. (y también [convertida para Windows](#)).

Una vez determinado el nombre del dispositivo asociado a la partición [7E](#) se debe usar esta como destino del fichero [rootfs](#). Por ejemplo, para [/dev/rdisks3](#):

```
sudo dd if=rootfs of=/dev/rdisks3
```

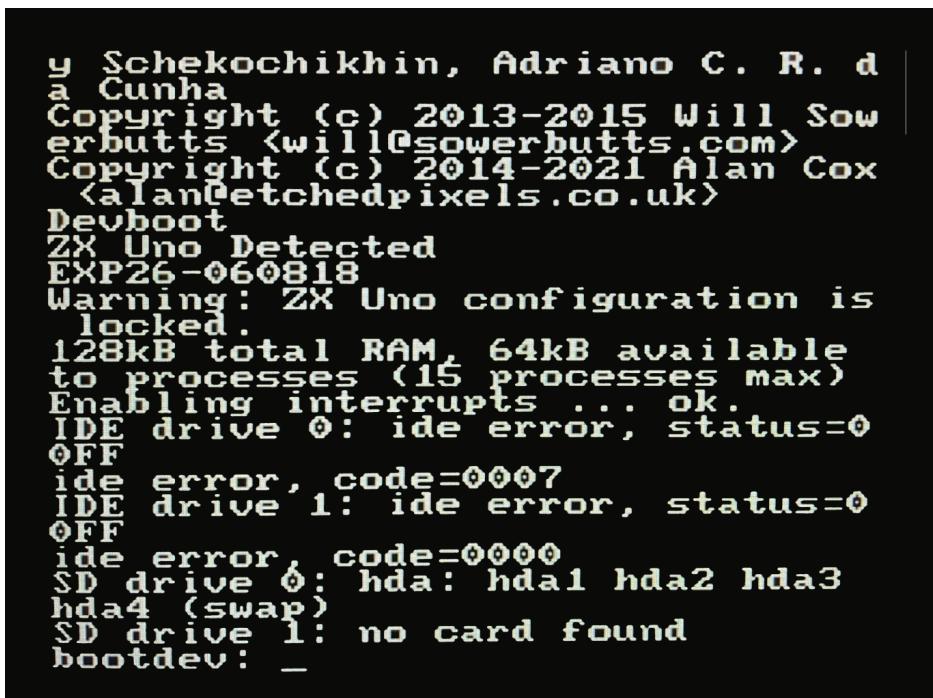
Copiar el fichero [FUZIX](#) dentro del directorio [BIN](#) y copiar [FUZIX.BIN](#) al directorio raíz de la particion esxdos.

Iniciar un core de Spectrum con una ROM de 128K y con esxdos, escribir **.fuzix**, y pulsar **Enter**.



La configuración del teclado en la BIOS debería ser con una disposición en inglés, o no será posible teclear algunos caracteres, como **|**.

Al cabo de unos pocos segundos, el sistema debería detectar la tarjeta microSD y encontrar las particiones. En este ejemplo, la raíz está en la tercera partición de SD 0.



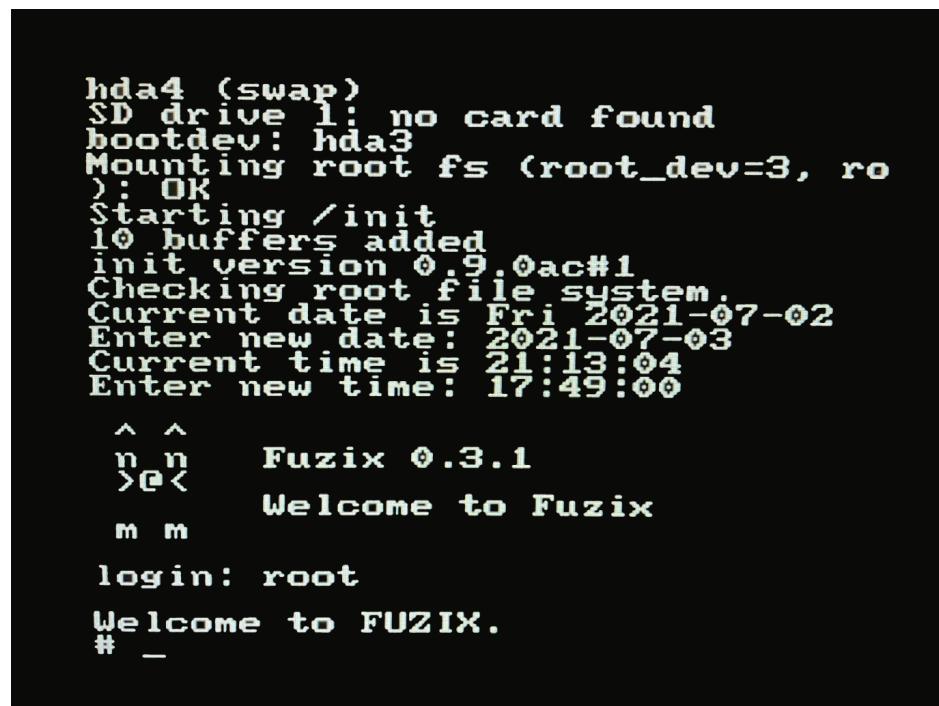
Escribir el identificador de la partición raíz (por ejemplo, **hd3**) y pulsar **Enter**.

```
Devboot
ZX Uno Detected
EXP26-060818
Warning: ZX Uno configuration is
locked.
128kB total RAM, 64kB available
to processes (15 processes max)
Enabling interrupts ..: ok.
IDE drive 0: ide error; status=0
OFF
ide error, code=0007
IDE drive 1: ide error, status=0
OFF
ide error, code=0000
SD drive 0: hda: hda1 hda2 hda3
hda4 (swap)
SD drive 1: no card found
bootdev: hda3
Mounting root fs (root_dev=3, ro
): OK
Starting /init
10 buffers added
init version 0.9.0ac#1
-
```

Definir la fecha (pulsar **Enter**) y la hora (pulsar **Enter**).

```
locked.
128kB total RAM, 64kB available
to processes (15 processes max)
Enabling interrupts ..: ok.
IDE drive 0: ide error; status=0
OFF
ide error, code=0007
IDE drive 1: ide error, status=0
OFF
ide error, code=0000
SD drive 0: hda: hda1 hda2 hda3
hda4 (swap)
SD drive 1: no card found
bootdev: hda3
Mounting root fs (root_dev=3, ro
): OK
Starting /init
10 buffers added
init version 0.9.0ac#1
Checking root file system.
Current date is Fri 2021-07-02
Enter new date: 2021-07-03
Current time is 21:13:04
Enter new time: 17:49:00_
```

Iniciar sesión con el usuario **root** y sin contraseña.



```
hda4 (swap)
SD drive 1: no card found
bootdev: hda3
Mounting root fs (root_dev=3, ro)
): OK
Starting /init
10 buffers added
init version 0.9.0ac#1
Checking root file system.
Current date is Fri 2021-07-02
Enter new date: 2021-07-03
Current time is 21:13:04
Enter new time: 17:49:00
^ ^
n n   Fuzix 0.3.1
>@<   Welcome to Fuzix
m m
login: root
Welcome to FUZIX.
# _
```

Con esto ya se dispone de una consola de Fuzix.



Al terminar, recordar detener el sistema usando el comando **shutdown** o el sistema de archivos raíz quedará marcado como no limpio, y esto forzará un comprobación del mismo en el próximo arranque de Fuzix.

Actualizaciones

BIOS

Para actualizar BIOS se ha de obtener un fichero llamado **FIRMWARE.ZX2** (para un ZX DOS+ con placa FPGA LX16) o **FIRMWARE.ZXD** (para un ZX DOS+ con placa FPGA LX25). La última versión de los ficheros de firmware se puede descargar desde [el repositorio oficial](#)



Actualizar el firmware (BIOS) es delicado, no se debe hacer si no es necesario. En el caso de hacerlo, procurar que el ZX DOS+ tenga alimentación ininterrumpida (como un SAI o un USB de portatil con batería).

Copiar el fichero en la raíz de la tarjeta MicroSD, encender y pulsar **F2** para entrar en la BIOS, seleccionar **Upgrade**, elegir "*Upgrade BIOS for ZX*", y luego "*SDfile*". El sistema leerá el fichero **FIRMWARE…** y avisará cuando esté actualizado.

ROMs

La memoria flash del ZX DOS+ dispone de 64 "slots", de 16K cada uno, para almacenar imágenes ROM de ZX Spectrum y compatibles. Así, la ROM del ZX Spectrum original (16K) ocuparía un slot del almacenamiento, la del ZX Spectrum 128K (32K) ocuparía dos slots, y la del ZX Spectrum +2A (64K) ocuparía 4 slots.

Se puede añadir una nueva ROM desde [la pantalla ROMs](#) de la BIOS, pulsando la tecla **N**, conectando un cable de audio a la entrada de sonido de la placa, y reproduciendo una cinta de carga de ROM. Las cintas de carga de ROM se pueden crear desde un archivo **.tap** generado con la utilidad **GenRom**, disponible en el [repositorio de código de ZX-Uno](#).

Para actualizar las ROM instaladas para ZX Spectrum de forma masiva, se ha de obtener un fichero RomPack con el nombre **ROMS.ZX1**, y se tiene que copiar en la tarjeta MicroSD. Arrancar el ZX DOS+ usando una ROM en modo "root", y entonces bastará con introducir el comando **.romsupgr**. Esto grabará todas las ROM, que quedarán disponibles para su uso.



Recordar que, si se inicia el ZX DOS+ pulsando la tecla **/** (del teclado numérico, **Symbol Shift+V** en gomaDOS+), entonces se cargará la ROM por defecto del core de ZX Spectrum en modo "root".

Para hacer el proceso contrario (guardar las ROMs en un fichero RomPack llamado **ROMS.ZX1**), se puede usar el comando **.romsback**.



La versión actual de **romsback**, en ZX DOS+, sólo almacena correctamente en RomPack los primeros 35 slots de ROM ocupados.

Los ficheros RomPack se pueden editar fácilmente con la utilidad [ZX1RomPack](#). Aunque es un programa de Windows, funciona perfectamente, por ejemplo, usando [Wine](#) o programas similares, tanto en macOS como en Linux.

Cores

Hay un número de espacios disponibles para almacenar cores (el número depende del tamaño de la SPI Flash del modelo de ZXDOS), estando reservado el primer espacio para el de ZX Spectrum principal (esto no impide tener más cores de ZX Spectrum en otros espacios además del primero).

Los cores oficiales están [disponibles para descargar](#) en el repositorio en GitHub.

Para actualizar o instalar un nuevo core hay varias alternativas.

La forma más sencilla consiste en obtener la última versión del fichero que lo define, que será un fichero que hay que llamar **COREnn.ZX2** (para un ZXDOS+ con placa FPGA LX16) o **COREnn.ZXD** (para un ZXDOS+ con placa FPGA LX25), donde **nn** es el número de espacio donde realizar la instalación (por ejemplo **CORE2.ZX2** o **CORE2.ZXD** para el espacio 2).



A partir de la versión 0.80 de BIOS, los ficheros se nombran usando la convención **COREXXy.ZXn** donde XX *siempre* es un número de dos dígitos. Así, un antiguo fichero **CORE4.ZXD** ha de renombrarse como **CORE04.ZXD**. La parte **y** del nombre se ignora, así que se pueden usar nombres más largos y descriptivos (como, por ejemplo, **CORE04_ejemplo.ZXD**).

Copiar el fichero en la raíz de la tarjeta microSD, encender y pulsar **F2** para entrar en la BIOS. Elegir **Upgrade**, seleccionar la fila correspondiente al número de core elegido (por ejemplo, la 2 – justo después de la de Spectrum), pulsar enter y luego "SD file". El sistema leerá el fichero **COREnn…** y avisará cuando esté actualizado, aunque antes preguntará el nombre (con el que se verá en la lista para elegir en el arranque y en el listado de la BIOS). Una vez instalado, se podrá utilizar al arrancar.



La actualización del core de ZX Spectrum es exactamente igual que los otros cores, pero en lugar del fichero **CORE1.ZX2** o **CORE1.ZXD**, ha de ser un fichero llamado **SPECTRUM.ZX2** o **SPECTRUM.ZXD**.



Debido a limitaciones en cómo funciona internamente la FPGA, los cores almacenados en la segunda parte de la SPI flash han de ser instalados sin utilizar el espacio [en la ubicación 0x10B0000](#). En dicho espacio, se instala [un core especial](#) que garantiza que, al reiniciar otro core instalado en la mitad superior de la flash, se sigue cargando correctamente la BIOS con el core de Spectrum principal.

esxdos

Para actualizar esxdos a una nueva versión, se ha de obtener la distribución desde [la página oficial](#).

Una vez descargado y descomprimido, se ha de copiar, a la raíz de la tarjeta, el contenido de los directorios **BIN** y **SYS** sobreescribiendo los existentes (para preservar los comandos exclusivos de ZX DOS+).

Copiar **ESMMC.BIN** (o **ESMMC.ROM**, según la versión) en la raíz de la tarjeta microSD.

Iniciar el ZX DOS+ con la tarjeta insertada y pulsar F2 para acceder a la configuración de BIOS. Seleccionar el menú **Upgrade** y elegir "*Upgrade esxdos for ZX*". En el diálogo que aparece elegir "*SD file*" y, cuando pregunte "*Load from SD*" contestar "Yes" a la pregunta "*Are you sure?*". Se leerá el contenido del fichero **ESXDOS...**, se grabará en la flash y avisará cuando esté actualizado.

Realizar un Hard-reset, o apagar y encender.

Si todo se ha hecho correctamente, al encender el ZX DOS+ se verá cómo esxdos detecta la tarjeta y carga los componentes necesarios para funcionar, mostrando la nueva versión en la parte superior.

Memoria Flash

También es posible actualizar la memoria flash de la FPGA. Por el momento desde el menú de la BIOS sólo es posible utilizar imágenes de 16MiB. Para poder usar una imagen de 32MiB, se ha de usar el comando **UPGRZX2** o **UPGRZXD** de **esxdos** y un fichero con el nombre **FLASH_32.ZX2** o **FLASH_32.ZXD**.

Copiar el archivo de imagen (de 16MiB) **FLASH.ZXD** o **FLASH.ZX2** en la raíz de la tarjeta microSD.

Iniciar el ZX DOS+ con la tarjeta insertada y pulsar F2 (**Caps Shift+1** en gomaDOS+) para acceder a la configuración de BIOS. Seleccionar el menú **Upgrade** y elegir "*Upgrade flash from SD*". En el diálogo que pregunta "*Load from SD*" contestar "Yes" a la pregunta "*Are you sure?*". Se leerá el contenido del fichero **FLASH...**.

Realizar un Hard-reset, o apagar y encender.



Este proceso sustituye todos los cores instalados, la BIOS, así como las ROMs de ZX Spectrum y la configuración por lo que haya en la imagen, y no se puede deshacer.

Otros cores

ZX Spectrum 48K (Kyp)

[Core alternativo](#), cuyo objetivo es ser una implementación de un Spectrum 48K que sea lo más exacta posible en cuanto a la configuración de los tiempos (timing), contención de memoria, etc.

Sus características principales son:

- Salida de vídeo RGB y VGA
- Specdrum
- Turbosound (dos chips AY) con posibilidad de elegir mix ACB/ABC
- DivMMC con esxdos 0.8.8
- Joystick Kempston en el puerto 1
- Salida de audio I²S (con el [addon RTC+I²S+Pizero](#))

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD con la primera partición en formato FAT16 o FAT32, y que tenga instalada la distribución de esxdos 0.8.8 (ver [el apartado correspondiente de esxdos](#) para más información).

Teclado

Teclas especiales y botones

Durante la ejecución del core:

- **Esc** ([Caps Shift+Espacio](#) en gomaDOS+): BREAK
- **F5** ([Caps Shift+Symbol Shift+5](#) en gomaDOS+): NMI
- **F8** ([Caps Shift+Symbol Shift+8](#) en gomaDOS+): Alternar la configuración de mezcla de Turbosound entre ACB y ABC.
- **Ctrl+Alt+Backspace** ([Caps Shift+Symbol Shift+B](#) en gomaDOS+) o **F11** ([Caps Shift+Symbol Shift+Q](#) en gomaDOS+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#).
- **Ctrl+Alt+Supr** ([Caps Shift+Symbol Shift+N](#) en gomaDOS+) o **F12** ([Caps Shift+Symbol Shift+W](#) en gomaDOS+): Soft reset.

ZX Spectrum 128K (Kyp)

Core alternativo, cuyo objetivo es ser una implementación de un Spectrum +2 (Gris) que sea lo más exacta posible en cuanto a la configuración de los tiempos (timing), contención de memoria, etc.

Sus características principales son:

- Salida de vídeo RGB y VGA
- Specdrum
- Turbosound (dos chips AY) con posibilidad de elegir mix ACB/ABC
- DivMMC con esxdos 0.8.8
- Joystick Kempston en el puerto 1
- Salida de audio I²S (con el [addon RTC+I²S+Pizero](#))

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD con la primera partición en formato FAT16 o FAT32, y que tenga instalada la distribución de esxdos 0.8.8 (ver [el apartado correspondiente de esxdos](#) para más información).

Teclado

Teclas especiales y botones

Durante la ejecución del core:

- Esc ([Caps Shift+Espacio](#) en gomaDOS+): BREAK
- F5 ([Caps Shift+Symbol Shift+5](#) en gomaDOS+): NMI
- F8 ([Caps Shift+Symbol Shift+8](#) en gomaDOS+): Alternar la configuración de mezcla de Turbosound entre ACB y ABC.
- Ctrl+Alt+Backspace ([Caps Shift+Symbol Shift+B](#) en gomaDOS+) o F11 ([Caps Shift+Symbol Shift+Q](#) en gomaDOS+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#).
- Ctrl+Alt+Supr ([Caps Shift+Symbol Shift+N](#) en gomaDOS+) o F12 ([Caps Shift+Symbol Shift+W](#) en gomaDOS+): Soft reset.

ZX Spectrum Next

[ZX Spectrum Next](#) es un proyecto, basado en FPGA, que aspira a ser la evolución de los ordenadores Sinclair ZX Spectrum, manteniendo la compatibilidad hardware y software con los modelos anteriores, pero añadiendo nuevas características.

Principalmente gracias a avlixa, existe una versión del core de ZX Spectrum Next sintetizada para usarse con ZXDOS+.

El core para ZXDOS+ no tiene, por el momento, implementada ninguna de las siguientes características:

- Beeper interno
- Conector de expansión EDGE
- Módulo RTC
- Teclado de membrana
- Grabación de cores adicionales o actualización del propio core Next desde el core Next
- Salida MIC
- vídeo HDMI
- Utilización de puerto de conexión joystick para comunicación UART

Por otra parte, puede disponer de las siguientes características, que no existen en el core original:

- Modos de color incluyendo monocromo o fósforo verde/naranja
- Salida de audio I²S (con el [addon RTC+I²S+Pizero](#))

El manual de uso se puede descargar desde [la página oficial](#).



Para poder utilizar una Raspberry Pi, se necesita una versión del Core con el soporte necesario, y el addon RTC+I²S+Pizero. Ver la [sección correspondiente](#) de otro hardware para más detalles.

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD con la primera partición en formato FAT16 o FAT32, y que tenga instalada la distribución de esxdos correspondiente a la configuración actual de BIOS (ver el apartado correspondiente de esxdos para más información).

Obtener la distribución de NextZXOS [en la página oficial](#).

Descomprimir el contenido de NextZXOS en la tarjeta microSD, pero modificando el archivo `config.ini` en `/machines/next` para que contenga (si no existiera ya) la línea `ps2=0` (para asegurar que se utiliza correctamente el puerto del teclado) y la línea `intbeep=0` para apagar el zumbador interno (este último paso no es necesario en el caso de un gomaDOS+).

Si no estuviera ya, [instalar el core de ZX Spectrum Next](#) en el ZX DOS+.

Teclado

Teclas especiales y botones

Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado **ZX**. Véase el apartado dedicado a los modos de teclado de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado **PC XT**.

Notar que `Ctrl+Alt+backspace` no funciona con el core de Spectrum Next. Hay que apagar manualmente y volver a encender si se desea cambiar a otro core. Tampoco hay botón físico de Reset o Drive.

Durante la ejecución del core:

- **F1** (`Caps Shift+Symbol Shift+1` en gomaDOS+): Hard Reset
- **F2** (`Caps Shift+Symbol Shift+2` en gomaDOS+): Scandoubler. Dobla la resolución. Debería estar apagado para conexiones vía SCART
- **F3** (`Caps Shift+Symbol Shift+3` en gomaDOS+): Alternar la frecuencia vertical entre 50Hz y 60Hz
- **F4** (`Caps Shift+Symbol Shift+4` en gomaDOS+): Soft Reset
- **F7** (`Caps Shift+Symbol Shift+7` en gomaDOS+): Scanlines
- **F9** (`Caps Shift+Symbol Shift+9` en gomaDOS+): NMI
- **F10** (`Caps Shift+Symbol Shift+0` en gomaDOS+): divMMC NMI. Simula la pulsación del botón Drive. Si se usa con mayúsculas, fuerza volver a buscar unidades de almacenamiento y cargar la pantalla de arranque en esxdos
- **F11** (`Caps Shift+Symbol Shift+Q` en gomaDOS+): En las versiones más recientes del core, activar distintos modos de color monocromo
- **F12** (``Caps Shift+Symbol Shift+W` en gomaDOS+): Alternar entre la salida audio estándar y la salida I²S, si está conectado el [addon RTC+I²S+PIO](#). Notar que activar la salida I²S inhabilita parcialmente el sonido de la Raspberry Pi.

Guía básica

Al iniciarse la primera vez, aparecerán una serie de pantallas de ayuda. Tras pulsar la tecla **Espacio**, se mostrará el menú de inicio de NextZXOS.



Se puede navegar utilizando las teclas de cursor, las teclas **5**, **6**, **7** y **8**, o un joystick (si se ha configurado en modo Kempston, MD o cursor). **Enter** o el botón del joystick selecciona un elemento.

La opción **More...** muestra un segundo menú con más opciones.



Si se elige **Browser**, se cargará el navegador de NextZXOS, desde el que es posible desplazarse viendo el contenido de la tarjeta microSD y cargar directamente diferentes tipos de archivo (TAP, NEX, DSK, SNA, SNX, Z80, Z8, etc.).



El navegador muestra las entradas de archivos y directorios en el orden de la tabla FAT interna, y no de manera alfabética. Si se desea ver esta información ordenada, se debe reorganizar la estructura de la tarjeta con una utilidad como **FAT Sorter** para Windows, **FATsort** para Linux y macOS, **YAFS**, **SDSorter** u otros.



En el caso de utilizarse también la tarjeta con el core de **PC XT**, **no se debe utilizar ninguna utilidad de reordenación de FAT** ya que esto puede provocar que deje de arrancar correctamente DOS en dicho core.

C:/
DEMONS <DIR>
DOCS <DIR>
DOT <DIR>
EXTRAS <DIR>
GAMES <DIR>
MACHINES <DIR>
NEXTZXOS <DIR>
RPI <DIR>
SRC <DIR>
SYS <DIR>
TMP <DIR>
TOOLS <DIR>
CONTRIBUTING.md
LICENSE.MD
README.MD
TBBLUE.FW
TBBLUE.TBU
DEVEL <DIR>
ALL <DIR>

Browser Filter: *.
Cursor keys & ENTER, BREAK=exit, EDIT=up re M ount
Drive m K dir R ename C opy E rase U nmount

El core de ZX Spectrum Next para ZX DOS+ necesita el [acelerador basado en Raspberry Pi](#) para cargar ficheros TZX.



Por defecto, no es posible cargar ficheros TRD desde el navegador (se debe configurar NextZXOS para cargar una "personalidad" con esxdos).

Para más información, consultar el [manual de uso oficial](#).

MSX

MSX1FPGA es un proyecto para clonar MSX1. El desarrollo original es de Fabio Belavenuto y se encuentra disponible en [GitHub](#).

Algunas de sus características son:

- MSX1 a 50Hz o 60Hz;
- Utiliza Nextor ROM con un controlador para SD
- Mapa de teclado configurable
- Simulación de línea de exploración (Scanlines)
- Soporte para joystick
- Salida de audio I²S (con el [addon RTC+I²S+Pizero](#))

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD con la primera partición en formato FAT16 con [código 0x06 \(16-bit FAT\)](#). Es también posible utilizar una segunda partición FAT16 para albergar todo el software, dejando la primera sólo para arrancar el sistema.

Obtener lo siguiente:

- Ficheros básicos del proyecto para la microSD [desde GitHub](#)
- Controlador ([NEXTOR.SYS](#)) y ROM ([NEXTOR.ROM](#)) de Nextor [también desde GitHub](#)
- ROM de MSX1 ([MSX_INT.rom](#), [MSX_JP.rom](#) o [MSX_USA.rom](#)) en el mismo repositorio

Copiar el contenido del [directorio SD](#) en la raíz de la primera partición de la tarjeta microSD.



Como algunos de los directorios y ficheros de sistema de DOS pueden tener el mismo nombre, no se recomienda utilizar la misma tarjeta para el [core de PC XT](#) y el de MSX.

Copiar [NEXTOR.SYS](#) en el mismo lugar.

Copiar [NEXTOR.ROM](#) en el directorio [MSX1FPGA](#).

Copiar la ROM deseada de MSX1 ([MSX_INT.rom](#), [MSX_JP.rom](#) o [MSX_USA.rom](#)) en el directorio [MSX1FPGA](#), pero usando el nombre [MSX1BIOS.ROM](#).

En el fichero [/MSX1FPGA/config.txt](#) se guarda la configuración del core, según este formato:

```
11SP01
|||||
|||||+-Modo de línea de exploración: 1=Activo, 0=Inactivo
|||||---Turbo: 1=Arrancar con el modo turbo activo
|||+---Sistema de color: N=NTSC, P=PAL
||+---Mapa de Teclado: E=Inglés, B=Brasileño, F=Francés, S=Castellano
```

```
|+----Scandoubler(VGA): 1=Activo, 0=Inactivo  
+----Nextor: 1=Activo, 0=Inactivo
```

Si no estuviera ya, [instalar el core de MSX en el ZXDOS+](#).

Teclado

Teclas especiales y botones

Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado **MSX**. Véase el [apartado dedicado a los modos de teclado](#) de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado **PC XT**.

Durante la ejecución del core:

- **Impr Pant**: Cambia el modo entre VGA y RGB
- **Bloq Desp (Caps Shift+Symbol Shift+G en gomaDOS+)**: Cambia el modo de línea de exploración (Scanlines)
- **Pausa**: Cambia entre 50Hz y 60Hz
- **F11 (Caps Shift+Symbol Shift+Q en gomaDOS+)**: Activa o desactiva el modo turbo
- **Ctrl+Alt+Supr**: Soft Reset
- **Ctrl+Alt+F12**: Hard Reset
- **Ctrl+Alt+Backspace (Caps Shift+Symbol Shift+B en gomaDOS+, en modo de teclado ZX Spectrum)**: Reinicia la FPGA
- **ALT Izquierdo**: MSX GRAPH
- **ALT Derecho**: MSX CODE
- **Re Pág**: MSX SELECT
- **Inicio**: MSX HOME (Mayús+HOME: CLS)
- **Fin**: MSX STOP
- **Ñ o Windows**: MSX DEAD



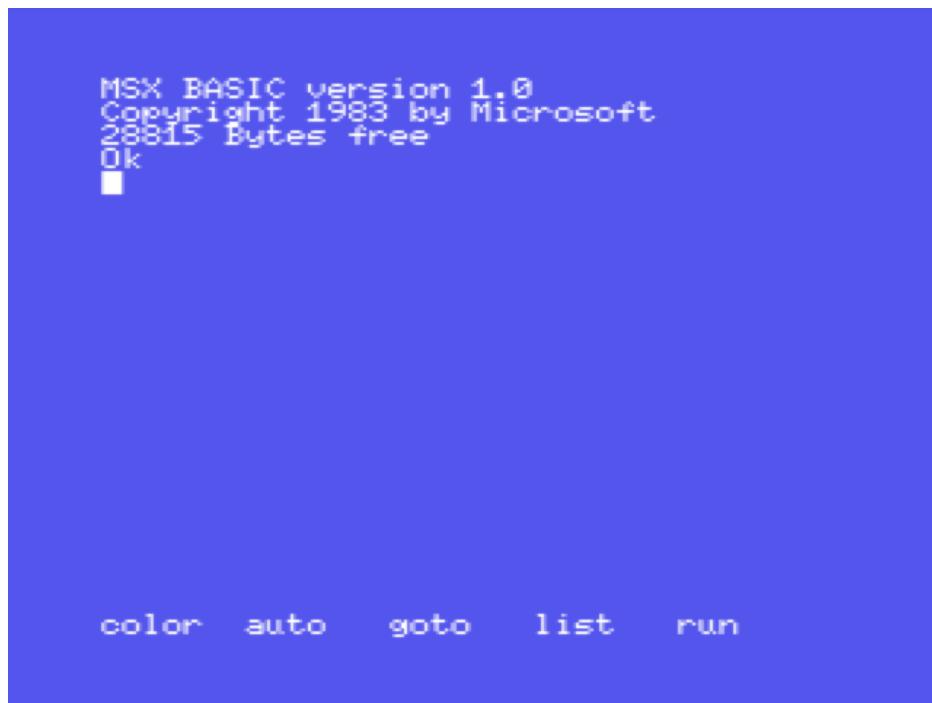
En BASIC, se puede usar **CTRL + STOP (Ctrl+Fin)** para detener la ejecución de un programa.



Para cambiar el modo de vídeo entre 50Hz y 60Hz (para ejecución correcta de programas PAL a través de VGA), se puede usar también **DISPLAY.COM**, que se puede obtener [en este hilo del foro de MSX](#).

Guía básica

Para acceder a BASIC desde MSX-DOS, ejecutar el comando **BASIC**.



MSX BASIC version 1.0
Copyright 1983 by Microsoft
28815 Bytes free
Ok
■

color auto goto list run

A screenshot of the MSX BASIC version 1.0 command prompt. The screen is blue with white text. At the top, it displays "MSX BASIC version 1.0", "Copyright 1983 by Microsoft", and "28815 Bytes free". Below that is the word "Ok" followed by a small black square icon. At the bottom of the screen, there is a horizontal menu bar with the words "color", "auto", "goto", "list", and "run" separated by spaces.

Desde BASIC, se puede cargar desde una cinta (u [otro dispositivo externo de audio](#)) con los comandos **RUN"CAS:", BLOAD"CAS:", R** o **CLOAD**.



Para que la carga desde audio funcione, el modo turbo tiene que estar desactivado.

Para acceder a MSX-DOS desde BASIC, ejecutar **CALL SYSTEM**.

MSXCTRL

Se trata de una utilidad exclusiva del core MSX1FPGA, que permite controlar todas las opciones del core que antes solo eran accesibles a través del fichero de configuración o pulsando determinadas teclas.

Al ejecutar **MSXCTRL** se muestran los parámetros de uso:

```
MSXCTRL.COM - Utility to manipulate MSX1FPGA core.  
HW ID = 06 - ZX-Uno Board  
Version 1.3  
Mem config = 82  
Has HWDS = FALSE
```

Use:

```
MSXCTRL -h -i -r -b -[5|6] -m<0-2>  
-c<0-1> -d<0-1> -t<0-1>  
[-w<filename> | -l<filename>]  
-k<0-255> -e<0-255> -p<0-255>  
-s<0-255> -o<0-255> -a<0-255>
```

MSXCTRL -h muestra ayuda para cada parámetro. Así, **MSXCTRL -i** presenta la configuración actual, los parámetros **-t 1** encienden el modo turbo, etc.

Otros

Existen múltiples sistemas para cargar los juegos dependiendo del tipo de archivo: .CAS, .DSK o ROM (ver [este hilo del foro de ZX-Uno](#) para más información).

El mapeo de para teclado español disponible con la distribución oficial se puede cambiar por otro más completo. Ver [aquí](#) para más información.

Amstrad CPC 6128

El core para ZXDOS+ de Amstrad CPC 6128 está basado en el proyecto [FPGAmstrad](#) de Renaud Hélias.

Algunas de sus características son:

- VGA: 640x480 VGA centrado a 60Hz
- Selección de discos: El primer disco detectado se inserta en el arranque y la pulsación de una tecla hace reset y carga el siguiente

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD con la primera partición en formato FAT32 (Tipo de partición **0B** Win95 FAT-32), de 4GB de tamaño máximo y 4096 bytes por cluster.

Además son necesarios los ficheros ROM siguientes (se pueden obtener [en la wiki oficial del proyecto original](#) o en el [repositorio de GitHub](#): - OS6128.ROM - BASIC1-1.ROM - AMSDOS.ROM - MAXAM.ROM

También es recomendable incluir uno o más ficheros con imágenes de disco (**DSK**) con el software que se quiera ejecutar.

Copiar tanto los ficheros **ROM** como los **DSK** a la raíz de la partición FAT32.

Teclado

Teclas especiales y botones

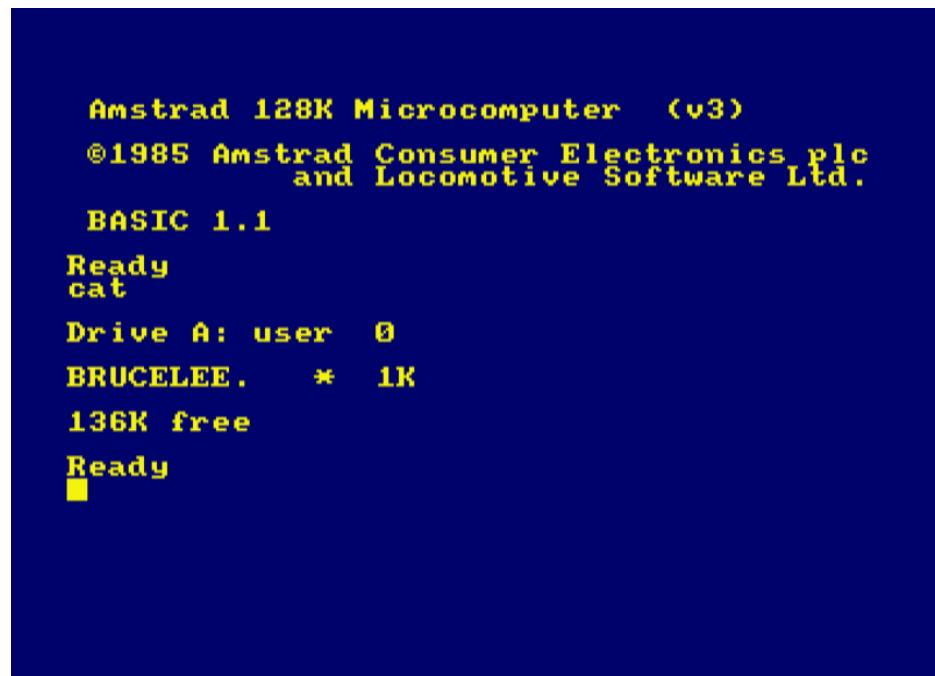
Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado **Amstrad CPC**. Véase el [apartado dedicado a los modos de teclado](#) de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado **PC XT**.

Durante la ejecución del core:

- **Re Pág** (**Caps Shift+Symbol Shift+E** en gomaDOS+): Hace un Reset del Amstrad y carga el siguiente archivo **DSK** en orden alfabético.
- En un teclado PS/2, sólo funciona la tecla mayúsculas del lado izquierdo del teclado.

Guía básica

Escribir el comando **CAT** para ver el contenido del fichero DSK cargado actualmente.



```
Amstrad 128K Microcomputer (v3)
©1985 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

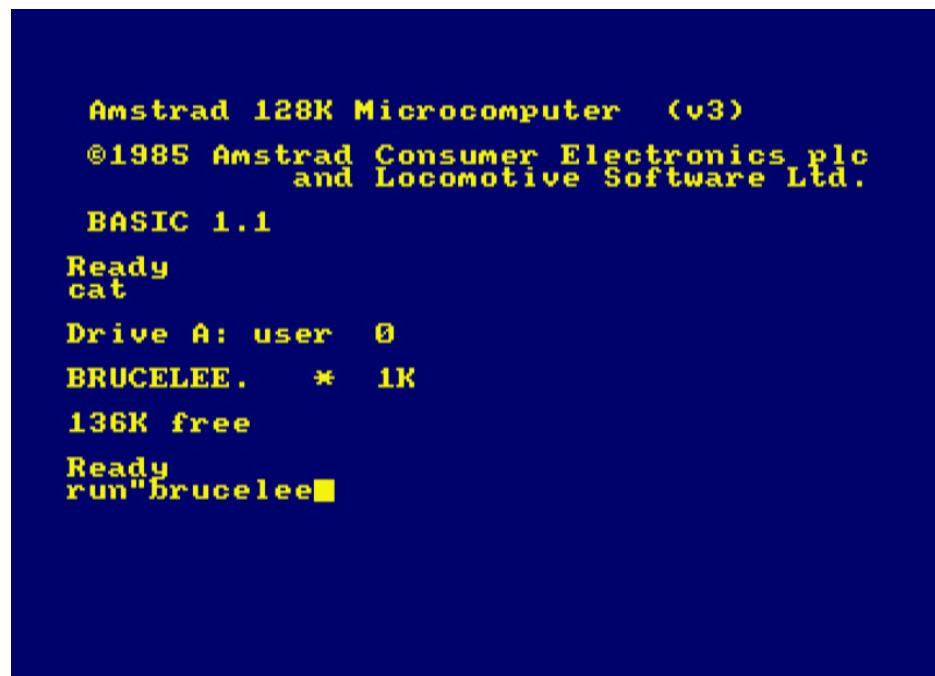
BASIC 1.1

Ready
cat

Drive A: user 0
BRUCELEE.* 1K
136K free

Ready■
```

Escribir el comando **RUN"<nombre>** para cargar un programa del disco



```
Amstrad 128K Microcomputer (v3)
©1985 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

BASIC 1.1

Ready
cat

Drive A: user 0
BRUCELEE.* 1K
136K free

Ready
run"brucelee"■
```

Usar la tecla **Re Pág** para hacer reset y cargar el siguiente archivo **DSK** en orden alfabético.

Acorn Atom

El [Acorn Atom](#) era un computador casero hecho por Acorn Computers. El core para ZXDOS+ (basado en el de ZX-Uno realizado por Quest) es una adaptación del proyecto [AtomFPGA](#). Se puede ver más información en [el foro de ZX-Uno](#).

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD con la primera partición en formato FAT16.

Descargar la última versión de Atom Software Archive [desde GitHub](#).

Ahora, se puede organizar la información en la tarjeta microSD de dos maneras distintas:

1. Descomprimir todo el contenido del archivo en la raíz de la tarjeta. El contenido del directorio **SYS** es compatible con el directorio **SYS** de esxdos, siendo posible combinar los dos en uno solo.
2. Organizar la información de una manera más reducida en la raíz, utilizando sólo dos directorios. Crear un directorio **ATOM** en la raíz de la tarjeta, y copiar en su interior todo el contenido del archivo, excepto el directorio **MANPAGES** que se tendrá que poner también en la raíz de la microSD. Luego, copiar los ficheros del archivo **trick_ATOM_folder** (disponible [en el foro de ZX-Uno](#)), reemplazando todos los que se encuentren con el mismo nombre. Así, quedará una estructura como la siguiente:

```
/  
+-ATOM/  
|   +-AA/  
|   (...)  
|   +-AGD/  
|   |   +-SHOW2  
|   |   +-SHOW3  
|   (...)  
|   +-MENU  
|   (...)  
|   +-TUBE/  
|   |   +-BOOT6502  
|   (...)  
|  
+-MANPAGES/  
|   +-CPM.MAN  
|   +-FLEX.MAN  
|   (...)  
|  
+-MENU
```

Teclado

Teclas especiales y botones

Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado **Acorn Electron**. Véase el [apartado dedicado a los modos de teclado](#) de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado **PC XT**.

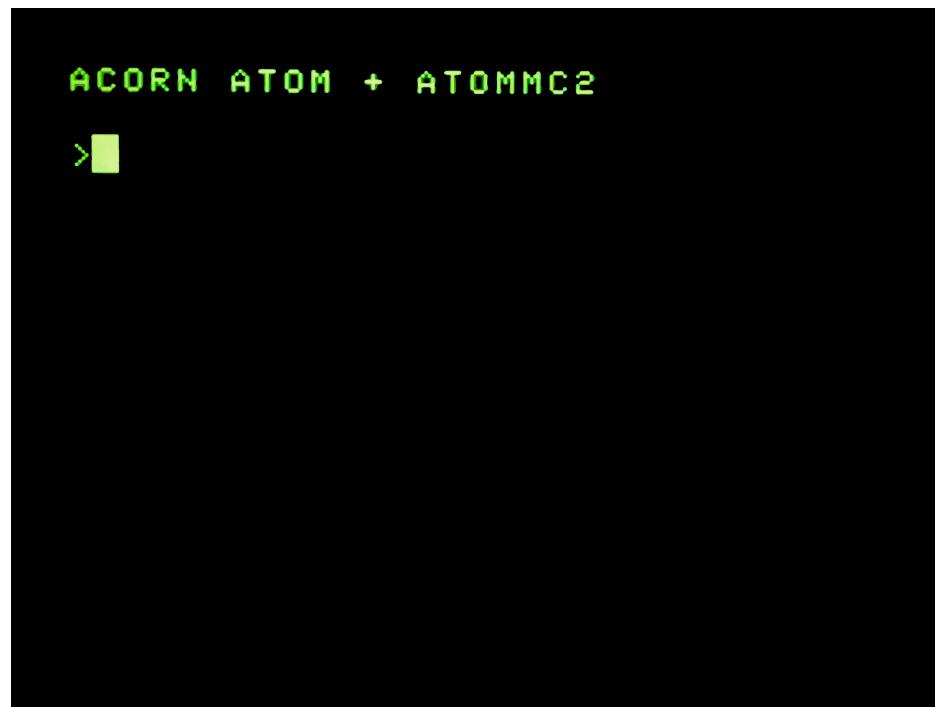
Durante la ejecución del core:

- **Mayús+F10**: Muestra el menú de Atom Software Archive
- **F10 (Caps Shift+Symbol Shift+0 en gomaDOS+)**: Soft Reset
- **F1 (Caps Shift+Symbol Shift+1 en gomaDOS+)**: Modo turbo 1Mhz
- **F2 (Caps Shift+Symbol Shift+2 en gomaDOS+)**: Modo turbo 2Mhz
- **F3 (Caps Shift+Symbol Shift+3 en gomaDOS+)**: Modo turbo 4Mhz
- **F4 (Caps Shift+Symbol Shift+4 en gomaDOS+)**: Modo turbo 8Mhz

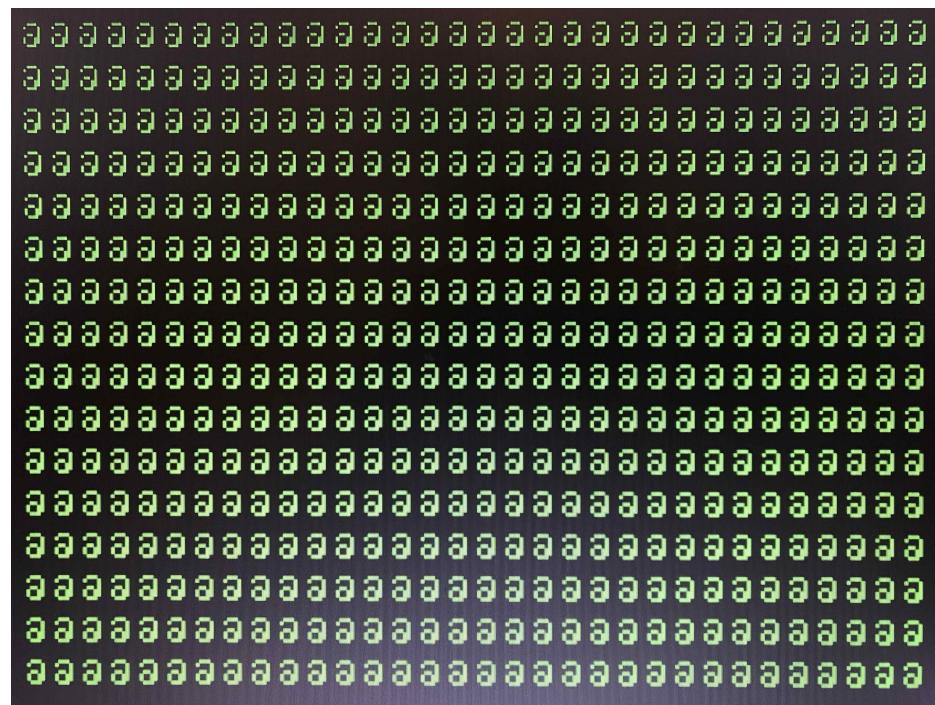
El teclado está mapeado en inglés, según el siguiente esquema:



Guía básica



Tras iniciar el core, en algunos casos, puede suceder que se muestre una pantalla llena de @. Basta con retirar e insertar, o simplemente insertar, la tarjeta microSD, para que empiece a funcionar.



Una vez iniciado, pulsar **Mayús+F10**, o bien escribir ***MENU** y **Enter**, para mostrar el menú desde el que se pueden cargar los programas de Atom Software Archive de la tarjeta.

Commodore 64

Commodore 64 (C64, CBM 64/CBM64, C=64,C-64, VIC-641) es una computadora doméstica de 8 bits desarrollada por Commodore International.

El core para ZX DOS+ está siendo desarrollado por Neuro.

Formato de Tarjeta microSD

Se puede utilizar una tarjeta microSD con la primera partición en formato FAT16 o FAT32. Es posible cargar desde la misma tanto archivos de imagen de disco ([D64](#)) como ficheros de cinta ([TAP](#)).

Si no estuviera ya, [instalar el core de Commodore 64](#) en el ZX DOS+.

Teclado

Teclas especiales y botones

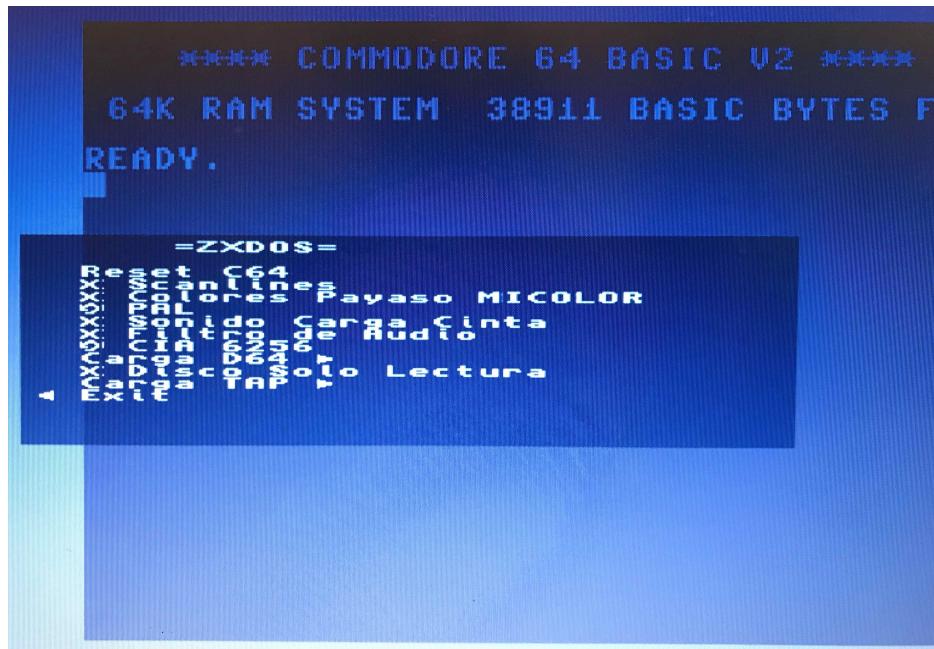
Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado [Commodore 64](#). Véase el [apartado dedicado a los modos de teclado](#) de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado [PC XT](#).

Durante la ejecución del core:

- **F9 Caps Shift+Symbol Shift+9** en gomaDOS+): Reproducir un archivo TAP de cinta
- **F12 (Caps Shift+Symbol Shift+W** en gomaDOS+): Muestra menú de opciones
- **Bloq. Despl. (Caps Shift+Symbol Shift+G** en gomaDOS+): cambia de modo vídeo RGB a VGA y viceversa.
- **Esc (Caps Shift+Espacio** en gomaDOS+): RUN/STOP (**Mayús+RUN/STOP**: Carga desde cinta)

Guía básica

Tras pulsar F12 (Caps Shift+Symbol Shift+W en gomaDOS+), aparece el menú de opciones.



Desde dicho menú se puede

- Hacer reset del core
- Activar o desactivar la simulación de línea de exploración (Scanlines)
- Cambiar la paleta de color
- Cambiar entre modo vídeo RGB y modo VGA
- Activar o desactivar el sonido de carga de cinta
- Encender o apagar un filtro de audio
- Cargar imagen de disco D64
- Cargar fichero de cinta TAP

Para cargar desde un disco, habitualmente, se ha de escribir `LOAD "*",8,1` y pulsar `Enter`. Una vez aparezca `READY` en la pantalla, escribir `RUN` y pulsar `Enter` para ejecutar el programa.

Si el disco tuviera varios programas para ejecutar, escribir `LOAD "$"` y pulsar `Enter`. A continuación, escribir `LIST`, y pulsar `Enter`, para ver una lista con los archivos dentro del disco. Ahora, para cargar el archivo deseado, escribir `LOAD "<nombre>",8` (donde `<nombre>` es el nombre del archivo a cargar) y pulsar `Enter`. Una vez aparezca `READY` en la pantalla, escribir `RUN` y pulsar `Enter` para ejecutar el programa. Si esto no funcionase, probar con el comando `LOAD "<nombre>",8,1`.

Para cargar desde una cinta, seleccionar la opción "Cargar fichero de cinta TAP" del menú de opciones. A continuación navegar por la tarjeta microSD y elegir el archivo de cinta a cargar, pulsar `ENTER` y cerrar el menú de opciones. Entonces escribir `LOAD` y pulsar `Enter`, o bien pulsar `Mayús+Esc` (`Mayús+RUN/STOP`). Finalmente, tras pulsar `F9` (`Caps Shift+Symbol Shift+9` en gomaDOS+) comenzará la reproducción del archivo de cinta (se puede usar la opción de activar el sonido de carga de la cinta del menú si se desea). Una vez finalizada la carga, escribir `RUN` y pulsar `ENTER` si fuese necesario.



En este core el puerto 1 del joystick se corresponde con el puerto de la derecha, mirando de frente al ZX DOS+, y el puerto 2 se corresponde con el puerto de la izquierda. Esto es al contrario de lo que sucede en la mayoría de otros cores.

Phoenix

Core del videojuego arcade de estilo matamarcianos diseñado por la empresa Amstar Electronics.

Algunas de sus características son:

- Dos modos de vídeo seleccionables: RGB/PAL60Hz y VGA 60Hz
- Simulación de línea de exploración (Scanlines) en VGA
- Comutación opcional para el giro de 90º en las direcciones de los controles

Formato de Tarjeta microSD

Este core no utiliza la tarjeta microSD.

Teclado

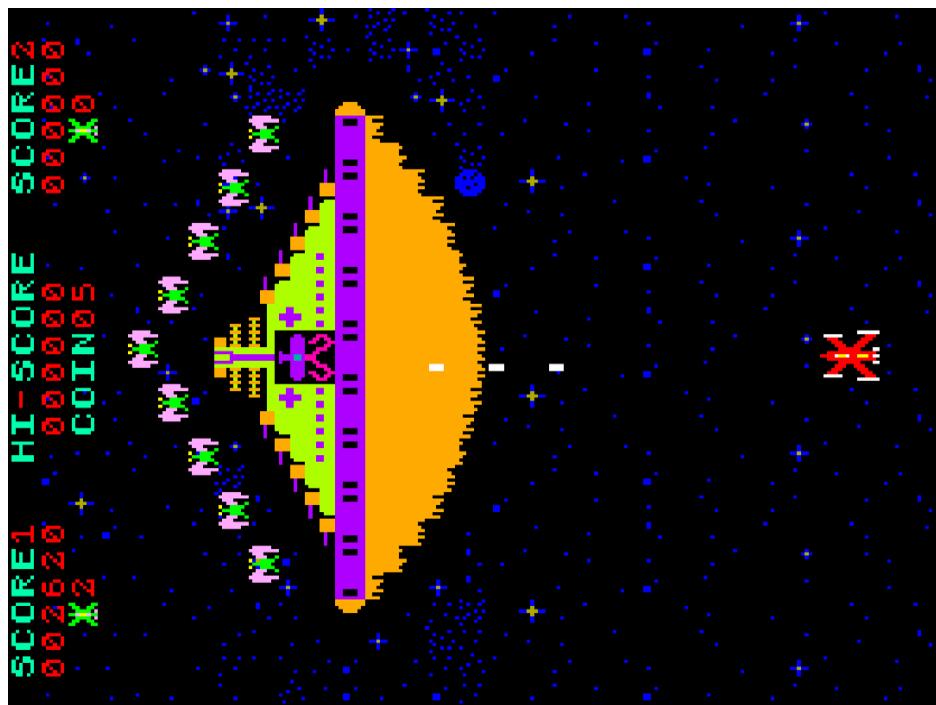
Teclas especiales y botones

Durante la ejecución del core:

- **Q** y **A** o **Cursor Izquierdo** y **Cursor Derecho** (o un joystick): Control de movimiento
- **Z** o **X** **Tecla Windows Izquierda** y **Espacio** (o botones 1 y 2 del joystick): Disparos 1 y 2, así como inserción de moneda y botón **Start**
- **F2** (**Caps Shift+Symbol Shift+2** en gomaDOS+): Cambia el modo de vídeo entre RGB y VGA
- **-** (del teclado numérico): Activa o desactiva la simulación de línea de exploración (Scanlines)
- **Tab** (**Caps Shift+Enter** en gomaDOS+, en modo de teclado **PC XT**): Activa o desactiva giro de 90º en las direcciones de los controles

Guía básica

Por defecto se inicia con los controles normales, para el uso de pantallas verticales.



Si se tiene la pantalla en horizontal (lo más habitual), la imagen se ve de lado, pero para ayudar en el control, y que sea más natural y acorde con lo que se ve, pulsando **Tab** se consigue que las direcciones arriba-abajo estén intercambiadas con izquierda-derecha. Afecta por igual al joystick y al teclado.

Pong

Pong [fue un videojuego](#) de la primera generación de videoconsolas publicado por Atari.

Algunas las características del core son:

- Dos modos de vídeo seleccionables: RGB/PAL60Hz y VGA 60Hz
- 7 tipos de juego
- Soporte 2 o 4 jugadores
- Compatible con joystick(s), teclado, ratón y codificadores (encoders) rotatorios (ver [aquí](#) para más información)
- Varios modos de color

Formato de Tarjeta microSD

Este core no utiliza la tarjeta microSD.

Teclado

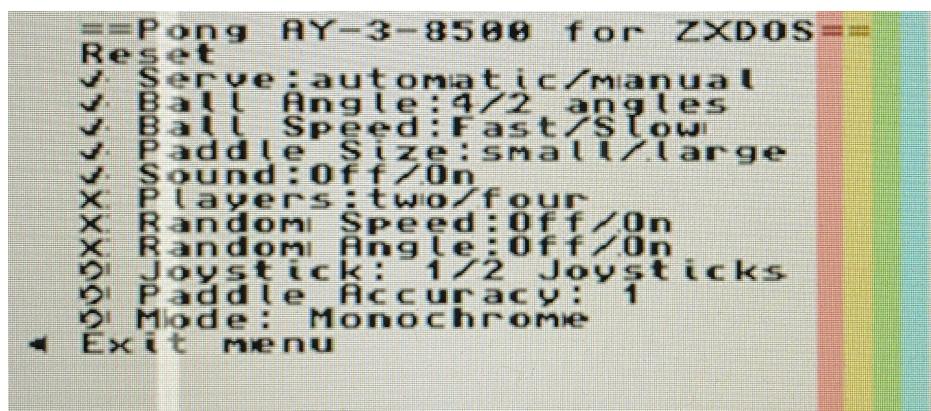
Teclas especiales y botones

Durante la ejecución del core:

- **Esc** o botón 2 del joystick ([Caps Shift+Espacio](#) en gomaDOS+, en modo de teclado **PC XT**): Mostrar u ocultar el menú de configuración
- **Ctrl+Alt+Backspace** ([Caps Shift+Symbol Shift+B](#) en gomaDOS+, en modo de teclado **ZX Spectrum**): Hard reset. Backspace es la tecla de borrar hacia atrás, encima del enter
- **Bloq. Despl.** ([Caps Shift+Symbol Shift+G](#) en gomaDOS+, en modo de teclado **ZX Spectrum**): cambia de modo vídeo RGB a VGA y viceversa
- **F3** o **F12** ([Caps Shift+Symbol Shift+3](#) o [Caps Shift+Symbol Shift+W](#) en gomaDOS+): Reinicio del juego
- Número del **1** al **7**: Cambiar el tipo de juego
- Joystick 2 (derecha): Control de paleta de la derecha (Jugador 1).
- Joystick 1 (izquierda): Control de paleta de la izquierda (Jugador 2)
- **Cursor arriba** y **Cursor abajo** o **0** y **K**: Control de paleta de la derecha (Jugador 1 en modo 2 jugadores y jugador 3 en modo de 4 jugadores)
- **Q** y **A**: Control de paleta de la izquierda (Jugador 2 en modo 2 jugadores y jugador 4 en modo de 4 jugadores)
- **Z**, **M** o botón de joystick 1: Saque manual
- Teclas de cursor ([Caps Shift+5](#), [Caps Shift+6](#), [Caps Shift+7](#) y [Caps Shift+8](#) en gomaDOS+, en modo de teclado **PC XT**) y **Enter** para navegar por el menú

Guía básica

Pulsando **Esc** o el botón 2 del joystick (o **Caps Shift+Espacio** en gomaDOS+, en modo de teclado **PC XT**) se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**) y **Enter** para elegir y seleccionar opciones del menú.



En él se pueden activar o desactivar las siguientes opciones:

- Servicio automático o manual (Serve)
- Ángulo de la bola (Ball Angle)
- Velocidad de la bola (Ball Speed)
- Tamaño de las palas (Paddle Size)
- Sonido (Sound)
- Cuatro jugadores (Four players)
- Velocidad Aleatoria (Random Speed)
- Ángulo Aleatorio (Random Angle)
- Control por Joystick, teclado o ratón (Joystick)
- Precisión del movimiento (Paddle Accuracy)
- Modo de color (Mode)
- Salir del menú (Exit menu)

NES

Nintendo Entertainment System (también conocida como Nintendo NES o simplemente NES) es la [segunda consola de sobremesa de Nintendo](#).

La versión para ZXDOS+ ha sido creada por Nihirash, basándose en la [anterior para ZX-Uno](#) de DistWave y Quest.

Algunas de las características del core son:

- Filtro HQ2X que "despixeliza" la imagen
- Simulación de línea de exploración (Scanlines)
- Utiliza el reloj de la NES NTSC, por tanto funcionan correctamente las ROMs USA. Las ROMs PAL van más rápido de lo que deberían
- Permite cargar ROMS desde la microSD
- Necesita, al menos, un mando o joystick conectado y que tenga varios botones de disparo
- Sólo soporta salida VGA y utiliza timings poco rigurosos, por lo que es posible que de problemas en algunos monitores

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD, con la primera partición en formato FAT16, para almacenar los ficheros con las imágenes ROM (extensión **.NES**) de los juegos que se desee cargar. Los ficheros pueden estar en subdirectorios.

Si no estuviera ya, [instalar el core de NES](#) en el ZX DOS+.

Teclado

Teclas especiales y botones

Durante la ejecución del core:

- **Esc** o botón 2 del joystick (o **Caps Shift+Espacio** en gomaDOS+, en modo de teclado **PC XT**): Mostrar u ocultar el menú de configuración
- Teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**), y **Enter** para usar el menú
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+B** en gomaDOS+, en modo de teclado **ZX Spectrum**): Hard reset. Backspace es la tecla de borrar hacia atrás, encima del enter

Guía básica

Pulsando **Esc** se muestra el menú de configuración. Para desplazarse por el menú y activar o elegir alguna opción, se utilizan las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**), y **Enter**.



En él se pueden activar o desactivar las siguientes opciones:

- Reiniciar la NES (Reset NES)
- Activar o desactivar línea de exploración (Scanlines)
- Encender o apagar el filtro que suaviza la imagen (HQ2X Filter)
- Simular la pulsación del botón Select del mando 1 (P1 Select)
- Simular la pulsación del botón Start del mando 1 (P1 Start)
- Elegir una ROM para cargar desde la microSD (Load ROM)
- Salir del menú (Exit)

Computers Lynx

El [Lynx](#) fue un ordenador doméstico británico de 8 bits lanzado a principios de 1983 por la compañía Computers. Se lanzaron en total tres modelos, con 48kB, 96kB o 128kB de RAM.

La versión para ZX DOS+ tiene estas características:

- Modos 48kB y 96 kB
- ROM Scorpion opcional
- Carga usando la entrada de audio
- Soporte para vídeo RGB y VGA

Formato de tarjeta microSD

Este core no utiliza la tarjeta microSD.

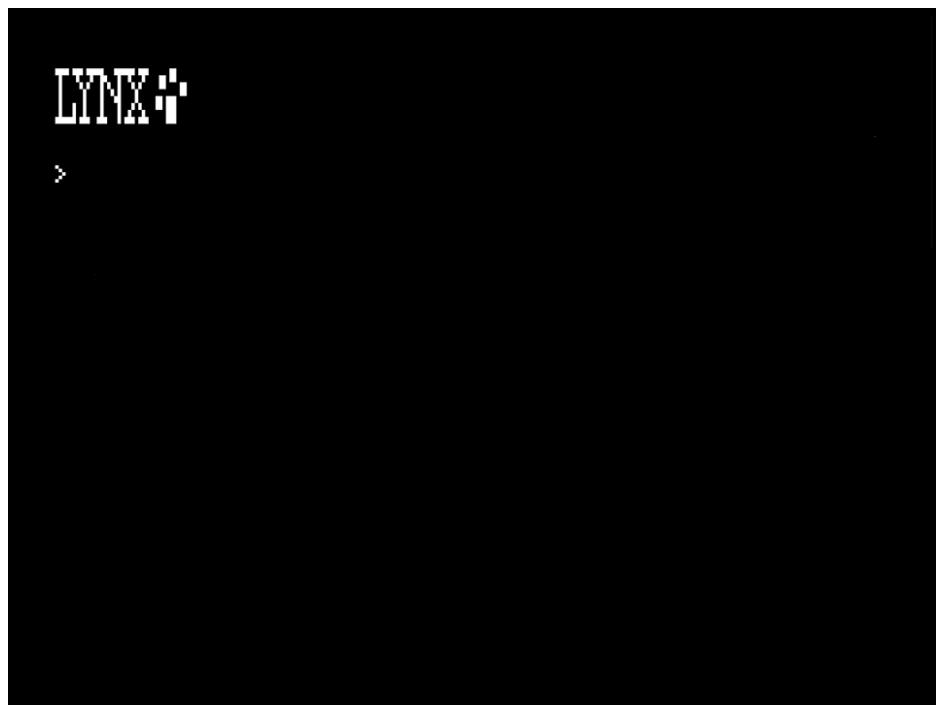
Teclado

Teclas especiales y botones

Durante la ejecución del core:

- **F6** (`Caps Shift+Symbol Shift+6` en gomaDOS+): Alterna entre el modo 48kB y el modo 96kB (por defecto)
- **F7** (`Caps Shift+Symbol Shift+7` en gomaDOS+): Alterna entre activar o desactivar la ROM Scorpio
- **F8** (`Caps Shift+Symbol Shift+8` en gomaDOS+): Activar o desactivar el tener en cuenta los bits 2 y 3 del puerto \$80 (señal CAS del banco 2), para que se vean bien los juegos de Level 9.
- **Bloq. Despl.** (`Caps Shift+Symbol Shift+G` en gomaDOS+): para cambiar entre modo RGB y VGA
- **Ctrl+Alt+Supr** (`Caps Shift+Symbol Shift+B` en gomaDOS+): Reset
- **Ctrl+Alt+Backspace** (`Caps Shift+Symbol Shift+B` en gomaDOS+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima del enter

Guía básica



Desde BASIC, se suele cargar desde una cinta (u [otro dispositivo externo de audio](#)) con la secuencia de comandos:

```
TAPE n  
LOAD "NOMBRE"
```

Donde `n` es un número (entre 1 y 5), según como se haya realizado la grabación, y `NOMBRE` es, obligatoriamente el nombre a cargar desde la cinta.

Si no se sabe el nombre a cargar, se puede averiguar con la misma secuencia de comandos, pero escribiendo `LOAD ""`.

Para ficheros binarios, se debe usar `MLOAD` en vez de `LOAD`.



El software Maxduino incorporando en [el miniduino](#) no tiene, por el momento, soporte para los archivos de cinta `TAP` de Lynx.

Es posible utilizar programas como [Lynx2Wav](#) con los ficheros `TAP` de cinta de Lynx. Los ficheros de audio obtenidos se pueden embeber a su vez dentro de ficheros `TSX` o `TZX` con herramientas como [MakeTSX](#) o [RetroConverter](#).

El script [lince](#) facilita todo este proceso, permitiendo crear directamente ficheros `TZX` compatibles con Maxduino desde ficheros `TAP` de Lynx.

Colecovision

Colecovision es una consola de videojuegos lanzada al mercado por la empresa Coleco.

La versión para ZXDOS+ está basada en la [versión para ZX-Uno](#) de Fabio Belavenuto.

Algunas de las características del core son:

- La ROM de la BIOS se carga desde la tarjeta microSD
- Soporta ROM multicartucho, que también se carga desde la microSD
- Sólo funciona en VGA ¿?

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD, con la primera partición en formato FAT16, para almacenar los ficheros con las imágenes ROM y otros archivos necesarios. Los archivos se pueden descargar desde la [web del proyecto original en GitHub](#).

Si no estuviera ya, [instalar el core de Colecovision](#) en el ZXDOS+.

Teclado

Teclas especiales y botones

Durante la ejecución del core:

- Cursor o **Q, A, E, R** o el joystick 1: Controles de dirección del jugador 1
- **Z** o el botón de joystick 1: Botón de disparo 1 del jugador 1
- **U, J, O, P** o el joystick 2: Controles de dirección del jugador 2
- **M** o el botón de joystick 2: Botón de disparo 1 del jugador 2
- **X** o el botón secundario de joystick 1: Botón de disparo 2 del jugador 1 y del jugador 2
- **0 a 9**: Botones del 0 al 9 del jugador 1 y el jugador 2
- **T**: Botón '*'
- **Y**: Botón '#'
- 'Esc' (o **Caps Shift+Espacio** en gomaDOS+, en modo de teclado **PC XT**): Soft Reset

Guía básica

Al iniciar, la ROM de la BIOS se carga desde la tarjeta microSD, así como la ROM multicartucho.



En el menú multicartucho, usar los controles de dirección para elegir la ROM a cargar, y luego el botón de disparo 1 para cargar la ROM elegida. Pulsando **Esc** (**Caps Shift+Espacio** en gomaDOS+, en modo de teclado **PC XT**) se reinicia el core y se vuelve a cargar el menú de selección de ROM.

Atari 2600

La [Atari 2600](#) es una videoconsola lanzada al mercado bajo el nombre de Atari VCS (Video Computer System).

La versión para ZX DOS+ está desarrollada por avlixa.

Algunas de las características del core son:

- Dos modos de vídeo seleccionables: RGB y VGA
- Compatible con joystick(s), teclado, ratón y codificadores (encoders) rotatorios (ver [aquí](#) para más información)

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD, con la primera partición en formato FAT16, para almacenar los ficheros con las imágenes ROM que se deseen cargar.

Si no estuviera ya, [instalar el core de Atari 2600](#) en el ZX DOS+.

Teclado

En gomaDOS+, se recomienda cambiar el modo de teclado a [Atari 800](#) ([Caps Shift+Symbol Shift+U](#) y luego [4](#)) o [PC XT](#) ([Caps Shift+Symbol Shift+U](#) y luego [9](#)).

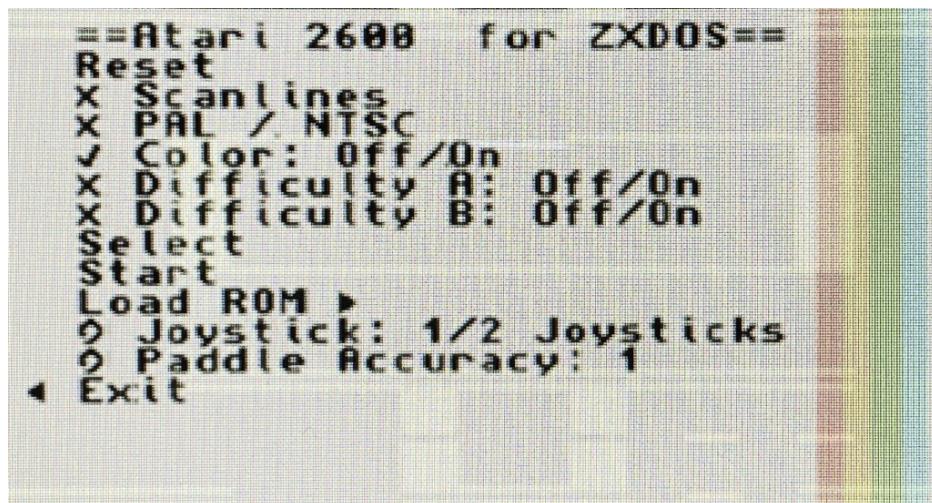
Teclas especiales y botones

Durante la ejecución del core:

- [W, A, S, D](#) o el joystick 1: Controles de dirección del jugador 1
- [F](#) o el botón de disparo del joystick 1: Disparo del jugador 1
- [I, J, K, L](#) o el joystick 2: Controles de dirección del jugador 2
- [H](#) o el botón de disparo del joystick 2: Disparo del jugador 2
- [Bloq. Despl.](#) ([Caps Shift+Symbol Shift+G](#) en gomaDOS+): para cambiar entre modo de vídeo RGB y VGA
- [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+B](#) en gomaDOS+): Hard reset.

Guía básica

Pulsando **Esc** o el botón 2 del joystick (o **Caps Shift+Espacio** en gomaDOS+, en modo de teclado **Atari800**) se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **Atari800**) y **Enter** para elegir y seleccionar en el menú.



En él se pueden activar, desactivar o configurar las siguientes opciones:

- Reiniciar el core (Reset)
- Línea de exploración (Scanlines)
- Modo RGB (PAL/NTSC)
- Tamaño de las palas (Paddle Size)
- Sonido (Sound)
- Color (Color)
- Dificultad A (Difficulty A)
- Dificultad B (Difficulty B)
- Select
- Start
- Cargar ROM (Load ROM)
- Joystick
- Precisión de paddle (Paddle Accuracy)
- Salir del menú (Exit)

Videopac

La [Philips Videopac](#), también conocida como Magnavox Odyssey², Philips Videopac G7000 o Philips Odyssey² es una videoconsola lanzada en 1978.

La versión para ZX DOS+ está desarrollada por avlixa, basada en el core de ZX DOS de yomboprime.

Algunas de las características del core son:

- Dos modos de vídeo seleccionables: RGB y VGA
- Necesita Joystick(s) para funcionar
- Modos de color incluyendo monocromo o fósforo verde/naranja
- Soporte para modificar el tipo de letra (charset) de la ROM de VDC
- Módulo de efectos de sonido y voz (The Voice)

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD, con la primera partición en formato FAT16, para almacenar los ficheros con las imágenes ROM que se deseen cargar.

Si no estuviera ya, [instalar el core de Videopac](#) en el ZX DOS+.

Teclado

En gomaDOS+, se recomienda cambiar el modo de teclado a **PC XT** (**Caps Shift+Symbol Shift+U** y luego **9**).

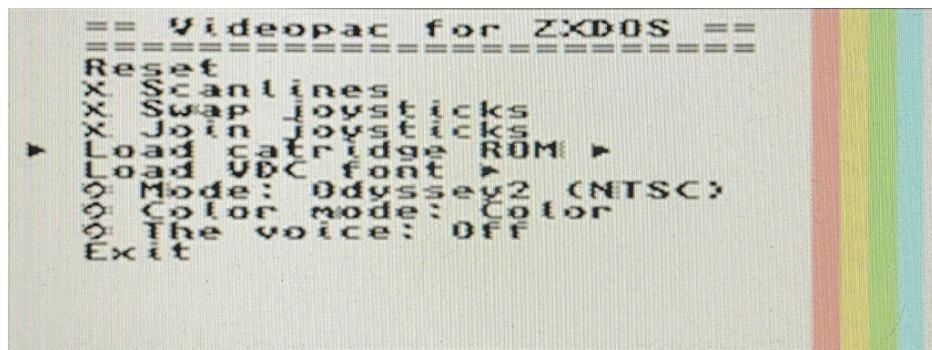
Teclas especiales y botones

Durante la ejecución del core:

- **F1** (**Caps Shift+Symbol Shift+5** en gomaDOS+): Carga una ROM de prueba.
- **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+): para cambiar entre modo de vídeo RGB y VGA
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+B** en gomaDOS+): Hard reset.
- Tras cargar una ROM, la mayoría de los juegos mostrarán un mensaje "SELECT GAME". Pulsar una tecla entre **0** y **9** o el botón del joystick para jugar.
- **Esc** o el botón 2 del joystick (o **Caps Shift+Espacio** en gomaDOS+) para mostrar u ocultar el menú.
- **W, A, S, D** o las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**) y luego **Enter** para elegir y seleccionar en el menú.

Guía básica

Pulsando **Esc** o el botón 2 del joystick (o **Caps Shift+Espacio** en gomaDOS+, en modo de teclado **PC XT**) se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**) y **Enter** para elegir y seleccionar en el menú.



En él se pueden activar, desactivar o configurar las siguientes opciones:

- Reiniciar el core (Reset)
- Línea de exploración (Scanlines)
- Intercambiar el orden de los joysticks (Swap Joysticks)
- Unir los joysticks (Join Joysticks). Útil para ROMs que alternan el joystick en las partidas
- Cargar ROM (Load Cartridge ROM)
- Cargar un tipo de letra alternativo (Load VDC Font)
- Modo de vídeo (PAL VideoPac/NTSC Odyssey2)
- Modo de color (Color Mode)
- Activar o desactivar el módulo sintetizador (The Voice)
- Salir del menú (Exit)



El sistema no tiene claramente definido qué joystick se corresponde a cada jugador, e incluso puede irse alternando para algunos juegos, así que puede ser necesario ir cambiando entre los dos, o bien (para partidas de un sólo jugador) usar la opción del menú que unifica los dos joysticks.



No suele haber información en pantalla de las opciones, así que es interesante consultar las instrucciones de cada juego (por ejemplo en [este enlace](#)) para saber qué hacer para jugar.



Si, al navegar por el directorio donde están las ROMS, no se ven todas, se puede dividir el contenido en varios subdirectorios con menos ficheros por directorio (por ejemplo, clasificando por letras) para solucionarlo.

Cambio del charset de la ROM de VDC

Es posible, para algunas ROM, cargar un archivo **CHR** con una fuente modificada, en vez de la original incluida en el chip Intel 8244/8245.

Estos archivos se pueden crear siguiendo las instrucciones y utilizando el editor disponible en el repositorio del proyecto, siguiendo [este enlace](#).

ZX81

El computador personal [Sinclair ZX81](#), lanzado por Sinclair Research en 1981, fue el de menor precio de la época.

La versión para ZXDOS+ ha sido creada por avlixa, basada en la página de Grant Searle's sobre el ZX80

Características:

- Se puede elegir entre ZX80 y ZX81 (ZX80 sólo con salida RGB por el momento)
- 16k/32k/48k RAM packs
- 8KB con addon CHR\$128/UDG (sin probar)
- QS CHRS (sin probar)
- CHROMA81
- Turbo en modo lento: NoWait, x2, x8
- Chip de sonido YM2149 (compatible con ZON X-81)
- Múltiples joysticks (Cursor, Sinclair, ZX81, ZXpand)
- Timings PAL y NTSC
- Turbo carga de ficheros .o y .p
- Carga de ROM alternativa
- Carga de programas a través de la entrada de audio

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta microSD, con la primera partición en formato FAT16 o FAT32, para almacenar los ficheros con archivos de cinta o ROM que se deseen cargar.

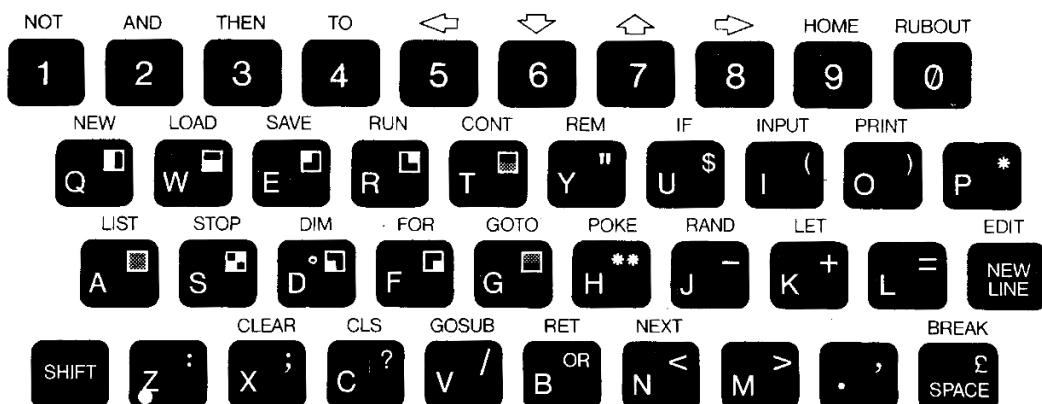
Se puede copiar, si se desea, un fichero [ZX8X.ROM](#) (disponible [the repositorio oficial](#) en el directorio [/zx81/roms](#): se trata de los ficheros ZX81 rom (8k) + ZX80 rom (4k) concatenados).

Si no estuviera ya, [instalar el core de ZX81](#) en el ZX DOS+.

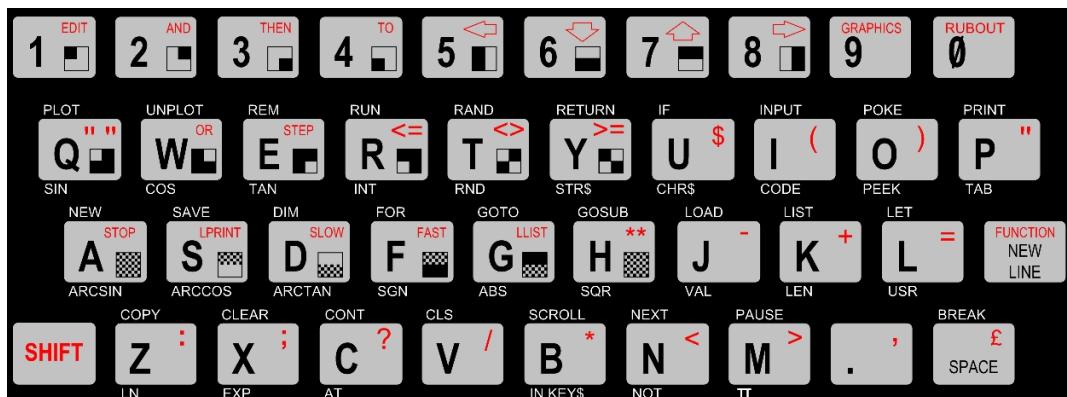
Teclado

El teclado no está mapeado a la distribución PS/2 estándar, y mantiene la distribución de la máquina original. Así, por ejemplo, para obtener " se ha de pulsar Mayús+P o para borrar, Mayús+0.

ZX80



ZX81



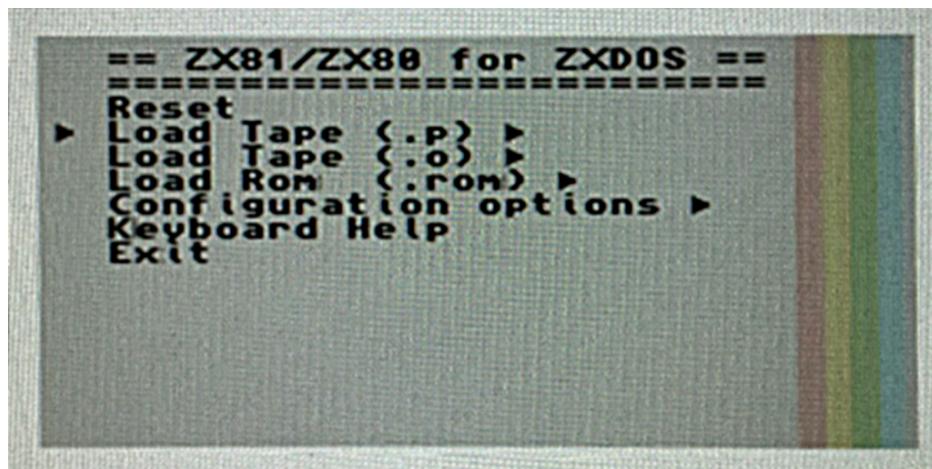
Teclas especiales y botones

Durante la ejecución del core:

- F1 (Caps Shift+Symbol Shift+1 en gomaDOS+): Habilitar o deshabilitar los caracteres alternativos
- F5 (Caps Shift+Symbol Shift+5 en gomaDOS+) o el botón 2 del joystick para mostrar u ocultar el menú.
- F9 (Caps Shift+Symbol Shift+9 en gomaDOS+): Deshabilita o habilita la salida MIC al conector de audio, porque algunos juegos generan un ruido molesto
- F10 (Caps Shift+Symbol Shift+0 en gomaDOS+): Habilita o deshabilita sonido de entrada por la salida de audio, para escuchar la cinta si se carga por audio
- Bloq. Despl. (Caps Shift+Symbol Shift+G en gomaDOS+): para cambiar entre modo de vídeo RGB y VGA
- Ctrl+Alt+Supr (Caps Shift+Symbol Shift+B en gomaDOS+): Reset
- Ctrl+Alt+Backspace (Caps Shift+Symbol Shift+B en gomaDOS+): Hard reset.

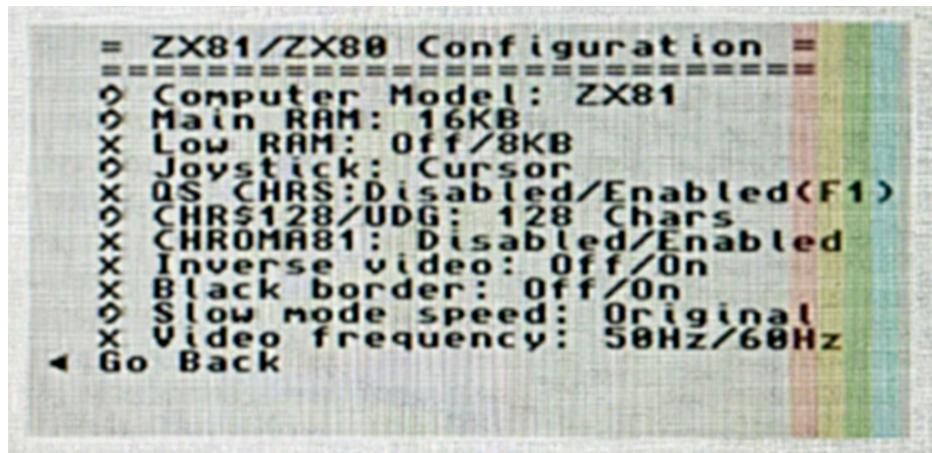
Guía básica

Pulsando **F5** (**Caps Shift+Symbol Shift+5** en gomaDOS+) o el botón 2 del joystick se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**) y **Enter** para elegir y seleccionar opciones del menú.



En él se pueden activar, desactivar o configurar las siguientes opciones:

- Reiniciar el core (Reset)
- Cargar un archivo de cinta (Load Tape)
- Cargar una ROM alternativa (Load ROM)
- Configurar opciones (Configuration Options)
- Salir del menú (Exit)



- Elegir el modelo de computador (Computer Model): ZX80/ZX81
- Tamaño de RAM (Main RAM): 16K/32K
- RAM baja (Low RAM): Off/8KB
- Joystick: Cursor/Sinclair/ZX81
- Habilitar o deshabilitar QS CHRS
- Configuración de CHR\$128/UDG: 128 chars/64 chars/Disabled
- Habilitar o deshabilitar Chroma81: Disabled/Enabled
- Vídeo inverso (Inverse Video): Off/On
- Borde negro (Black Border): Off/On
- Modo de velocidad (Slow mode speed): Original/No Wait/x2
- Frecuencia de vídeo (Video frequency): 50Hz/60Hz

Desde BASIC, se puede cargar desde un fichero cinta, que se haya seleccionado desde el menú, con el comando **LOAD ""** y pulsando **Enter**.



Algunos monitores dejan de reproducir la entrada de audio si se desactiva la señal de vídeo. Se recomienda conectar unos auriculares o altavoces externos para poder oír el sonido durante la carga de una cinta.

Se pueden cargar archivos **.P** con color y caracteres alternativos.

Para que funcione el coloreado, se debe activar CHROMA81 antes de la carga. Igualmente, para los caracteres alternativos, QS CHRS debería estar activo antes de cargar.



Las opciones recomendadas para la mayoría de los juegos son:

Main RAM: 16KB Low RAM: 8KB CHR\$128: 128 chars QS CHRS: enabled
CHROMA81: enabled

PC XT

El [IBM Personal Computer XT](#), fue el sucesor del IBM PC original. Estaba basado esencialmente en la misma arquitectura que el PC original, pero con algunas mejoras y se acabó convirtió en un estándar.

El core de ZX DOS+ es una implementación adaptada de [Next186](#).

Características del sistema implementado:

- Núcleo Next186 con CPU a 30 MHz (bus de 32 bit a 166.66Mhz)
- 64 MB DDR3 RAM (DDR3-1333 333.33Mhz)
- DSP a 40 MHz
- Soporte para OPL3 (corregido para detección correcta de adlib)
- Soporte para teclado y ratón PS/2
- Detección de BIOS en la tarjeta microSD en los primeros 64 sectores o en los últimos 16

Formato de Tarjeta microSD

Se debe de utilizar una tarjeta SDHC (por tanto de al menos 4GB), con la primera partición FAT16 y con MS-DOS (o similar) instalado. Esto se puede conseguir usando, por ejemplo, software de virtualización y conectando directamente el dispositivo de la tarjeta microSD como disco. En los foros de [ZX-Uno](#) y [ZXDOS+](#) se puede obtener más información y encontrar algunas imágenes de tarjeta microSD.

Una vez formateada y particionada (con la reserva al final), se debe grabar la imagen de BIOS [Next186_BIOS_zxdos_ddr3.COM](#) (disponible [en este enlace](#)) en algún lugar de los primeros 64 sectores, o bien en los últimos 16 sectores de la tarjeta microSD.



No se debe utilizar ninguna utilidad de reordenación de FAT ya que esto puede provocar que deje de arrancar correctamente DOS.



Como algunos de los directorios y ficheros de sistema de MSXDOS pueden tener el mismo nombre, no se recomienda utilizar la misma tarjeta para el [core de MSX](#) y el de PC XT.

Windows

Se puede utilizar el programa [HxD \(editor hexadecimal para Windows\)](#):

1. Abrir el programa como administrador y elegir la opción [Abrir disco…](#) en el menú [Herramientas](#)
2. Desmarcar la opción para abrir como solo lectura y seleccionar la tarjeta microSD
3. Abrir en otra ventana el fichero con la imagen de BIOS
4. Copiar el contenido de la imagen de BIOS, y pegarlo, reemplazando, en el final de la microSD, o bien, si hubiera espacio libre antes de la primera partición, en algún lugar dentro de los primeros 64 sectores (de 512 bytes cada uno)
5. Guardar los cambios en la SD

Linux

Seguir los siguientes pasos:

1. Tras identificar el disco de la tarjeta (en este ejemplo, `sdb`), invocar `fdisk` para ver el tamaño

```
fdisk -l /dev/sdb
Disk /dev/sdb: 59.49 GiB, 63864569856 bytes, 124735488 sectors
Disk model: SD/MMC
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
(...)
```

2. Como los sectores son de 512 bytes, restar 16 (en este ejemplo, $124735488-16=124735472$) y usar esta información para inyectar la BIOS al final:

```
sudo dd if=Next186_BIOS_zxdos_ddr3.COM of=/dev/sdb bs=512 seek=124735472
```

En lugar del final de la tarjeta, si hay espacio libre antes de la primera partición, se puede utilizar el principio. Por ejemplo, para hacerlo a partir del sector 10:

```
sudo dd if=Next186_BIOS_zxdos_ddr3.COM of=/dev/sdb bs=512 seek=10
```

macOS

Pasos a seguir:

1. Tras identificar el disco de la tarjeta (en este ejemplo, `disk6`), invocar `fdisk` para ver el tamaño en sectores

```
sudo diskutil unmountDisk /dev/disk6
sudo fdisk /dev/rdisk6

Disk: /dev/rdisk6    geometry: 15566/255/63 [30535680 sectors]
(...)
Enter 'help' for information
fdisk: 1>q
```

2. Como los sectores son de 512 bytes, restar 16 (en este ejemplo, $30535680-16=30535664$) y usar esta información para inyectar la BIOS al final:

```
sudo diskutil unmountDisk /dev/disk6
sudo dd if=Next186_BIOS_zxdos_ddr3.COM of=/dev/rdisk6 bs=512 seek=30535664
```

En lugar del final de la tarjeta, si hay espacio libre antes de la primera partición, se puede utilizar el

principio, por ejemplo, para hacerlo a partir del sector 10:

```
sudo dd if=Next186_BIOS_zxdos_ddr3.COM of=/dev/rdisk6 bs=512 seek=10
```

Si no estuviera ya, [instalar el core de PC XT](#) en el ZXDOS+.

Teclado

En el caso de un gomaDOS+, para las pulsaciones del teclado (Spectrum) sean aproximadas a las equivalentes en un teclado de PC, se pueden seguir estos pasos:

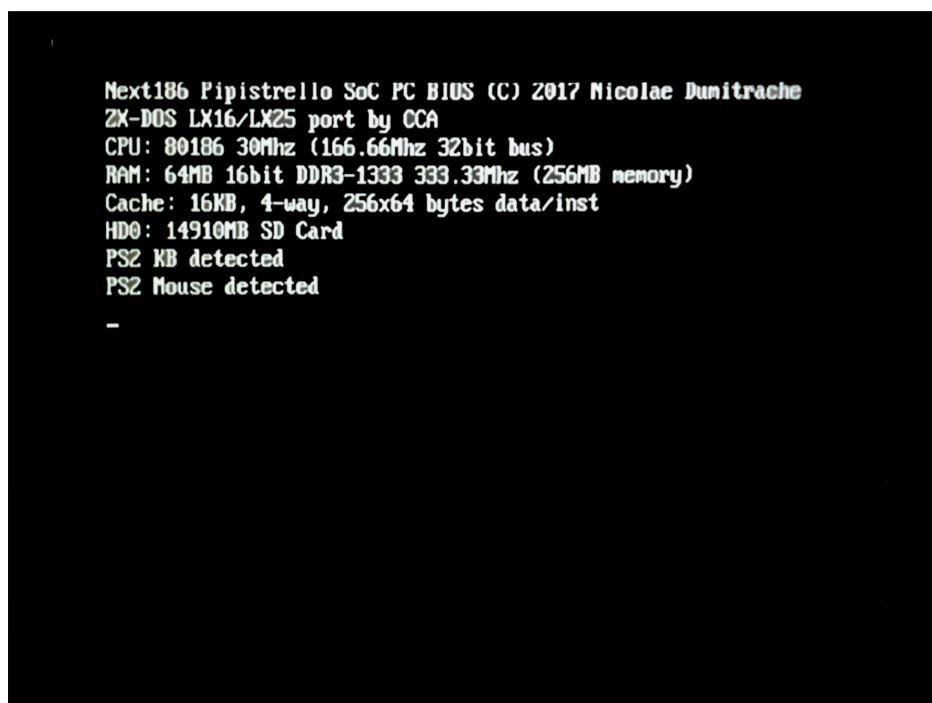
1. Iniciar el gomaDOS+
2. Cambiar el modo de teclado a **PC XT (Caps Shift+Symbol Shift+U** y luego **9**)
3. Reiniciar el gomaDOS+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+B**)
4. Rápidamente, pulsar **Caps Shift + 2** y elegir el core de PC XT en el arranque
5. Asegurarse de que la configuración de teclado de DOS es en un modo en inglés (**KEYB US**, o **KEYB UK,**)

Teclas especiales y botones

Durante la ejecución del core:

- **Ctrl+Alt+Supr (Caps Shift+Symbol Shift+N** en gomaDOS+): Soft reset.

Guía Básica



Chip-8

CHIP-8 es un lenguaje de programación interpretado, desarrollado por Joseph Weisbecker. Fue inicialmente usado en los microcomputadores de 8 bits COSMAC VIP y Telmac 1800 a mediados de 1970. CHIP-8 tiene un descendiente llamado SCHIP (Super Chip), presentado por Erik Bryntse.

El core de ZX DOS+ está basada en una implementación ya existente [para FPGA](#) de la máquina virtual de SuperChip.

Existen múltiples sitios como [CHIP-8 Archive](#) o [CHIP-8 de Matthew Mikolay](#) donde se puede obtener software para este tipo de máquinas.

Formato de Tarjeta microSD

Se puede utilizar una tarjeta con la primera partición en formato FAT16 o FAT32 para almacenar ficheros ROM en formato **BIN** o **CH8** para usar con el core.

Teclado

La máquina CHIP-8 utiliza un teclado hexadecimal como entrada. La asignación en el teclado es la siguiente:

Chip-8	PS/2
1 2 3 C	1 2 3 4
4 5 6 D	Q W E R
7 8 9 E	A S D F
A 0 B F	Z X C V

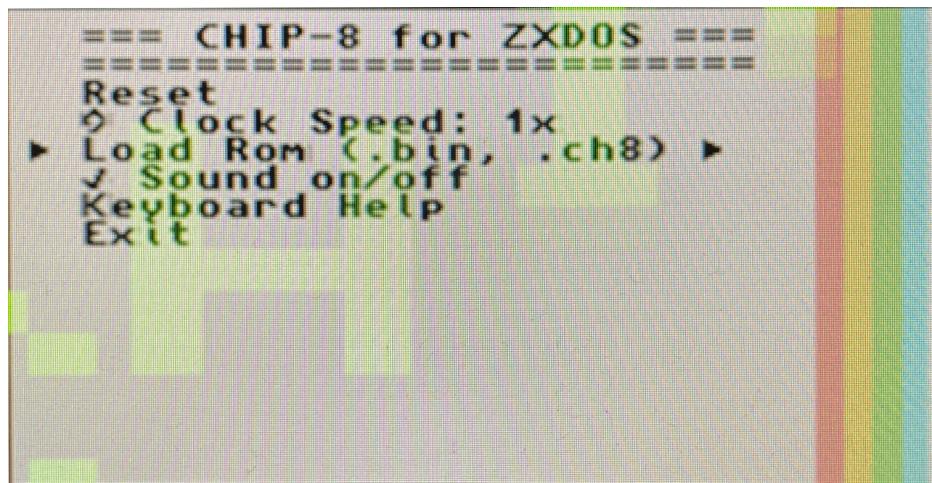
Teclas especiales y botones

Durante la ejecución del core:

- **Esc** (o **Caps Shift+Espacio** en gomaDOS+) para mostrar u ocultar el menú.
- **F11** (**Caps Shift+Symbol Shift+Q** en gomaDOS+): Hard Reset
- **F12** (**Caps Shift+Symbol Shift+W** en gomaDOS+): Reset

Guía Básica

Pulsando **Esc** (o **Caps Shift+Espacio** en gomaDOS+, en modo de teclado **PC XT**) se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en gomaDOS+, en modo de teclado **PC XT**) y **Enter** para elegir y seleccionar las distintas opciones.



En él se pueden activar, desactivar o configurar las siguientes opciones:

- Reiniciar el core (Reset)
- Cambiar la velocidad de reloj del core (Clock Speed)
- Cargar un archivo de ROM desde la tarjeta microSD (Load Rom)
- Activar o desactivar el sonido (Sound On/Off)
- Ayuda sobre el uso del teclado (Keyboard Help)
- Salir del menú (Exit)

Oric Atmos (Kyp)

El [Oric Atmos](#) era un ordenador doméstico fabricado por Oric Products International Ltd , sucesor del Oric 1.

La versión para ZX DOS+ está basada en la versión de Rampa para MiST, Mistica y SiDi, y tiene las siguientes características:

- Salida de vídeo compuesto o VGA (Scandoubler)
- 64KB de ram y 16KB de rom. La ROM oculta la parte alta de la memoria, pero se puede acceder a todos los bancos de memoria mediante ensamblador
- ROM seleccionable entre la versión original (1.1), y la versión la 1.22, ROM alternativa que corrige errores
- Las direcciones del joystick están mapeadas a las teclas del cursor, y los dos botones de disparo a las teclas **Espacio** y **S**.
- Carga de programas a través de la entrada de audio
- Grabación de programas a través de la salida de audio

Para más información consultar el [foro RetroWiki](#).

Formato de Tarjeta microSD

Este core no utiliza la tarjeta microSD.

Teclado

La disposición de las teclas en el teclado de PC intenta ser lo más parecida a la que tiene un Oric Atmos real.

La distribución del teclado está en inglés (así, por ejemplo, la tecla **-** se corresponde con **,**, al lado del número **0**).

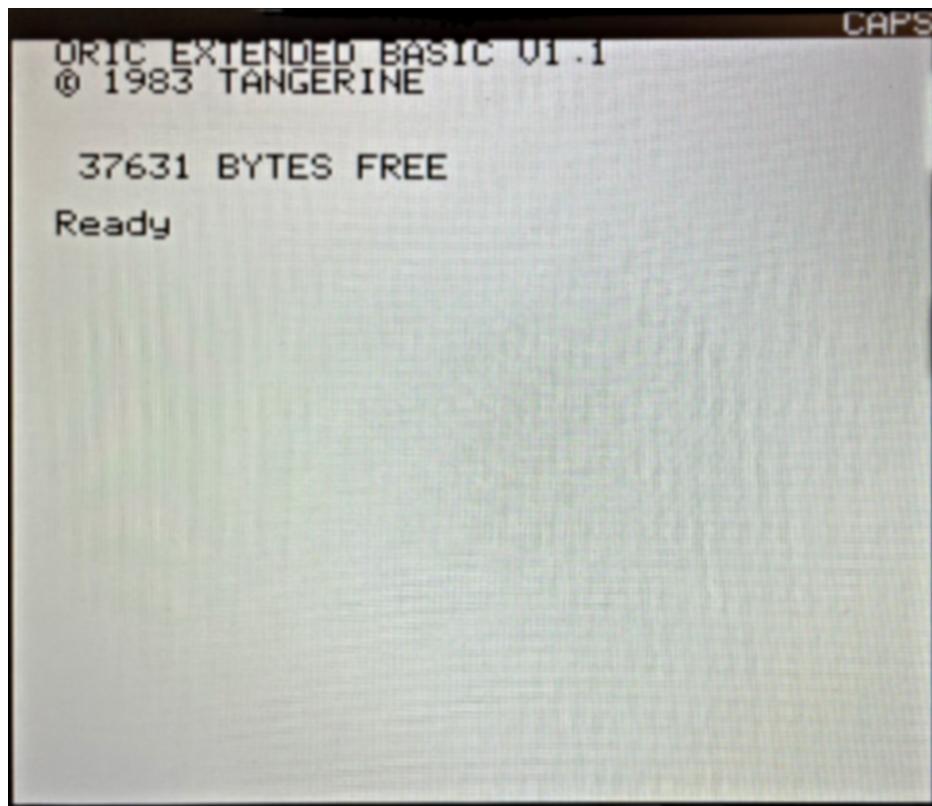


Teclas especiales y botones

Durante la ejecución del core:

- **F1** (**Caps Shift+Symbol Shift+1** en gomaDOS+): Seleccionar la ROM 1.1
- **F2** (**Caps Shift+Symbol Shift+2** en gomaDOS+): Seleccionar la ROM 1.22
- **F5** (**Caps Shift+Symbol Shift+5** en gomaDOS+): NMI
- **F10** (**Caps Shift+Symbol Shift+0** en gomaDOS+): Habilita la salida de audio al ejecutar **CSAVE**
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+B** en gomaDOS+) o **F11** (**Caps Shift+Symbol Shift+Q** en gomaDOS+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**
- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+N** en gomaDOS+) o **F12** (**Caps Shift+Symbol Shift+W** en gomaDOS+): Soft reset
- **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+): para cambiar entre modo de vídeo compuesto y VGA
- **Alt Gr.**: Tecla **FUNC**
- **Esc** (**Caps Shift+Espacio** en gomaDOS+): Tecla **ESC**
- **Supr**: Tecla **DEL**
- **Ctrl**: Tecla **CTRL**
- **Mayus.**: Tecla **SHIFT**
- **Windows Izq., Alt Izq, Windows Der., Menú**: Teclas de cursor **Izquierda, Abajo, Arriba, Derecha**

Guía básica



Desde BASIC, se puede cargar desde una cinta (u [otro dispositivo externo de audio](#)) usando el comando **CLOAD ""** y [Enter](#). A continuación, aparecerá en pantalla el mensaje

Searching...

Cuando se detecte un programa en la carga, el texto cambiará a

Loading...<nombre de programa>

En los [foros de RetroWiki](#) hay disponible un script de linux llamado [calorico.sh](#) que usa bash y PHP, junto con las utilidades [TSXphpclass](#) de [Natalia Pujol](#) para convertir ficheros [.TAP](#) de Oric en archivos [.TSX](#).



Para grabar un programa BASIC usando la salida de audio audio usando el comando **CSAVE "NOMBRE"** y [Enter](#). Si se desea poder escuchar durante la grabación, se debe haber habilitado previamente el sonido de grabación con la tecla **F10** ([Caps Shift+Symbol Shift+0](#) en gomaDOS+).



No debe estar activo el sonido de grabación (tecla **F10** o [Caps Shift+Symbol Shift+0](#) en gomaDOS+) durante una carga desde audio externo. El motivo es que la habilitación del sonido de grabación provoca una retroalimentación con un ruido alto y agudo que imposibilita la carga por audio.

Otro Hardware

Codificadores rotatorios

Los cores de Pong y Atari 2600 están también preparados para usar [codificadores rotatorios](#) (encoders), de tipo cuadratura, conectados al puerto de joystick. Está probado con codificadores de 600 ppr (pulsos por revolución), pero teóricamente deberían servir también de menor ppr, como de 400 o 300.

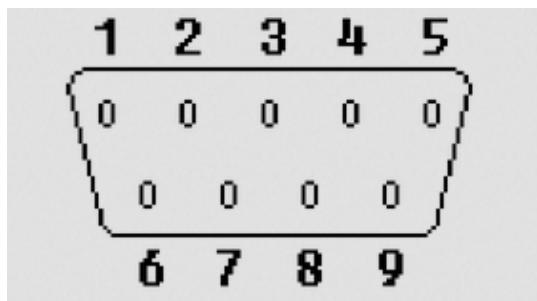
También se puede utilizar con [paddle](#) del tipo volante de Atari 2600, pero **no** con los paddle del tipo raqueta. Eso sí, la experiencia en este caso es pobre, ya que son de muy pocos ppr y se deben hacer varios giros completos. Si se utilizan, se recomienda ajustar la precisión(accuracy) al valor 8 para conseguir una velocidad aceptable.

Conexión

Tanto el ZX DOS+ como el gomaDOS+ tienen conectado el pin 5 del joystick a voltaje positivo, que se utiliza para alimentación, y el pin 8 como GND. Los codificadores utilizados deben admitir voltajes de 3,4v a 5v.

Los codificadores tienen 5 cables: Tierra (no conectada), Vcc (+), GND (~~0V~~ ó ~~-~~), A y B.

Las conexiones A y B se conectan en los pines 1 y 2 para el primer codificador, y 3 y 4 para el segundo. Así se pueden tener hasta 4 codificadores conectados entre los 2 puertos de joystick.



De esta manera, las conexiones serían:

1. Línea A codificador 1
2. Línea B codificador 1
3. Línea A codificador 2
4. Línea B codificador 2
5. Vcc(+)
6. Disparo 1
7. NC
8. GND
9. Disparo 2

Configuración del Core de Pong

La forma de elegir la configuración es la siguiente:

- Para 1 o 2 codificadores en el puerto 2 de joystick, seleccionar la opción **1/2 Paddle in J2**.
- Para 2 codificadores, uno en cada puerto de joystick, seleccionar la opción **2/4 Paddle in J1&J2**. Esta opción también sirve para conectar 2 paddle tipo driving de Atari 2600.
- Para 4 codificadores, dos en cada puerto de joystick, seleccionar la opción **2/4 Paddle in J1&J2**.
- Para 1 o 2 codificadores en el puerto 2 de joystick junto con el ratón (en este caso los codificadores se asocian para los jugadores 2 y 4), seleccionar la opción **Mouse PS/2**.

Se recomienda no realizar la conexión hasta haber seleccionado la opción deseada, ya que los codificadores actúan sobre el botón arriba/abajo del joystick y por tanto no permiten manejar el menú. Otra opción sería añadir un interruptor on/off en el codificador que desabilite la alimentación y así no emita señal mientras se realiza la selección de menú.

Carga desde cinta

Para algunos cores como, por ejemplo, el de Spectrum o el de MSX, es posible cargar, igual que se hacía en las máquinas originales, desde un dispositivo externo de audio como un reproductor de cassette u otro que lo simule.

Normalmente, se debe utilizar un cable adecuado para la [entrada de sonido del ZX DOS+](#), concretamente, ha de ser un cable con un jack estéreo de 3,5 mm en un extremo dos salidas mono en el otro (una para cada canal de audio), conectando el extremo del canal mono derecho al dispositivo de reproducción de cintas (esto no es necesario en Miniduino, ya que este automáticamente utiliza sólo el canal derecho de sonido al reproducir).

Reproductor de cassette

El funcionamiento es exactamente igual que se hacía con los equipos originales:

1. Conectar el cable de audio
2. Ejecutar en el equipo o elegir la opción correspondiente a la carga desde cinta. Por ejemplo, en ZX Spectrum 48K, pulsando **J**, a continuación, dos veces, " ([Symbol Shift + P](#) en gomaDOS+) y luego **Enter** para ejecutar el clásico **LOAD "" + Enter**
3. Iniciar la reproducción de la cinta (es posible que haya que hacer distintos intentos ajustando el volumen del reproductor)

Ordenador

Según el sistema operativo (Windows, macOS, Linux) existen múltiples alternativas de programas que pueden, o bien reproducir directamente un archivo de cinta (**TAP**, **TZX**, **PZX**, etc.) y emitir el sonido por la salida de auriculares, o bien crear un fichero de sonido (**WAV**, **VOC**, **AU**, etc.) que se puede reproducir también con programas de música o sonido.

PlayTZX

Este programa para Windows, macOS o Linux, permite reproducir un fichero de cinta **TZX** a través de la salida de sonido del ordenador.

Se puede descargar el fichero binario (por ejemplo, para Windows desde [World of Spectrum Classic](#) y para Mac desde [este repositorio de GitHub](#)) o compilar el código fuente como se explica [a continuación](#).

1. Conectar el cable de audio entre la salida del ordenador y la entrada de audio del ZX DOS+ (recordar conectar únicamente el extremo del canal mono derecho al extremo del PC/Mac, etc.)
2. Ejecutar en el equipo o elegir la opción correspondiente a la carga desde cinta. Por ejemplo, en ZX Spectrum 48K pulsando **J** a continuación, dos veces, " ([Symbol Shift + P](#) en gomaDOS+) y luego **Enter** para ejecutar el clásico **LOAD "" + Enter**

3. Iniciar la reproducción de un fichero de cinta con el siguiente comando (es posible que haya que hacer distintos intentos ajustando el volumen de salida del equipo)

```
./playtzx <fichero de cinta>
```

Si todo va bien, se irá viendo en la consola los distintos bloques de carga de la cinta, mientras el sonido se produce y el core del ZX DOS+ carga el programa.



En Linux, el programa utiliza como salida el dispositivo `/dev/dsp`, así que, en versiones más modernas, es posible, por ejemplo, que haya que cargar módulos como `snd_pcm_oss` (en el caso de sistemas que utilicen ALSA), para que funcione correctamente.

Compilar código fuente (macOS o Linux)

Verificar que están instaladas las herramientas de desarrollo, incluyendo un compilador de C (`gcc`, `clang` herramientas de desarrollo de línea de comandos en Mac, etc.) y [GNU Autotools](#).

Descargar el código fuente [desde este repositorio](#), descomprimirlo y acceder en una consola al directorio, y ejecutar los comandos:

```
aclocal && autoconf && autoheader && automake --add-missing  
.configure  
make
```

Si todo se ha hecho correctamente, se habrá generado el fichero `playtzx` que se puede copiar a donde se desee en el disco duro y utilizar. Se puede borrar el directorio descomprimido donde se compiló.

Teléfono móvil, tableta, reproductor de sonido MP3, etc.

En general, existen muy pocas alternativas (o ninguna) alternativas de programas que puedan reproducir directamente un archivo de cinta en un dispositivo móvil, así que, normalmente, será necesario convertir a un fichero de audio el contenido de la cinta antes de intentar cargarla con uno de estos aparatos.

<https://zxtape.net> es una web, compatible con dispositivos móviles, con soporte para reproducir ficheros TAP y TZX

[ZX Tape Player](#) es una App para sistemas iOS y Android que es capaz de reproducir directamente a través de la salida de auriculares

[PlayZX](#) es otra App similar, pero sólo para sistemas Android.



Los dispositivos con salida de auriculares modernos, suelen estar pensados para manejar impedancias en el orden de unas pocas decenas de ohmios. Esto, a veces, puede ser insuficiente para la entrada del ZX DOS+.

En esos casos, se recomienda (si es posible) desactivar las limitaciones de volumen máximo de auriculares y/o utilizar un dispositivo amplificador de auriculares, que eleve la impedancia.

Los pasos a seguir en este caso son:

1. Conectar el cable de audio entre la salida del dispositivo móvil y la entrada de audio del ZX DOS+ (recordar conectar únicamente el extremo del canal mono derecho al extremo del PC/Mac, etc.)
2. Ejecutar en el equipo o elegir la opción correspondiente a la carga desde cinta. Por ejemplo, en ZX Spectrum 48K pulsando **J** a continuación, dos veces, " (Symbol Shift + P en gomaDOS+) y luego **Enter** para ejecutar el clásico **LOAD "" + Enter**
3. Iniciar la reproducción del fichero de audio (o del fichero de cinta en el caso de PlayZX). Es posible que haya que hacer distintos intentos ajustando el volumen del reproductor/amplificador.

Conversión a fichero de audio

A continuación se indican algunos de los muchos programas que existen para distintos sistemas operativos, y que pueden exportar ficheros de cinta a ficheros de audio.

[Tapir](#) es un programa con interfaz gráfica para Windows (pero que se puede usar también con Wine en Linux o Mac) que permite cargar ficheros **TZX** y **TAP** y exportarlos como **WAV** de audio

[tape2wav](#) de [Fuse Utilities](#) es una utilidad de línea de comandos que exporta desde ficheros **TZX** **PZX** y **TAP** a **WAV**. Disponibles para muchos sistemas operativos distintos.

[pxz2wav](#) en [PZX Tools](#) es otra utilidad de comandos que exporta desde ficheros **PZX** a **WAV**. Disponible como ejecutable para Windows, y con el código fuente disponible para compilar en otros sistemas.

[tsx2wav](#) en [TSXphpclass](#) es una utilidad en PHP y que sirve para exportar desde ficheros **TSX** a **WAV**.

[Lynx2Wav](#) es un programa que permite convertir ficheros **TAP** de Computers Lynx como **WAV** de audio.

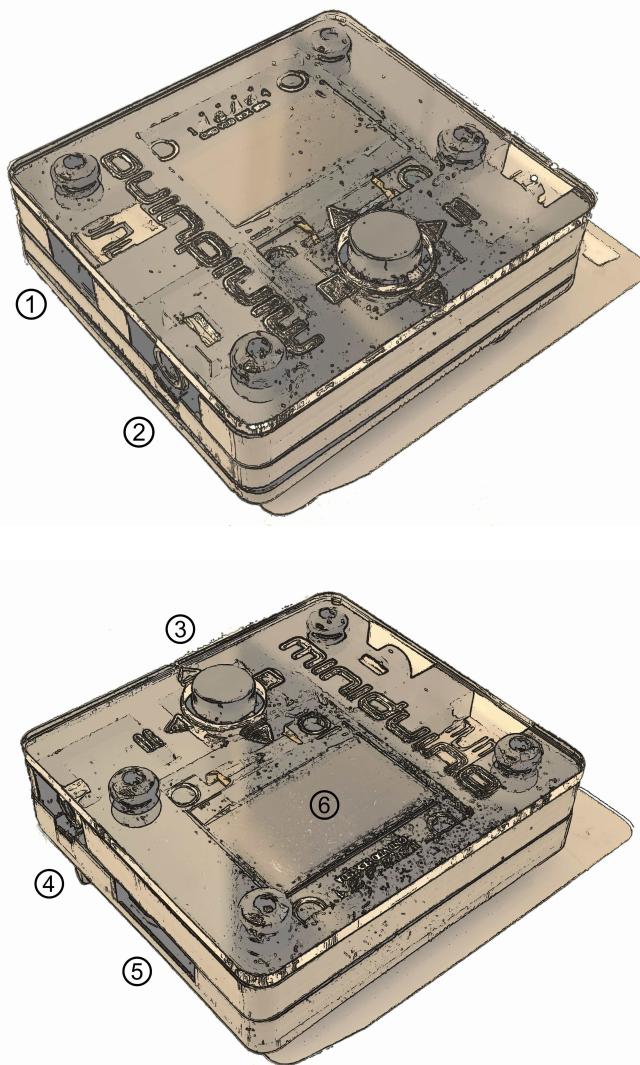
[2lynx2wav](#) también permite convertir ficheros **TAP** de Computers Lynx a ficheros **WAV** de audio.

Miniduino

Miniduino es un sistema reproductor de archivos de cinta, basado en un microcontrolador STM32F103C8T6 o ATMega38P, y que trae instalado de serie el firmware [Maxduino](#).

Maxduino permite la reproducción, de una forma similar a como se manejaban las cintas de [cassette](#) originales, de archivos digitales de cinta en múltiples formatos como [TAP](#) y [TZX](#) (ZX Spectrum), [O](#) (ZX80), [P](#) (ZX81), [CDT](#) (Amstrad CPC), [CAS](#)(MSX) [TSX](#) (MSX, Acorn, etc). También es posible reproducir como cinta archivos de sonido AY, para cargarlos en [SpecAY](#) desde ZX Spectrum.

Puertos y Botones



1	Alimentación
2	Salida de sonido
3	Botón de control
4	Control de motor
5	Ranura de tarjeta SD
6	Pantalla

Preparación

Se necesita una tarjeta SD para almacenar los archivos de cinta que se quiere reproducir. Se recomienda que no sea de alta velocidad (Clase 10 o superior) porque puede haber problemas durante la lectura, ni de alta capacidad (SDXC o superior).

La tarjeta debe estar formateada con la primera partición en formato FAT16 o FAT32.

Además de la tarjeta, recordar utilizar un cable adecuado para la [entrada de sonido](#), conectando el Miniduino.

Para los equipos que soporten control por motor, también se puede utilizar un cable con un jack de 2,6 mm.

En la primera partición de la tarjeta se han de copiar los archivos de cinta ([TAP](#), [TZX](#), [O](#), [P](#), [CAS](#), [TSX](#), etc), que se pueden organizar en carpetas o directorios.



El reproductor muestra las entradas de archivos y directorios en el orden de la tabla FAT interna, y no de manera alfabética. Si se desea ver esta información ordenada, se debe reorganizar la estructura de la tarjeta con una utilidad como [FAT Sorter](#) para Windows, [FATsort](#) para Linux y macOS, [YAFS](#), [SDSorter](#) u otros.

Uso

Una vez insertada la tarjeta SD con ficheros de datos, el Miniduino se enciende conectando el cable de alimentación USB incluido.



Si se pulsa directamente el botón de control, se accede al menú de opciones que permite modificar lo siguiente:

- Velocidad (Baud Rate): Para ajustar la velocidad turbo en bloques 4B en archivos de MSX ([CAS](#) y [TSX](#))
- Control de motor (Motor Ctrl): Para indicar que se ha conectado un cable de control remoto a un equipo que lo soporte (Amstrad, CPC, MSX, etc.)
- Conversión (TSXCzxpUEFWs): Activa la carga turbo para los archivos [.CAS](#) y [.TSX](#), cambiar la polaridad de la señal de audio de los archivos para Spectrum y Amstrad CPC y/o cambiar la paridad en los archivos [.UEF](#) de Acorn Electron y BBC Micro
- Saltar bloques (Skip BLK): Para deshabilitar (Skip ON) o habilitar la pausa automática al encontrar bloques del tipo 2A

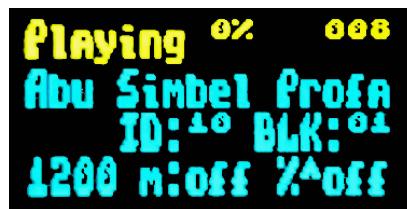
Estando fuera del menú de opciones, el botón de control se utiliza como una palanca de control de cuatro direcciones, que se comporta de dos maneras distintas, según esté la reproducción detenida (Stop) o en pausa (Pause).



Con la reproducción detenida (navegación por archivos y directorios):

- Arriba y abajo permiten desplazarse por la lista actual de ficheros y directorios
- Izquierda (Stop) retrocede un nivel en el árbol de directorios
- Derecha (Play/Pause) accede al contenido de un directorio o, si lo que hay seleccionado es un archivo, intenta reproducirlo

Una vez un archivo está en reproducción, el botón izquierda (Stop), la detiene, y el botón derecho (Play/Pause) la pone en pausa.



Con la reproducción en pausa (navegación por bloques de cinta):

- Arriba y abajo permiten desplazarse por los bloques del archivo de cinta que ya hayan sido reproducidos (útil para juegos multicarga, para volver a cargar un bloque de un nivel, por ejemplo)
- Izquierda (Stop) cancela la reproducción y entra de nuevo en el modo de navegación de archivos y directorios
- Derecha (Play/Pause) reanuda la reproducción en el bloque seleccionado
- Presionar directamente el botón de control permite activar o desactivar el modo turbo para MSX



Para obtener información mucho más detallada, se puede consultar el manual oficial del firmware Maxduino, disponible en el [repositorio oficial](#).

Creación de ficheros TZX o TSX desde otros formatos

Existen algunos formatos de cinta (Commodore, Computers Lynx, etc.) que, por el momento, no están directamente soportados por Maxduino. No obstante, existen algunos programas que pueden permitir, con mayor o menor éxito, transformar ficheros de cinta desde [formato de audio](#) a [TSX](#) o [TZX](#), y así poder utilizarlos con Miniduino.

MakeTSX

Para usar [MakeTSX](#) de NataliaPC y crear un fichero [TSX](#) con audio embebido, se ha de usar un comando como el siguiente:

```
...MakeTSX -b15 -wav fichero_audio.wav -tsx fichero_nuevo.tsx
```

RetroConverter

Para utilizar [RetroConverter](#) de Jorge Fuertes, para crear un fichero [TZX](#) con audio embebido, usar un comando como:

```
...retroconv fichero_audio.wav fichero_nuevo.tzx
```

Actualización de firmware Maxduino

El firmware Maxduino es actualizado y mejorado periódicamente. Se puede hacer un seguimiento de los cambios y mejoras en el [foro Va de Retro](#) o bien en la [página del proyecto en GitHub](#). Para poder aprovechar estas mejoras se ha de actualizar la memoria flash del Miniduino con la versión de firmware correspondiente.

Existen dos modelos de Miniduino, uno basado en el microcontrolador STM32, y otro basado en ATmega328P.

Modelo STM32

Preparación del entorno

Para poder instalar el firmware, se ha de hacer desde un ordenador (Windows, Mac, Linux) con el entorno [Arduino IDE](#).

Una vez instalado, se ha de añadir la biblioteca de software SDFat (1.1.4) eligiendo la opción de menú Programa → incluir librería → administrar bibliotecas

También se ha de añadir el soporte para el microcontrolador del Miniduino. Esto se hace en dos pasos:

Primero añadiendo soporte para procesadores ARXM Cortex M3 en el menú Herramientas → placa → gestor de tarjetas Instalar "Arduino SAM boards (Cortex-M3)"

A continuación se ha de añadir el soporte para microcontroladores STM32, descargando el fichero disponible en [este enlace](#).

Descomprimir el contenido en la carpeta del usuario actual en

```
...Arduino/hardware/Arduino_STM32
```

En Windows instalar el controlador USB ejecutando con privilegios elevados:

```
...\\drivers\\win\\install_drivers.bat
```

En Linux instalar con privilegios de root las reglas de [udev](#) necesarias:

```
...tools/linux/install.sh
```

En macOS, si no apareciese el Miniduino como dispositivo USB en Arduino IDE al conectarlo, puede que sea necesario instalar [libusb](#).

Finalmente, en el caso de Mac o Linux, el fichero `maple_upload` dentro de `Arduino_STM32` tiene que modificarse con un editor de texto. Estas líneas no funcionan bien:

```
if [ $# -eq 5 ]; then
    dfuse_addr="--dfuse-address $5"
else
    dfuse_addr=""
fi
```

Y se tienen que cambiar por

```
dfuse_addr=""
```

Actualización

Una vez preparado el entorno, descargar la versión deseada del proyecto desde el [repositorio oficial en GitHub](#)



El reproductor Miniduino con microcontrolador STM32 sólo está soportado a partir de la versión 1.65

Cargar el fichero del proyecto en Arduino IDE (por ejemplo `MaxDuino_v1.69.ino`).

Verificar en el fichero `userSTM32Config.h` que están comentadas todas las entradas de logo excepto la de Miniduino, y si no, cambiarlas.

```
...
#ifndef tanque4
#ifndef tanque1
#ifndef dostenques
#ifndef cablemax
#ifndef sony
#define miniduino
...
...
```

Conectar el Miniduino al equipo usando el cable USB e identificar el puerto asignado, normalmente con un nombre del tipo "Maple Mini" (por ejemplo: COM5 Maple Mini)

Configurar en el menú "Herramientas" las opciones

```
Placa: Generic STM32F103C Series
Variant: STM32F103C8 (20k RAM, 64k Flash)
Upload Method: STM32duino bootloader
CPU Speed: 72Mhz (Normal)
Optimize: Smallest (default)
Puerto: <Puerto identificado antes>
```

Finalmente, pulsar el botón de carga del firmware y esperar unos segundos mientras se compila el proyecto y se carga en el dispositivo.

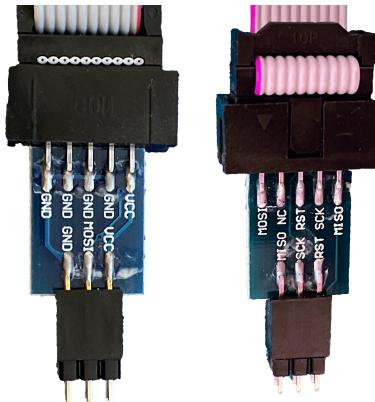
Si todo se ha hecho correctamente se verá cómo el Miniduino se reinicia y en la pantalla aparece la versión correspondiente de firmware.

Modelo ATmega328P

Preparación del entorno

Material necesario:

- Una [llave Allen](#) del tamaño adecuado para poder retirar la tapa
- Programador USBASP



Además, la instalación, se ha de hacer desde un ordenador (Windows, Mac, Linux) con el entorno [Arduino IDE](#).

Una vez instalado Arduino IDE, se ha de añadir la biblioteca de software SDFat (1.1.4) eligiendo la opción de menú Programa → incluir librería → administrar bibliotecas.

Actualización

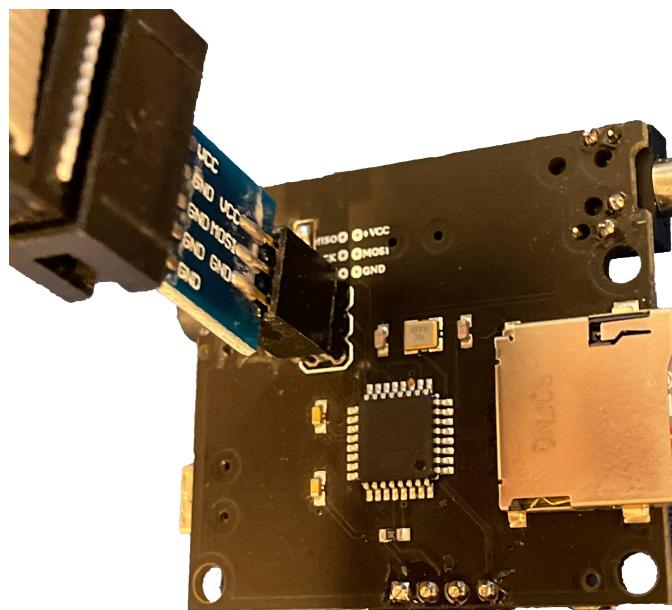
Una vez preparado el entorno, descargar la versión deseada del proyecto desde el [repositorio oficial en GitHub](#)

Cargar el fichero del proyecto en Arduino IDE (por ejemplo [MaxDuino_v1.69.ino](#)).

Verificar en el fichero [userconfig.h](#) que estén comentadas todas las entradas de logo excepto la de Miniduino, y si no, cambiarlas.

```
...
#ifndef tanque4
#ifndef tanque1
#ifndef dostenques
#ifndef cablemax
#ifndef sony
#define miniduino
...
...
```

Conectar el Miniduino al adaptador USBasp, asegurándose de que la serigrafía de los dos conectores se corresponde (ej: VCC con VCC, MOSI con MOSI, GND con GND, etc.), y conectar el adaptador USB al equipo.



Configurar en el menú "Herramientas" las opciones

Placa: Arduino Pro or Pro Mini

Procesador: ATmega328P (5V, 16 Mhz)

Programador: "USBasp"

Finalmente, mantener pulsada la tecla mayúsculas mientras se hace clic en el botón de carga del firmware y esperar unos segundos hasta que se compile el proyecto y se cargue en el dispositivo.

Si todo se ha hecho correctamente se verá cómo el Miniduino se reinicia y en la pantalla aparece la versión correspondiente de firmware.

Addon RTC+I²S+PIzero

Este complemento de hardware, compatible con ZX DOS+, GomaDOS+ y neptUNO, incorpora un [reloj en tiempo real \(RTC\)](#), un chip I²S UDA1334A y una cabecera de 40 pines donde se puede conectar una Raspberry Pi que, con el software adecuado, se puede utilizar como placa aceleradora del [core de ZX Spectrum Next](#).

I²S

[I²S](#), conocido también como Inter-IC Sound, Integrated Interchip Sound, o IIS, es un estándar que se utiliza para transmitir audio digital entre circuitos. En el caso del ZX DOS+, esto permite enviar el sonido de forma digital desde un core al addon, ofreciendo así una mejor calidad en la reproducción.

Cores con soporte

En el momento de escribirse estas líneas, hay los siguientes cores con soporte para I²S:

- [ZX Spectrum](#) (versiones más recientes del core EXP27 y 48K de Kyp)
- [MSX](#) (desde la versión 1.3)
- [ZX Spectrum Next](#) (versiones más recientes)



En el core de ZX Spectrum Next, debido a que se comparte la línea I²S con el audio de la Raspberry Pi, no es posible utilizar simultáneamente las dos, y se ha de conmutar una u otra usando la tecla **F12** (**Caps Shift+Symbol Shift+W** en gomaDOS+).

Raspberry Pi

NextPI

La distribución NextPI, basada en dietPI, incluye todo lo necesario para que la Raspberry Pi se comunique con el core de ZX Spectrum Next. La versión más reciente se puede [descargar aquí](#).

Una vez obtenida, se ha de instalar en una tarjeta microSD de, al menos, 1GB de tamaño, que luego se utilizará con la Raspberry Pi.



Es posible, usando [una solución como PINN](#), instalar varios sistemas operativos en la tarjeta, y así utilizar la Raspberry Pi para otros usos distintos cuando no se quiera tener como aceleradora.

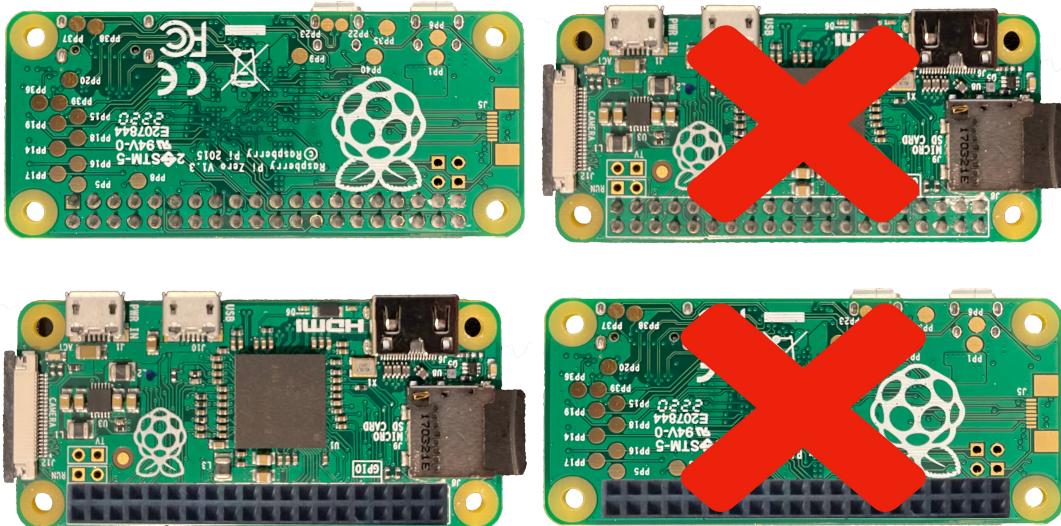


La distribución NextPI sólo está probada para usarse con Raspberry Pi Zero. No obstante, se ha podido comprobar que funciona, aparentemente, bien también con otros modelos (como Raspberry Pi 3, Raspberry Pi Zero W, etc.).

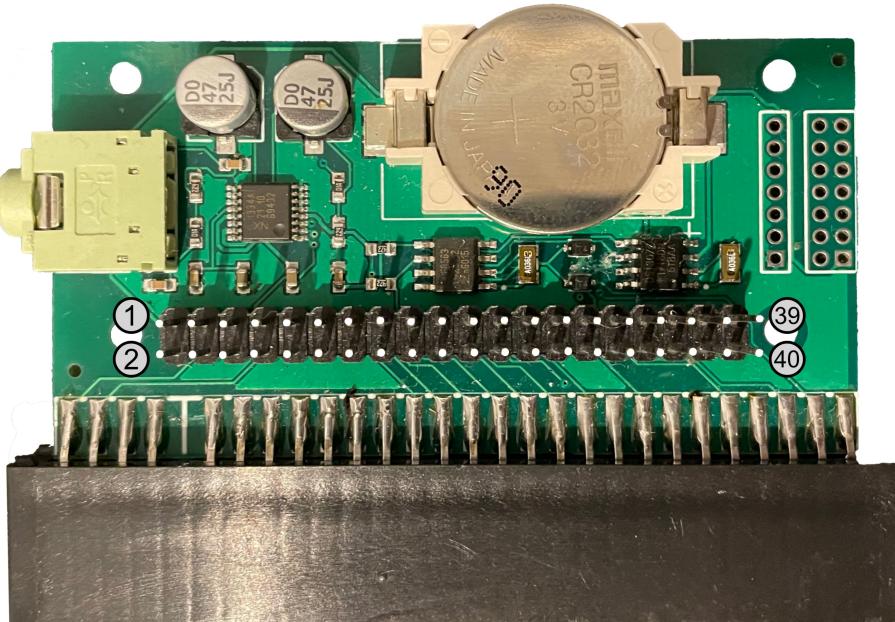
Conexión

A pesar de usar el mismo conector **la conexión con este addon no es compatible con las Raspberry PI Zero para ZX Spectrum Next**.

Como se puede ver en el esquema de más adelante, una Raspberry Pi Zero compatible con el addon se sitúa, entre otras cosas, con la parte de la placa donde están los puertos y el adaptador de microSD en la parte de abajo, mientras que la versión para ZX Spectrum Next, se encuentran en la parte de arriba.



Una conexión incorrecta de alguno de los pines (5V o 3,3V) con alimentación eléctrica a un punto equivocado puede provocar un daño irreversible y dejar una o más conexiones de la placa inservibles.



Para la versión de NextPI disponible al escribir estas líneas (0.99D RTM), no es necesario utilizar todas conexiones, siendo suficiente utilizar:

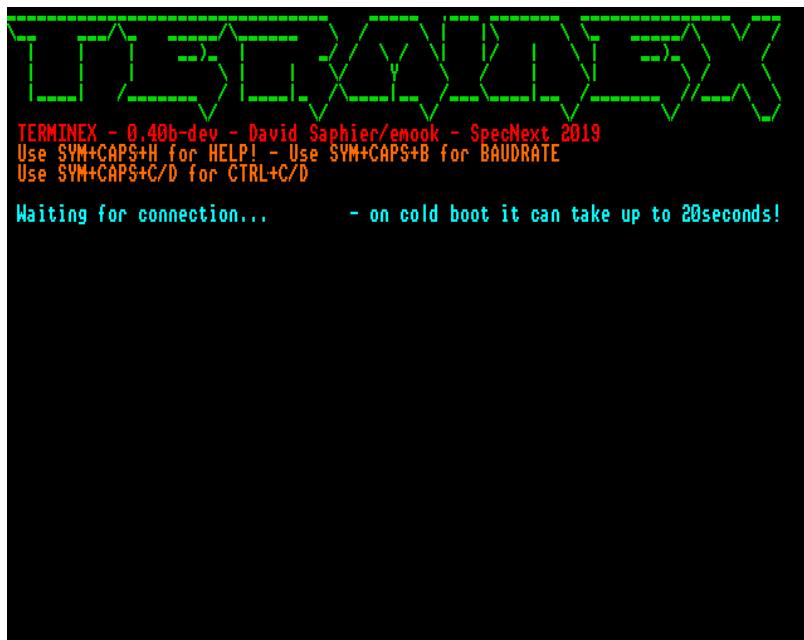
Nombre	Pin	Uso
5V	2 ó 4	Alimentación (opcional)
GND	6	Tierra
UART TX (GPIO 14)	8	Conexión Serie
UART RX (GPIO 15)	10	Conexión Serie
PCM CLK (GPIO 18)	12	Conexión de audio
PCM FS (GPIO 19)	35	Conexión de audio
PCM DIN (GPIO 20)	38	Conexión de audio
PCM DOUT (GPIO 21)	40	Conexión de audio

En el caso de que se conecte alguno de los pines de alimentación, **no se debe hacer simultáneamente a la Raspberry Pi y al ZX DOS+**.

Se ha de tener cuidado de utilizar una fuente únicamente con uno de los dos, el ZX DOS+ o la Raspberry Pi conectada al addon. Aquel dispositivo que esté alimentado, suministrará energía al otro. Es decir, si se alimenta el ZX DOS+, este proveerá energía a la Raspberry Pi, o bien, si se alimenta la Raspberry Pi, esta proveerá de energía al ZX DOS+.

Uso desde NextZXOS

Para comprobar que la comunicación serie con la Raspberry Pi funciona, se puede utilizar el programa **Terminex**, que está disponible en la distribución de Next en [/DEMONS/UART/TERMINEX/TERMINEX.SNX](#). Aparecerá una pantalla como la siguiente



Si la conexión es correcta y la Raspberry Pi está completamente encendida (desde que se enciende, puede llegar a tardar entre 1 y 2 minutos en estar disponible), debería aparecer una consola con el texto:

```
SUP>
```

Si aparecieran mensajes **Waiting for connection..** o **Waiting for DietPi..** podría ser que aún se estuviera iniciando la Raspberry Pi, o bien la conexión no es correcta.

Para comprobar la conexión de audio, se puede intentar cargar un archivo **.TZX** desde el navegador o bien elegir un fichero de audio (por ejemplo **.MOD**, **XM**, **SND** o **SID**) y comprobar que, tras ser copiado en la Raspberry Pi, podemos escuchar el sonido (bien de carga de cinta, bien de música).

Solución de problemas

Gestión de imágenes de firmware

Existen distintas herramientas que permiten generar y/o editar el contenido de los ficheros [ZX1](#), [ZX2](#), [ZXD](#).

zx123_tool

Esta es una herramienta que analiza, extrae o añade datos en ficheros de imagen de SPI flash de ZX-Uno, ZX DOS y otros dispositivos similares.

Para poder utilizarla se necesita [Python 3](#). Según el sistema operativo que se utilice puede que sea necesario [instalarlo](#).

Teniendo Python 3, basta con descargar la última versión de la herramienta desde su repositorio oficial, [en este enlace](#).

Una vez descomprimido, se debe invocar desde una consola el script principal usando Python 3. Esto puede variar según el sistema operativo.

Por ejemplo, en Windows, suele ser:

```
py -3 zx123_tool.py
```

Mientras que en otros sistemas operativos debería bastar con algo parecido a:

```
python3 ./zx123_tool.py
```

También hará falta un archivo de imagen flash. Este se puede obtener desde el core de Spectrum, en modo "root", con alguno de los comandos [back16m](#), [backzx2](#) o [backzxd](#). Tras obtener el fichero generado en la microSD, se puede "limpiar" dejando sólo el core de Spectrum y la primera ROM de Spectrum con un comando similar a este:

```
... zx123_tool.py -i FLASH.ZXD -w -o FLASHempty.ZXD
```

Donde [FLASH.ZXD](#) es la ruta al fichero obtenido desde el core de Spectrum, y [FLASHempty.ZXD](#) es la ruta al nuevo fichero "limpio".

Mostrar contenido de una imagen

Para ver el contenido de una imagen llamada **FLASH.ZXD** (cores instalados y algunos datos de configuración), se puede usar el comando

```
... zx123_tool.py -i FLASH.ZXD -l
```

Para mostrar contenido de esa misma imagen, incluyendo datos de ROMs de ZX Spectrum:

```
... zx123_tool.py -i FLASH.ZXD -l -r
```

Modificar la BIOS de una imagen

Para modificar la BIOS de un fichero llamado **FLASH.ZXD**, usando la BIOS en otro fichero llamado **FIRMWARE.ZXD**

```
...zx123_tool.py -i FLASH.ZXD -a BIOS,FIRMWARE.ZXD
```

Además de instalar la BIOS, se pueden modificar algunos de los valores por defecto. Por ejemplo, con las opciones; **-m** para el modo de vídeo: 0 (PAL), 1 (NTSC) ó 2 (VGA), **-k** para la distribución del teclado: 0 (Auto), 1 (ES), 2 (EN) ó 3 (Spectrum).

Así, para modificar la BIOS de un fichero llamado **FLASH.ZXD**, usando la BIOS en otro fichero llamado **FIRMWARE.ZXD**, y además configurar el modo de vídeo en VGA, y el teclado en modo Spectrum (para gomaDOS`+):

```
...zx123_tool.py -i FLASH.ZXD -a BIOS,FIRMWARE.ZXD -m 2 -k 3
```

Existen también opciones para ajustar el tiempo de espera inicial de la BIOS, el core a ejecutar por defecto, o la ROM de Spectrum a utilizar por defecto. Véase la [documentación de la herramienta](#) para más información.

Añadir una ROM de Spectrum a una imagen

Para añadir una ROM de Spectrum llamada `48.rom`, poniendo el nombre `Spec48` y ocupando el slot 5, se puede usar un comando como:

```
...zx123_tool.py -i FLASH.ZXD -a ROM,5,xdnlh17,Spec48,48.rom
```

Véase la [documentación de la herramienta](#) para ver todas las posibles opciones a la hora de añadir una ROM de Spectrum.

Entre los datos que se indican al añadir una ROM, hay una serie de indicadores para definir qué opciones de hardware, etc. se desean habilitar o deshabilitar al cargar esa ROM en concreto, según se puede ver en esta tabla:

<code>i</code>	Habilitar teclado issue 3 (en vez de issue 2)
<code>c</code>	Deshabilitar la contención de memoria
<code>d</code>	Habilitar DivMMC
<code>n</code>	Habilitar NMI DivMMC (menú de esxdos)
<code>p</code>	Usar timings de Pentagon
<code>t</code>	Usar timings de 128K
<code>s</code>	Deshabilitar puertos de DivMMC y ZXMMC
<code>m</code>	Habilitar MMU horizontal de Timex
<code>h</code>	Deshabilitar bit alto de ROM (bitd 2 de 1FFD)
<code>l</code>	Deshabilitar bit bajo de ROM (bit 4 de 7FFD)
<code>1</code>	Deshabilitar puerto 1FFD (paginado de +2A/3)
<code>7</code>	Deshabilitar puerto 7FFD (paginado de 128K)
<code>2</code>	Deshabilitar TurboSound (chip AY secundario)
<code>a</code>	Deshabilitar chip AY
<code>r</code>	Deshabilitar modo Radastaniano
<code>x</code>	Deshabilitar modo Timex
<code>u</code>	Deshabilitar ULAPlus

Instalar un Core en una imagen

Para instalar un core en la posición 3, desde un fichero llamado **NEXT.ZXD**, llamándolo **Spectrum Next**, usar un comando como este:

```
...zx123_tool.py -i FLASH.ZXD -a 'CORE,3,Spectrum Next,NEXT.ZXD'
```

Si además se quiere configurar como el core por defecto, se puede indicar también con un comando como:

```
...zx123_tool.py -i FLASH.ZXD -a 'CORE,3,Spectrum Next,NEXT.ZXD' -c 3
```

Modificar la ROM de esxdos de una imagen

De forma similar a como se instala el firmware de la BIOS, se puede instalar directamente un fichero con la ROM de esxdos, con un comando como:

```
...zx123_tool.py -i FLASH.ZXD -a esxdos,ESXMMC.BIN
```

Combinar varias órdenes en una única línea

Notar que se pueden acumular varias órdenes en una única línea de comandos. Por ejemplo, para "limpiar" un archivo de imagen llamado **FLASH.ZXD**, creando uno nuevo llamado **FLASHnew.ZXD**, instalar la BIOS desde el fichero **FIRMWARE.ZXD**, configurar el modo de vídeo en VGA, el teclado en modo Spectrum (para gomaDOS`+), añadir una ROM de Spectrum llamada **48.rom**, poniendo el nombre **Spec48** y ocupando el slot 5, instalar un core en la posición 3, desde un fichero llamado **NEXT.ZXD**, llamándolo **Spectrum Next**, configurado como el core por defecto, usar un comando como este:

```
... zx123_tool.py -i FLASH.ZXD -w -o FLASHnew.ZXD -a BIOS,FIRMWARE.ZXD -m 2 -k 3 -a ROM,5,xdnlh17,Spec48,48.rom -a 'CORE,3,Spectrum Next,NEXT.ZXD' -c 3
```

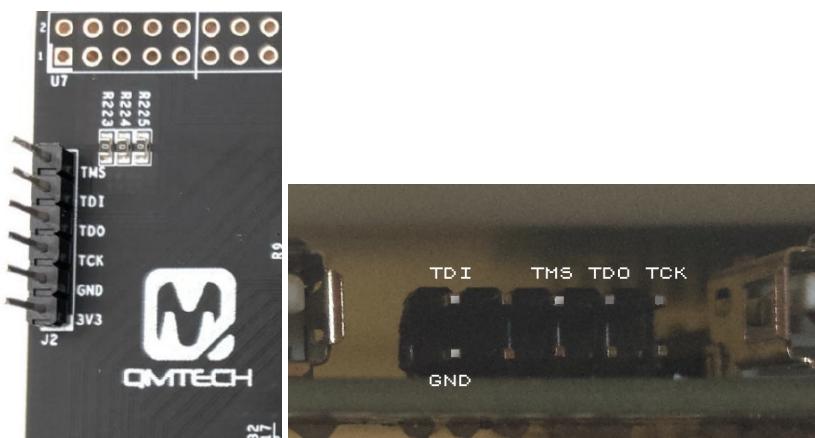
Recuperación del firmware

En algunos casos (por ejemplo al instalar un core experimental o hacer una actualización del core de ZX Spectrum o la BIOS) puede suceder que el ZX DOS+ deje de arrancar. Se encienden los LEDs pero no hay imagen ni responde a las distintas combinaciones de teclado para acceder a la BIOS, etc.

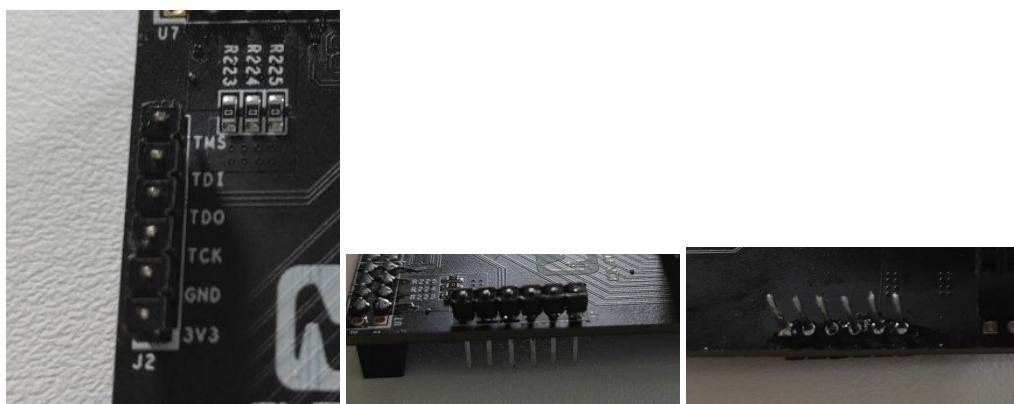
En esta situación, existen diferentes métodos de recuperación que permiten volver a instalar el firmware.

Preparación del cableado

En los siguientes pasos de recuperación se habla de conectar cables puente o USB-Blaster a la placa de ZX DOS+ o gomaDOS+. Para ello, usar las siguientes imágenes como referencia.



Notar que, en algunos modelos, el conector de JTAG está con los pines por la parte de debajo.



NO se ha de conectar la línea de 3V



En el caso de utilizar USB-Blaster, el gomaDOS+ tiene la distribución adecuada para conectar directamente el conector 2x5 incluido. Para ZX DOS+, podría ser necesario preparar el cableado adecuado, comparando las imágenes anteriores.

Recuperación usando una Raspberry Pi

Material necesario:

- Raspberry Pi (con tarjeta SD, teclado, monitor, fuente de alimentación, etc.) y con conexión a internet
- 5 [cables puente para prototipos](#) (idealmente, hembra en los dos extremos), o bien, en vez de los cables puente, un adaptador USB-Blaster
- Una [llave Allen](#) del tamaño adecuado para poder retirar la tapa del ZX DOS+ o bien un destornillador de estrella adecuado si se va abrir un gomaDOS+ (esto no es necesario si se utiliza USB-Blaster)
- Tarjeta microSD para el ZX DOS+/gomaDOS+ con la primera partición en formato FAT16 o FAT32
- Teclado (no necesario en gomaDOS+) y monitor para conectar el ZX DOS+

Software necesario:

- Imagen Flash y recovery para ZX DOS+ (LX25), del [repositorio oficial, en este enlace](#)

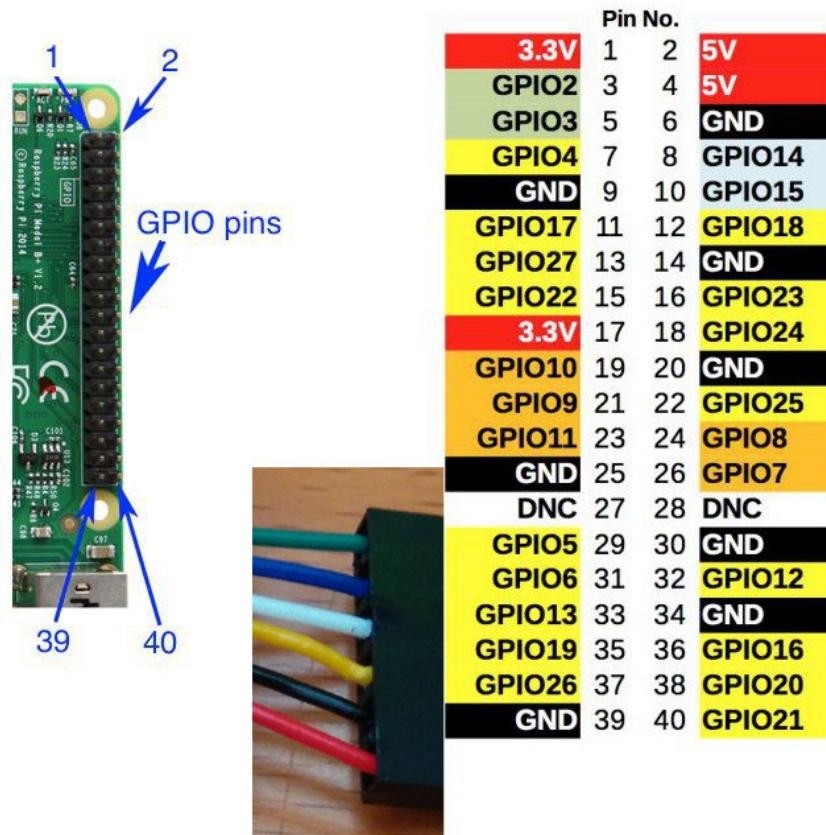
Pasos a seguir:

1. Si no estuviera ya, instalar Raspberry Pi OS (antes llamado Raspbian) en la Raspberry Pi (usando [la descarga oficial, NOOBS, PINN](#), etc.)
2. Instalar Open OCD en la Raspberry Pi:

```
sudo apt-get update
sudo apt-get install git autoconf libtool make pkg-config
sudo apt-get install libusb-1.0-0 libusb-1.0-0-dev telnet
sudo apt-get install libusb-dev libftdi-dev
git clone git://git.code.sf.net/p/openocd/code openocd-code
cd openocd-code/
./bootstrap
./configure --enable-usb_blaster --enable-sysfsgpio --enable-bcm2835gpio
make
sudo make install
cd ..
rm -rf ./openocd-code
```

- Conectar el USB-Blaster o los cables puente para GPIO tal y como se explica anteriormente. Si se va a hacer la conexión usando GPIO, abrir la carcasa del ZX DOS+ o el gomaDOS+ y conectar las líneas de JTAG de la FPGA (TMS, TDI, TDO, TCK y GND) con los cables a los pines GPIO de la Raspberry Pi.

Si se hace conexión vía GPIO, tomar nota de los pines elegidos, teniendo cuidado de conectar GND con GND.



En este ejemplo, se utilizarán los pines 31, 33, 35, 37 y 39 (correspondientes a GPIO #6, GPIO #13, GPIO #19, GPIO #26 y GND), de la siguiente manera:

JTAG ZX DOS+	GPIO	Pin Raspberry Pi
TMS	GPIO#6	31
TDI	GPIO#13	33
TDO	GPIO#19	35
TCK	GPIO#26	37
GND	GND	39

- Copiar en la Raspberry Pi el fichero `recovery.zxd.bit` obtenido anteriormente del [repositorio oficial](#). En nuestro ejemplo, se dejará en `/home/pi/zxdosplus/unbrick/`

5. Para la conexión usando GPIO, realizar una copia del archivo de configuración de Open OCD, en el mismo lugar donde está `recovery.zxd.bit`. Este paso no es necesario si se usa USB-Blaster.

```
cp /usr/local/share/openocd/scripts/interface/raspberrypi2-native.cfg  
/home/pi/zxdosplus/unbrick/
```

6. Para la conexión vía GPIO, editar la copia de `raspberrypi2-native.cfg` actualizando `bcm2835gpio_jtag_nums` (y descomentando, si fuera necesario), según como se haya hecho la conexión entre JTAG y GPIO en la línea `bcm2835gpio_jtag_nums`. En nuestro ejemplo:

```
# Header pin numbers: 37 31 33 35  
bcm2835gpio_jtag_nums 26 6 13 19
```

7. Comentar, si no lo está, la línea `bcm2835gpio_swd_nums` (de nuevo, no necesario si la conexión es con USB-Blaster):

```
#bcm2835gpio_swd_nums 11 25
```

8. Añadir, al final, la línea `adapter speed 250` (no necesario para uso con USB-Blaster):

```
adapter speed 250
```

9. Encender el ZX DOS+ o el gomaDOS+

10. Asegurarnos de que estamos en el directorio donde se encuentra el archivo `recovery.zxd.bit`, y lanzar el comando que carga la BIOS en modo recuperación, indicando la ruta al archivo `raspberrypi2-native.cfg` que habíamos editado anteriormente.

Para conexión vía GPIO:

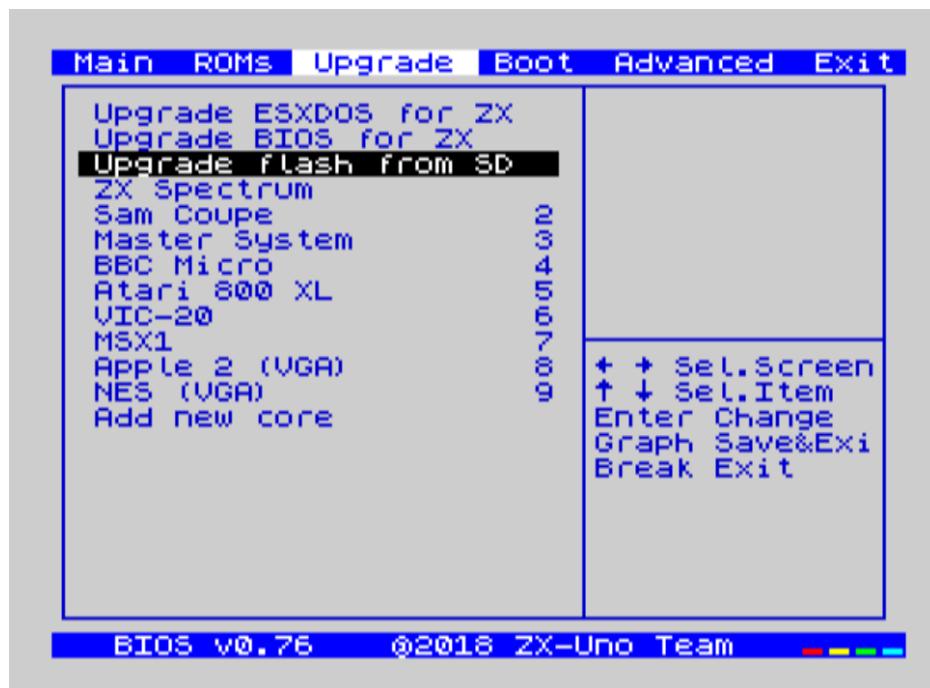
```
cd /home/pi/zxdosplus/unbrick  
sudo openocd -f /home/pi/zxdosplus/unbrick/raspberrypi2-native.cfg -f  
/usr/local/share/openocd/scripts/cpld/xilinx-xc6s.cfg -c "init; xc6s_program xc6s.tap;  
pld load 0 recovery.zxd.bit ; exit"
```

Con USB-Blaster:

```
sudo openocd -f /usr/local/share/openocd/scripts/interface/altera-usb-blaster.cfg -f  
/usr/local/share/openocd/scripts/cpld/xilinx-xc6s.cfg -c "init; xc6s_program xc6s.tap;  
pld load 0 recovery.zxd.bit ; exit"
```

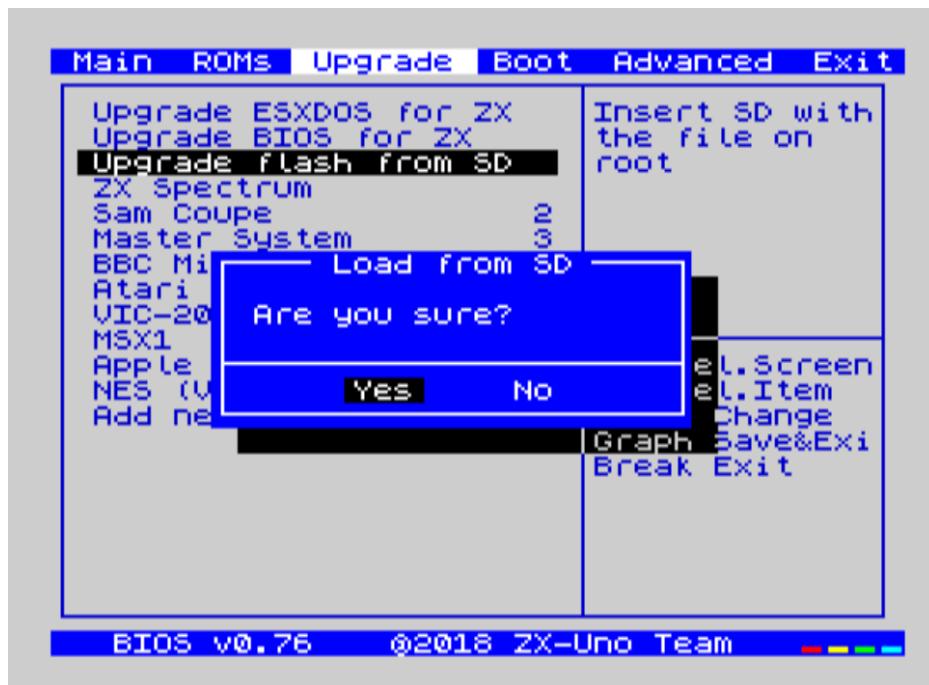
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+) para cambiar entre modo de vídeo RGB y VGA, por si acaso la BIOS ha arrancado en un modo que no corresponde a la conexión del monitor.



- Insertar en el ZX-DOS+ la tarjeta microSD con la primera partición en formato FAT16 o FAT32, y en la que habremos copiado el fichero **FLASH.ZXD** descargado anteriormente.
- Si se está utilizando USB-Blaster, desconectar el cable.

14. Elegir la opción **Upgrade Flash from SD**. Pulsar Enter, elegir **Yes**, y pulsar Enter de nuevo para comenzar el proceso que graba de nuevo la Flash.



Este proceso sustituirá todos los cores instalados, la BIOS, así como las ROMs de ZX Spectrum y la configuración por lo que haya en la imagen, y no se puede deshacer.



Habitualmente, la imagen de recuperación está configurada para usar un teclado PS/2 y no el teclado de membrana de Spectrum, así que, en el caso de gomaDOS+, puede que no funcionen las combinaciones como **Caps Shift + 5**, etc. En este caso, se ha de cambiar el modo de teclado a **PC XT** (**Caps Shift+Symbol Shift+U** y luego **9**), para que funcionen de nuevo temporalmente.

15. Tras unos minutos, el proceso finalizará, y podremos comprobar como, al apagar y encender, el ZX DOS+ (o el gomaDOS+) vuelve a arrancar correctamente.



Si no se obtiene imagen, pulsar de nuevo **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+) para cambiar entre modo de vídeo RGB y VGA. En este caso, sería necesario acceder a la BIOS y cambiar el **ajuste avanzado correspondiente** para indicar la configuración de nuestro monitor.

En el caso de gomaDOS+, como la configuración de la imagen de recuperación por defecto espera un teclado PS/2, se pueden seguir los siguientes pasos para configurar la BIOS correctamente:



1. Si no hay imagen, cambiar entre modo vídeo RGB y modo VGA (**Caps Shift+Symbol Shift+G**)
2. Cambiar al modo **PC XT** de teclado (**Caps Shift+Symbol Shift+U** y luego **9**)
3. Reiniciar el gomaDOS+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+B**)
4. Rápidamente, pulsar **Caps Shift + 1**
5. De nuevo, si no hay imagen, cambiar entre modo VGA y modo vídeo RGB (**Caps Shift+Symbol Shift+G**)
6. Navegar por la BIOS y configurar las siguientes opciones:
 - **Advanced** → **Keyboard Layout: Spectrum**
 - **Advanced** → **video: VGA** (sólo si no teníamos imagen)
7. Guardar los cambios de la BIOS:
 - **Exit** → **Save changes and exit**. Apagar completamente el gomaDOS+ y volver a encenderlo

Recuperación usando macOS y Cable USB-Blaster

Material necesario:

- Cable USB-Blaster [preparado con las conexiones adecuadas para ZX DOS+](#)
- Imagen Flash y recovery para ZX DOS+ (LX25). Los mismos que se indican para Raspberry pi, del [repositorio oficial, en este enlace](#)

Software necesario:

- Sistema macOS
- Carpeta data adicional para UrJTAG, obtenido desde [aqui](#)
- [Homebrew para macOS](#)
- UrJTAG: según las instrucciones de [este enlace](#):

Pasos a seguir:

1. Preparar la instalación de UrJTAG

```
brew install libftdi libusb pkg-config  
git clone https://github.com/C-Elegans/urjtag.git  
cd urjtag
```

2. Copiar la carpeta data adicional para UrJTAG, dentro de la carpeta data de urjtag.

3. Lanzar el proceso de compilacion:

```
./configure --with-libftdi --with-libusb --with-ftd2xx --with-inpout32 --enable-python=no  
make -j4  
sudo make install
```

4. Copiar el archivo **FLASH.ZXD** en la raiz de la tarjeta microSD para el ZX DOS+.

5. Conectar el cable USB-Blaster al ZX DOS+ y al Mac

6. Encender el ZX DOS+ o el gomaDOS+

7. Asegurarnos de que estamos en el directorio donde se encuentra el archivo **recovery.zxd.bit**, y lanzar el comando **jtag**.

8. Aparecerá una consola donde se han de escribir los comandos:

```
cable usbblaster  
detect  
pld load recovery.zxd.bit
```

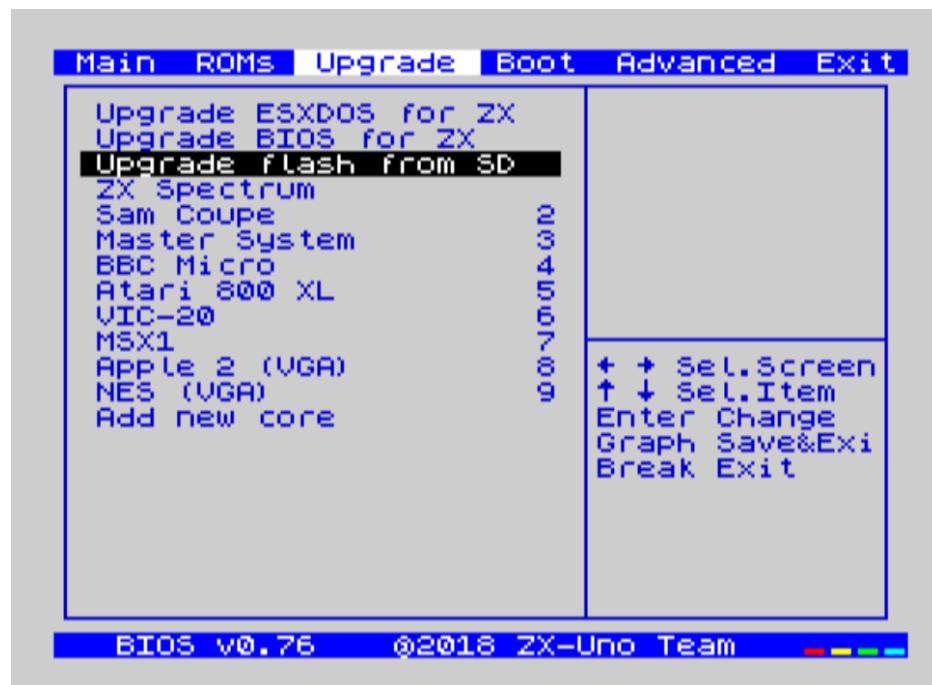


En el comando detect es importante que aparezca el dispositivo detectado. Puede que sea necesario lanzar el comando **detect** de manera repetida hasta que aparezca.

```
UrJTAG 2018.06 #  
Copyright (C) 2002, 2003 ETC s.r.o.  
Copyright (C) 2007, 2008, 2009 Kolja Waschk and the respective authors  
  
UrJTAG is free software, covered by the GNU General Public License, and you are  
welcome to change it and/or distribute copies of it under certain conditions.  
There is absolutely no warranty for UrJTAG.  
  
warning: UrJTAG may damage your hardware!  
Type "quit" to exit, "help" for help.  
  
jtag> cable usbblaster  
Connected to libftdi driver.  
jtag> detect  
IR length: 6  
Chain length: 1  
Device Id: 00100100000000000100000010010011 (0x24004093)  
Manufacturer: Xilinx (0x093)  
Part(0): xc6slx25 (0x4004)  
Stepping: 2  
Filename: /usr/local/share/urjtag/xilinx/xc6slx25/xc6slx25  
jtag> pld load recovery.zxd.bit  
Bitstream information:  
    Design: tld_zxuno_lx16.ncd;UserID=0xFFFFFFFF  
    Part name: 6slx25ftg256  
    Date: 2020/05/29  
    Time: 01:32:07  
    Bitstream length: 801462  
jtag>
```

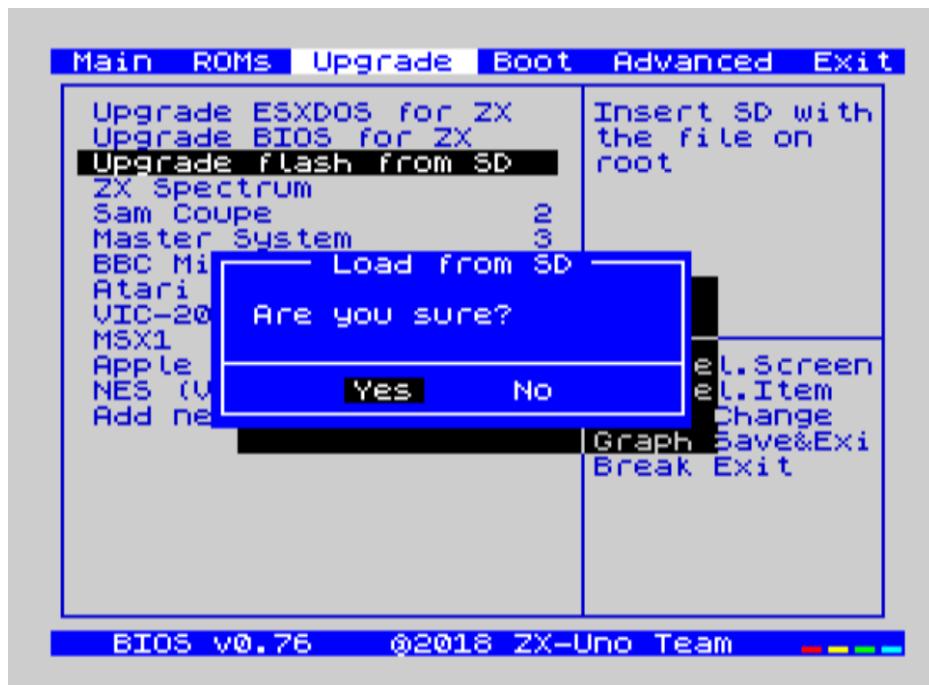
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+) para cambiar entre modo de vídeo RGB y VGA, por si acaso la BIOS ha arrancado en un modo que no corresponde a la conexión del monitor.



- Insertar en el ZX-DOS+ la tarjeta microSD con la primera partición en formato FAT16 o FAT32, y en la que habremos copiado el fichero **FLASH.ZXD** descargado anteriormente.
- Desconectar el cable USB-Blaster.

12. Elegir la opción **Upgrade Flash from SD**. Pulsar Enter, elegir **Yes**, y pulsar Enter de nuevo para comenzar el proceso que graba de nuevo la Flash.



Este proceso sustituirá todos los cores instalados, la BIOS, así como las ROMs de ZX Spectrum y la configuración por lo que haya en la imagen, y no se puede deshacer.



Habitualmente, la imagen de recuperación está configurada para usar un teclado PS/2 y no el teclado de membrana de Spectrum, así que, en el caso de gomaDOS+, puede que no funcionen las combinaciones como **Caps Shift + 5**, etc. En este caso, se ha de cambiar el modo de teclado a **PC XT** (**Caps Shift+Symbol Shift+U** y luego **9**), para que funcionen de nuevo temporalmente.

13. Tras unos minutos, el proceso finalizará, y podremos comprobar como, al apagar y encender, el ZX DOS+ (o el gomaDOS+) vuelve a arrancar correctamente.



Si no se obtiene imagen, pulsar de nuevo **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+) para cambiar entre modo de vídeo RGB y VGA. En este caso, sería necesario acceder a la BIOS y cambiar el **ajuste avanzado correspondiente** para indicar la configuración de nuestro monitor.

En el caso de gomaDOS+, como la configuración de la imagen de recuperación por defecto espera un teclado PS/2, se pueden seguir los siguientes pasos para configurar la BIOS correctamente:



1. Si no hay imagen, cambiar entre modo vídeo RGB y modo VGA (**Caps Shift+Symbol Shift+G**)
2. Cambiar al modo **PC XT** de teclado (**Caps Shift+Symbol Shift+U** y luego **9**)
3. Reiniciar el gomaDOS+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+B**)
4. Rápidamente, pulsar **Caps Shift + 1**
5. De nuevo, si no hay imagen, cambiar entre modo VGA y modo vídeo RGB (**Caps Shift+Symbol Shift+G**)
6. Navegar por la BIOS y configurar las siguientes opciones:
 - **Advanced → Keyboard Layout: Spectrum**
 - **Advanced → video: VGA** (sólo si no teníamos imagen)
7. Guardar los cambios de la BIOS:
 - **Exit → Save changes and exit** .Apagar completamente el gomaDOS+ y volver a encenderlo



En Linux, usando el mismo software UrJTAG, el proceso debería ser similar, aunque en vez de instalar las dependencias (libftdi libusb pkg-config) mediante brew, se debería hacer mediante apt, yum o similar

Recuperación usando Windows y Cable USB-Blaster

Material necesario:

- Cable USB-Blaster. [preparado con las conexiones adecuadas para ZXDOS+](#)
- Imagen Flash y recovery para ZXDOS+ (LX25). Los mismos que se indican para Raspberry pi, del [repositorio oficial, en este enlace](#)

Software necesario:

- Sistema Windows
- Drivers de USB-Blaster para Windows, disponible en [el foro de ZX-Uno](#)
- UrJTAG, para Windows, disponible en [el repositorio oficial](#)

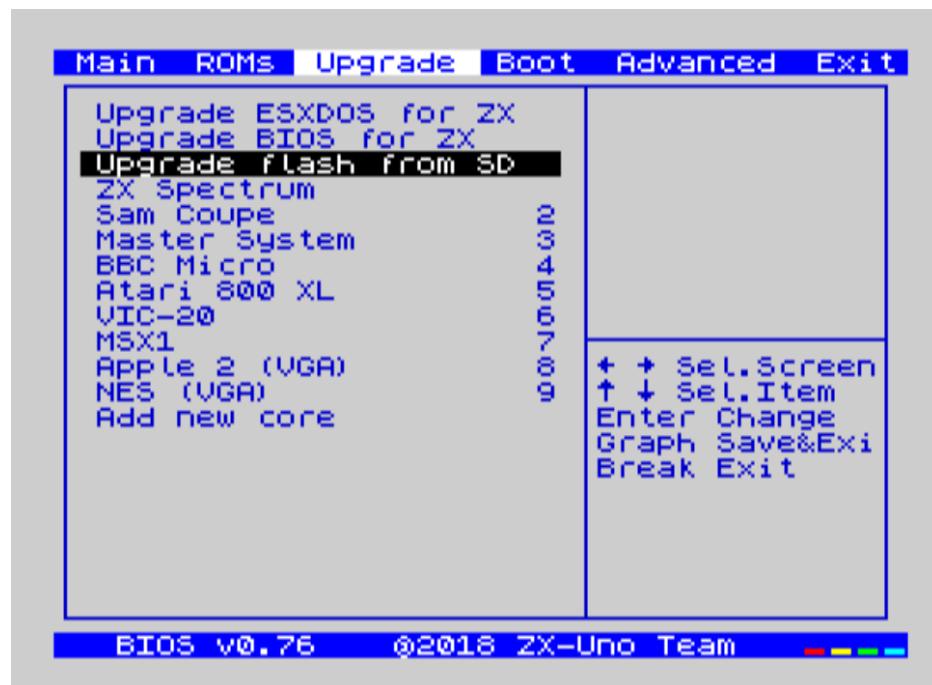
Pasos a seguir:

1. Descomprimir el fichero ZIP con los drivers (obtenido en [el foro de ZX-Uno](#))
2. Conectar el cable USB-Blaster al PC con Windows e instalar el driver eligiendo la opción de añadir ficheros manualmente y seleccionando la carpeta **drivers** que se ha obtenido al descomprimir el fichero anterior
3. Descomprimir el software de UrJTAG, obtenido en [el repositorio oficial](#)
4. Copiar el archivo **FLASH.ZXD** en la raíz de la tarjeta microSD para el ZX DOS+.
5. Conectar el cable USB-Blaster al ZX DOS+ y al PC con Windows
6. Encender el ZX DOS+ o el goma DOS+
7. Asegurarnos de que estamos en el directorio donde se encuentra el archivo **recovery.zxd.bit**, y lanzar **jtag.exe**.
8. Aparecerá una consola donde se han de escribir los comandos:

```
cable usbblaster  
detect  
pld load recovery.zxd.bit
```

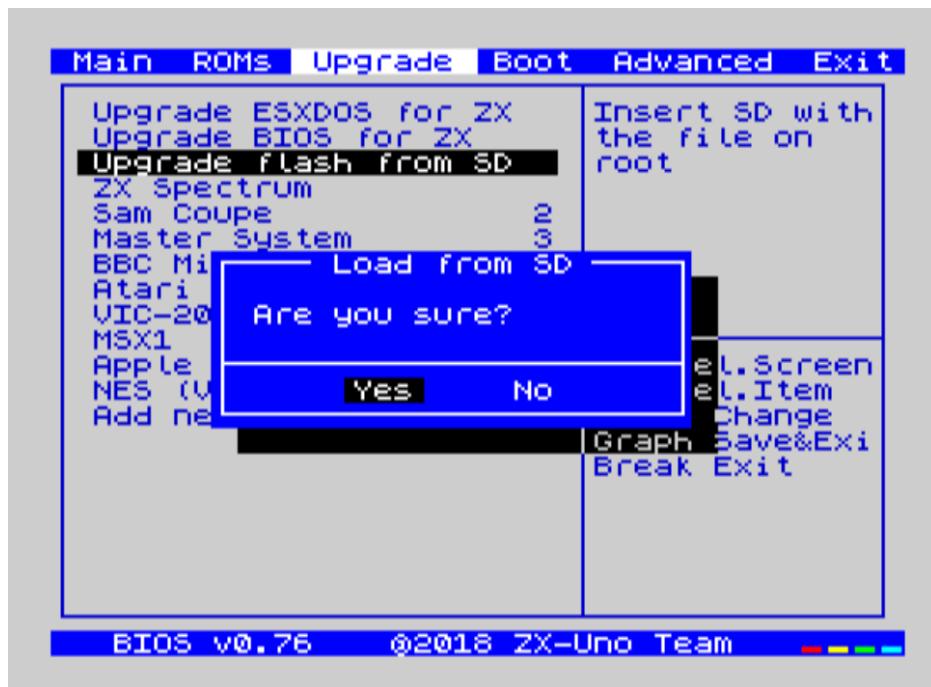
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+) para cambiar entre modo de vídeo RGB y VGA, por si acaso la BIOS ha arrancado en un modo que no corresponde a la conexión del monitor.



- Insertar en el ZX-DOS+ la tarjeta microSD con la primera partición en formato FAT16 o FAT32, y en la que habremos copiado el fichero **FLASH.ZXD** descargado anteriormente.
- Desconectar el cable USB-Blaster.

12. Elegir la opción **Upgrade Flash from SD**. Pulsar Enter, elegir **Yes**, y pulsar Enter de nuevo para comenzar el proceso que graba de nuevo la Flash.



Este proceso sustituirá todos los cores instalados, la BIOS, así como las ROMs de ZX Spectrum y la configuración por lo que haya en la imagen, y no se puede deshacer.



Habitualmente, la imagen de recuperación está configurada para usar un teclado PS/2 y no el teclado de membrana de Spectrum, así que, en el caso de gomaDOS+, puede que no funcionen las combinaciones como **Caps Shift + 5**, etc. En este caso, se ha de cambiar el modo de teclado a **PC XT** (**Caps Shift+Symbol Shift+U** y luego **9**), para que funcionen de nuevo temporalmente.

13. Tras unos minutos, el proceso finalizará, y podremos comprobar como, al apagar y encender, el ZX DOS+ (o el gomaDOS+) vuelve a arrancar correctamente.



Si no se obtiene imagen, pulsar de nuevo **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+) para cambiar entre modo de vídeo RGB y VGA. En este caso, sería necesario acceder a la BIOS y cambiar el **ajuste avanzado correspondiente** para indicar la configuración de nuestro monitor.

En el caso de gomaDOS+, como la configuración de la imagen de recuperación por defecto espera un teclado PS/2, se pueden seguir los siguientes pasos para configurar la BIOS correctamente:



1. Si no hay imagen, cambiar entre modo vídeo RGB y modo VGA (**Caps Shift+Symbol Shift+G**)
2. Cambiar al modo **PC XT** de teclado (**Caps Shift+Symbol Shift+U** y luego **9**)
3. Reiniciar el gomaDOS+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+B**)
4. Rápidamente, pulsar **Caps Shift + 1**
5. De nuevo, si no hay imagen, cambiar entre modo VGA y modo vídeo RGB (**Caps Shift+Symbol Shift+G**)
6. Navegar por la BIOS y configurar las siguientes opciones:
 - **Advanced** → **Keyboard Layout: Spectrum**
 - **Advanced** → **video: VGA** (sólo si no teníamos imagen)
7. Guardar los cambios de la BIOS:
 - **Exit** → **Save changes and exit**. Apagar completamente el gomaDOS+ y volver a encenderlo

Referencias

Spectrum

Scan Codes

101-, 102-, and 104-key Scan Codes

KEY	MAKE	BREAK
A	1C	F0,1C
B	32	F0,32
C	21	F0,21
D	23	F0,23
E	24	F0,24
F	2B	F0,2B
G	34	F0,34
H	33	F0,33
I	43	F0,43
J	3B	F0,3B
K	42	F0,42
L	4B	F0,4B
M	3A	F0,3A
N	31	F0,31
O	44	F0,44
P	4D	F0,4D
Q	15	F0,15
R	2D	F0,2D
S	1B	F0,1B
T	2C	F0,2C
U	3C	F0,3C
V	2A	F0,2A
W	1D	F0,1D
X	22	F0,22
Y	35	F0,35
Z	1A	F0,1A
0	45	F0,45
1	16	F0,16
2	1E	F0,1E
3	26	F0,26
4	25	F0,25
5	2E	F0,2E
6	36	F0,36
7	3D	F0,3D
8	3E	F0,3E
9	46	F0,46
.	0E	F0,0E
-	4E	F0,4E
=	55	F0,55
\	5D	F0,5D
BKSP	66	F0,66
SPACE	29	F0,29
TAB	0D	F0,0D
CAPS	58	F0,58
L SHIFT	12	F0,12
L CTRL	14	F0,14
L GUI	E0,1F	E0,F0,1F
L ALT	11	F0,11
R SHIFT	59	F0,59
R CTRL	E0,14	E0,F0,14
R GUI	E0,27	E0,F0,27
R ALT	E0,11	E0,F0,11
APPS	E0,2F	E0,F0,2F
ENTER	5A	F0,5A
ESC	76	F0,76
F1	5	F0,05
F2	6	F0,06
F3	4	F0,04
F4	0C	F0,0C
F5	3	F0,03
F6	0B	F0,0B
F7	83	F0,83
F8	0A	F0,0A
F9	1	F0,01
F10	9	F0,09
F11	78	F0,78
F12	7	F0,07
PRNT SCRN	E0,12, E0,7C	E0,F0, 7C,E0, F0,12
SCROLL	7E	F0,7E
PAUSE	E1,14,77, E1,F0,14,F0,77	-NONE-

101-, 102-, and 104-key Scan Codes

KEY	MAKE	BREAK
[54	F0,54
INSERT	E0,70	E0,F0,70
HOME	E0,6C	E0,F0,6C
PG UP	E0,7D	E0,F0,7D
DELETE	E0,71	E0,F0,71
END	E0,69	E0,F0,69
PG DN	E0,7A	E0,F0,7A
U ARROW	E0,75	E0,F0,75
L ARROW	E0,6B	E0,F0,6B
D ARROW	E0,72	E0,F0,72
R ARROW	E0,74	E0,F0,74
NUM	77	F0,77
KP /	E0,4A	E0,F0,4A
KP *	7C	F0,7C
KP -	7B	F0,7B
KP +	79	F0,79
KP EN	E0,5A	E0,F0,5A
KP .	71	F0,71
KP 0	70	F0,70
KP 1	69	F0,69
KP 2	72	F0,72
KP 3	7A	F0,7A
KP 4	6B	F0,6B
KP 5	73	F0,73
KP 6	74	F0,74
KP 7	6C	F0,6C
KP 8	75	F0,75
KP 9	7D	F0,7D
]	5B	F0,5B
;	4C	F0,4C
,	52	F0,52
.	41	F0,41
/	49	F0,49
	4A	F0,4A

Windows Multimedia Scan Codes

Key	Make Code	Break Code
Next Track	E0, 4D	E0, F0, 4D
Previous Track	E0, 15	E0, F0, 15
Stop	E0, 3B	E0, F0, 3B
Play/Pause	E0, 34	E0, F0, 34
Mute	E0, 23	E0, F0, 23
Volume Up	E0, 32	E0, F0, 32
Volume Down	E0, 21	E0, F0, 21
Media Select	E0, 50	E0, F0, 50
E-Mail	E0, 48	E0, F0, 48
Calculator	E0, 2B	E0, F0, 2B
My Computer	E0, 40	E0, F0, 40
WWW Search	E0, 10	E0, F0, 10
WWW Home	E0, 3A	E0, F0, 3A
WWW Back	E0, 38	E0, F0, 38
WWW Forward	E0, 30	E0, F0, 30
WWW Stop	E0, 28	E0, F0, 28
WWW Refresh	E0, 20	E0, F0, 20
WWW Favorites	E0, 18	E0, F0, 18

ACPI Scan Codes

Key	Make Code	Break Code
Power	E0, 37	E0, F0, 37
Sleep	E0, 3F	E0, F0, 3F
Wake	E0, 5E	E0, F0, 5E

Registros E/S para control de ZXDOS+

En el core de Spectrum se dispone de los puertos **\$FC3B** y **\$FD3B**, registrados y asignados por el comité **ZXI**. A través de estos puertos se accede a un total de 256 registros de E/S diferentes exclusivos para la familia del ZX-Uno.

El puerto **\$FC3B (64571)** guarda la dirección (**\$00 - \$FF**) del registro de E/S al que se quiere acceder. Puede leerse para saber cuál es la última dirección de registro asignada.

El puerto **\$FD3B (64827)** es el puerto de acceso al registro seleccionado con el puerto anterior. Su sentido (lectura/escritura) dependerá de la implementación de cada registro.

Por ejemplo, para asignar el banco 16 de la SRAM al espacio de direcciones **\$C000 - \$FFFF** durante el modo boot, mediante el registro **MASTERMAPPER**, se haría así:

```
ld bc,$fc3b      ;Puerto para establecer el número de registro a usar
ld a,1           ;Registro $01 (MASTERMAPPER)
out (c),a        ;Se selecciona. A partir de ahora, cualquier acceso a $FD3B está
usando MASTERMAPPER
inc b            ;Puerto de acceso al registro ($FD3B, basta con incrementar B)
ld a,16          ;Banco 16 de la SRAM
out (c),a        ;Se escribe en el registro MASTERMAPPER
```

Los registros implementados en la familia del ZX-Uno se describen a continuación.

\$00 MASTERCONF

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000001	00000001

Formato binario (en **negrita**, los campos que sólamente pueden alterarse cuando LOCK=0):

LOCK	MODE1	DISCONT	MODE0	I2KB	DISNMI	DIVEN	BOOTM
------	-------	---------	-------	------	--------	-------	-------

- **BOOTM**: a 1 indica que ZX-Uno está en modo boot (modo configuración). El modo boot sólo tiene sentido mientras se está ejecutando el firmware de arranque, donde se permite configurar algunos aspectos de ZX-Uno antes de pasar al modo ejecución. MASTERCONF siempre puede ser leído, tanto en modo boot como en modo ejecución. Se pone a 0 manualmente por programa, y en ese momento, ZX-Uno entra en modo ejecución.
- **DIVEN**: a 0 indica que DIVMMC no está habilitado en el sistema, aunque los puertos de acceso a la interfaz SPI del slot SD/MMC siguen estando disponibles. La memoria usada por DIVMMC queda disponible para otros usos. A 1 indica que DIVMMC está habilitada. Si se habilita, antes de pasar al modo ejecución hay que cargar una imagen de ESXDOS en el banco de RAM correspondiente. El valor por defecto de este bit es 0.
- **DISNMI**: a 1 indica que la función NMI de DIVMMC no estará disponible. NMI funcionará, pero no causará que ESXDOS se automapee, dejando así el control de NMI en manos de la ROM principal del sistema. Bit añadido para mejorar la compatibilidad de DIVMMC con SE Basic IV. Su valor por defecto es 0 (ESXDOS maneja los eventos NMI).
- **I2KB**: a 1 configura la ULA para que devuelva un valor acorde con un Spectrum issue 2 al leer el puerto de teclado (\$FE). A 0, el valor devuelto es compatible con issue 3 y posteriores. Por defecto su valor es 0.
- **MODE1, MODE0**: especifica el modo de timing de la ULA para acomodarse a diferentes modelos de Spectrum. 00 = ULA ZX Spectrum 48K PAL, 01 = ZX Spectrum 128K/+2 gris, 10 = Pentagon 128, 11 = 48K NTSC (262 scans).
- **DISCONT**: indica si se debe producir contienda de memoria en la memoria de video. 0 para habilitar contienda (compatibilidad con 48K y 128K). 1 para deshabilitar contienda (compatibilidad con Pentagon 128)
- **LOCK**: Cuando vale 1, impide que se puedan hacer más cambios a ciertos bits del registro MASTERCONF, e impide además acceder a la SPI Flash. Este bit se pone a 0 sólo mediante un reset maestro (Ctrl-Alt-BkSpace) o al apagar y encender el clon.

\$01 MASTERMAPPER

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

De este registro sólo se usan sus 5 bits inferiores (valores \$00 a \$1F). El valor guardado es el número de un banco de 16KB de SRAM que será paginado en las direcciones \$C000-\$FFFF durante el modo boot. Los valores de este registro no tienen ningún efecto cuando ZX-Uno está en modo ejecución. 32 valores diferentes para este registro permiten direccionar hasta 512KB de SRAM. Si ZX-Uno se amplía con más memoria, se usarán más bits en este registro. La máxima cantidad de memoria manejable es de 4MB.

\$02 FLASHSPI

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	00000000

Puerto de acceso al registro SPI conectado a la SPI Flash. Escribiendo un valor en este registro, se envía a la SPI Flash, si ésta está seleccionada. Leyendo un valor de este registro, se lee el último valor enviado por la SPI Flash, y además, la misma operación de lectura provoca que la SPI envíe un nuevo valor (que sería leído con la siguiente operación de lectura a este registro). Por esta razón, en operaciones de lectura de bloques, el primer byte leído con este puerto debe descartarse.

\$03 FLASHCS

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	00000001

Sólo se emplea el bit 0. El valor escrito en este registro determinado el estado de la línea CS de la Flash SPI (0 = Flash seleccionada, 1 = flash no seleccionada).

\$04 SCancode

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	No cambia

En lectura, permite obtener el valor del último scancode generado por el teclado. En escritura permite enviar comandos al teclado.

\$05 KEYSTAT

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura	No cambia	No cambia	No cambia

Varios bits que indican si hay o no una nueva tecla pulsada, o soltada, y si ésta es una tecla extendida o normal.

BSY	0	0	0	ERR	RLS	EXT	PEN
-----	---	---	---	-----	-----	-----	-----

- BSY: a 1 para indicar que hay una transmisión de datos al puerto PS/2 aún en proceso. Se debe esperar a que valga 0 para iniciar una nueva transmisión.
- ERR: a 1 para indicar que la última transmisión a o desde el puerto PS/2 tuvo errores.
- RLS: a 1 para indicar que el último evento pertenece a una tecla que ha sido soltada.
- EXT: a 1 para indicar que el último evento pertenece a una tecla con código extendido (E0+scancode)
- PEN: a 1 para indicar que hay un dato nuevo listo para ser leído en el registro SCANDCODE. Tras leer KEYSTAT, este bit pasa a valer 0.

\$06 JOYCONF

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	00100001

Los bits 0 a 2 indican el modo de funcionamiento del joystick mapeado en teclado (o del segundo joystick físico si hay splitter). Los bits 4 a 6 indican el modo de funcionamiento del joystick físico (conector DB-9 lateral). El bit 3 indica el autofuego del segundo joystick o keypad. El bit 7 indica el autofuego del joystick principal. Los valores son: 000 = Disabled, 001 = Kempston, 010 = Sinclair 1, 011 = Sinclair 2, 100 = Protek/Cursor/AGF, 101 = Fuller, 110 = OPQAspM, 111 = reservado.

\$07 KEYMAP

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	Ver descripción	Ver descripción	Ver descripción

En lectura, cada acceso proporciona el siguiente byte del mapa de teclado cargado en ZX-Uno actualmente. En escritura, el byte correspondiente al mapa de teclado marcado por la posición actual es modificado. En ambos casos, el puntero de direcciones se incrementa automáticamente para señalar al siguiente byte del mapa de teclado. Este puntero vuelve a 0 automáticamente tras un reset, una escritura en el registro \$FC3B, o cuando se termina el mapa de teclado. El mapa de teclado, en la implementación actual, ocupa 16384 bytes.

\$09 MOUSEDATA

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	No cambia

Registro de datos del puerto PS/2 del ratón. Usado para leer o enviar comandos directos al ratón PS/2. Por ejemplo: para inicializar el ratón, debe enviarse el valor \$F4 a este registro.

\$0A MOUSESTATUS

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	No cambia

Registro de estado del puerto PS/2 del ratón. Se definen los siguientes bits:

BSY	0	0	0	ERR	0	0	PEN
-----	---	---	---	-----	---	---	-----

- BSY: a 1 para indicar que hay una transmisión de datos al puerto PS/2 aún en proceso. Se debe esperar a que valga 0 para iniciar una nueva transmisión.
- ERR: a 1 para indicar que la última transmisión a o desde el puerto PS/2 tuvo errores.
- PEN: a 1 para indicar que hay un dato nuevo listo para ser leído en el registro MOUSEDATA. Tras leer MOUSESTATUS, este bit pasa a valer 0.

\$0B SCANBLCTRL

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	00000000

Registro de control del scandoubler y control de la velocidad del sistema. Se definen los siguientes bits:

TURBO	COPT	FREQ	ENSCAN	VGA
-------	------	------	--------	-----

- TURBO: 00 para seleccionar 3.5 MHz, 01 para seleccionar 7 MHz, 10 para seleccionar 14 MHz y 11 para seleccionar 28 MHz. Estos bits también son actualizados con el valor de los bits D0-D3 del puerto \$8E3B, usado en el ZX Prism para seleccionar las distintas velocidades para la CPU.
- COPT: selecciona la forma de generar los sincronismos compuestos para RGB y video compuesto: 0 para usar el tipo de sincronismo original del Spectrum. 1 para usar pulsos de sincronismo según el estándar PAL
- FREQ: estos tres bits definen la frecuencia del reloj maestro, del cual derivan el resto de relojes del circuito. Entre otras, se define también aquí la frecuencia de refresco vertical, que puede usarse para mejorar la compatibilidad con algunos monitores VGA que no soportan una frecuencia de refresco vertical de 50Hz. Los valores de refresco vertical son los siguientes:
 - 000 : 50Hz para modo 48K y Pentagon
 - 001 : 50Hz para modo 128K
 - 010 : 52 Hz
 - 011 : 53 Hz
 - 100 : 55 Hz
 - 101 : 57 Hz
 - 110 : 59 Hz
 - 111 : 60 Hz
- ENSCAN: a 1 para habilitar el efecto de scanlines en el modo VGA. No tiene efecto si el modo VGA está deshabilitado.
- VGA: a 1 para habilitar el scandoubler. La salida del scandoubler es la misma que la salida RGB normal, pero doblando la frecuencia de retrazo horizontal. A 0 para usar una salida de 15kHz RGB / video compuesto.

\$0C RASTERLINE

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	11111111	11111111	11111111

Almacena los 8 bits menos significativos de la línea de pantalla en la que se desea provocar un disparo de una interrupción enmascarable. Un valor 0 para este registro (con LINE8 también igual a 0) establece que la interrupción ráster se disparará, si está habilitada, justo al comenzar el borde derecho de la línea anterior a la primera línea de pantalla en la que comienza la zona de "paper". Dicho en otras palabras: el conteo de líneas de esta interrupción asume que una línea de pantalla se compone de: borde derecho + intervalo de blanking horizontal + borde izquierdo + zona de paper. Si se asume de esta forma, el disparo de la interrupción se haría al comienzo de la línea seleccionada. Un valor para RASTERLINE igual a 192 (con LINE8 igual a 0) dispara la interrupción ráster al comienzo del borde inferior. Los números de línea para el fin del borde inferior y comienzo del borde superior dependen de los timings empleados. El mayor valor posible en la práctica para RASTERLINE corresponde a una interrupción ráster disparada en la última línea del borde superior (ver RASTERCTRL)

\$0D RASTERCTRL

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	00000001	00000001	00000001

Registro de control y estado de la interrupción ráster. Se definen los siguientes bits.

INT	0	0	0	0	DISVINT	ENARINT	LINE8
-----	---	---	---	---	---------	---------	-------

- INT: este bit sólo está disponible en lectura. Vale 1 durante 32 ciclos de reloj a partir del momento en que se dispara la interrupción ráster. Este bit está disponible aunque el procesador tenga las interrupciones deshabilitadas. No está disponible si el bit ENARINT vale 0.
- DISVINT: a 1 para deshabilitar las interrupciones enmascarables por retrazo vertical (las originales de la ULA). Tras un reset, este bit vale 0.
- ENARINT: a 1 para habilitar las interrupciones enmascarables por línea ráster. Tras un reset, este bit vale 0.
- LINE8: guarda el bit 8 del valor de RASTERLINE, para poder definir cualquier valor entre 0 y 511, aunque en la práctica, el mayor valor está limitado por el número de líneas generadas por la ULA (311 en modo 48K, 310 en modo 128K, 319 en modo Pentagon). Si se establece un número de línea superior al límite, la interrupción ráster no se producirá.

\$0E DEVCONTROL

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

Registro de habilitacion/deshabilitacion de distintas caracteristicas. Se definen los siguientes bits.

DISD	ENMMU	DIROMSEL 1F	DIROMSEL 7F	DI1FFD	DI7FFD	DITAY	DIAY
------	-------	----------------	----------------	--------	--------	-------	------

- DISD: a 1 para deshabilitar la interfaz hardware SPI para SD (usada en DivMMC y ZXMMC). Deshabilitando esta interfaz se liberan los puertos \$1F en escritura, \$3F, \$E7 y \$EB. Tras reset maestro, vale 0.
- ENMMU: a 1 para habilitar la MMU horizontal usada en el Timex Sinclair. Habilitando esta interfaz se usa el bit 7 del puerto \$FF, se usa el puerto \$F4 y una lectura al puerto \$FF devuelve el ultimo valor escrito en él. Tras reset maestro, vale 0.
- DIROM1F: el valor de este bit se enmascara con el valor del bit 2 del puerto \$1FFD según la operación $\sim\text{DIROM1F} \& \$1FFD[2]$. El resultado neto es que si este bit vale 1, el sistema trabajará como si el valor del bit 2 del puerto \$1FFD fuera siempre 0, independientemente del valor que se escriba en él. Tras reset maestro, vale 0, que permite que los cambios al bit 2 de \$1FFD sean tenidos en cuenta.
- DIROM7F: el valor de este bit se enmascara con el valor del bit 4 del puerto \$7FFD según la operación $\sim\text{DIROM7F} \& \$7FFD[4]$. El resultado neto es que si este bit vale 1, el sistema trabajará como si el valor del bit 4 del puerto \$7FFD fuera siempre 0, independientemente del valor que se escriba en él. Tras reset maestro, vale 0, que permite que los cambios al bit 4 de \$7FFD sean tenidos en cuenta.
- DI1FFD: a 1 para deshabilitar el sistema de paginación compatible con +2A/+3. Deshabilitando esta interfaz se libera el puerto \$1FFD en escritura. Tenga en cuenta que la decodificación del puerto \$7FFD, si está activa, es diferente dependiendo de si el puerto \$1FFD está activo o no.
- DI7FFD: a 1 para deshabilitar el sistema de paginación compatible con 128K. La deshabilitación de este sistema implica la deshabilitación del sistema de paginación del +2A/+3 aunque no se haya desactivado explicitamente.
- DITAY: a 1 para deshabilitar el segundo chip AY, deshabilitando así el modo Turbo Sound.
- DIAY: a 1 para deshabilitar el chip principal AY. La deshabilitación de este chip implica la deshabilitación del segundo chip AY, aunque no se haya deshabilitado explicitamente.

\$0F DEVCTRL2

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

| Registro de habilitacion/deshabilitacion de distintas caracteristicas (continuación de DEVCONTROL). Se definen los siguientes bits.

Resv	Resv	SPLITTER	DIMIXER	DISPECDRUM	DIRADAS	DITIMEX	DIULAPLUS
------	------	----------	---------	------------	---------	---------	-----------

- Resv: Este bit está reservado. En la implementación actual, debe escribirse un 0 en él si se actualiza el valor del registro.
- SPLITTER: a 1 para habilitar el splitter de joystick.
- DIMIXER: a 1 se deshabilita la salida de sonido.
- DISPECDRUM: a 1 se deshabilita el soporte de SpecDrum/Covox.
- DIRADAS: a 1 para deshabilitar el modo radastaniano. Tenga en cuenta que si el modo radastaniano no se deshabilita, pero se deshabilita la ULAPLUS, al intentar usar el modo radastaniano, el datapath usado en la ULA no será el esperado y el comportamiento de la pantalla en este caso no está documentado.
- DITIMEX: a 1 para deshabilitar los modos de pantalla compatible Timex. Cualquier escritura al puerto \$FF es por tanto ignorada. Si la MMU del Timex está habilitada, una lectura al puerto \$FF devolverá 0.
- DIULAPLUS: a 1 para deshabilitar la ULAPLUS. Cualquier escritura a los puertos de ULAPLUS se ignora. Las lecturas a dichos puertos devuelven el valor del bus flotante. No obstante tenga en cuenta que el mecanismo de contención para este puerto sigue funcionando aunque esté deshabilitado.

\$10 MEMREPORT

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$40 RADASCTRL

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	00000000	00000000	00000000

Registro para establecer el modo radastaniano y establecer sus características. Las escrituras a este registro son ignoradas si el bit correspondiente en DEVCTRL2 está activo. Se definen los siguientes bits

Resv	Resv	Resv	Resv	Resv	Resv	EN1	EN0
------	------	------	------	------	------	-----	-----

- Resv: Este bit está reservado. En la implementación actual, debe escribirse un 0 en él si se actualiza el valor del registro.
- EN1, EN0: ambos bits deben valer 1 para habilitar el modo radastaniano. Si EN0 está habilitado, pero EN1 está deshabilitado, el resto de bits, del 2 al 7, está definido según lo implementado en el [emulador ZEsarUX](#) (consultar la documentación de César Hernández sobre esto).

\$41 RADASOFFSET

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	00000000	00000000	00000000

Contiene el número de bytes que hay que añadir a la dirección base de la pantalla (4000h, 6000h, C000h o E000h según la configuración) para obtener la dirección donde se encuentran los dos primeros píxeles en modo radastaniano. Es un registro de 14 bits. Para escribir un valor, primero se escriben los 8 bits menos significativos, seguidos inmediatamente de los 8 bits más significativos (los 2 bits más significativos de este valor se ignoran). Si el valor de offset es tal que hace que al explorar la memoria de pantalla para crear la imagen se llegue al final de una página de 16KB, la exploración continuará al principio de esa misma página.

\$42 RADASPADDING

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	00000000	00000000	00000000

Contiene el número de bytes - 64 que ocupa un scanline en modo radastaniano. Es decir, si este registro vale 0, la longitud en bytes de un scanline es de 64 bytes (128 píxeles). Si este registro vale 4, la longitud de un scanline es de 68 bytes (136 píxeles). Si vale 255, la longitud de un scanline es de 64+255=319 bytes, o 638 píxeles. Si el valor de offset es tal que hace que al explorar la memoria de pantalla para crear la imagen se llegue al final de una página de 16KB, la exploración continuará al principio de esa misma página.

\$43 RADASPALBANK

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	00000000	00000000	00000000

Registro para establecer qué sección de la paleta de ULAplus se usará para definir los colores en el modo radastaniano, y cómo se comportará el borde.

Resv	Resv	Resv	Resv	Resv	BOR3	RADPALQUA RTER
------	------	------	------	------	------	----------------

- Resv: Este bit está reservado. En la implementación actual, debe escribirse un 0 en él si se actualiza el valor del registro.
- BOR3: dentro de la paleta actual seleccionada, indica si el color del borde se tomará de las entradas 0 a 7 (0) o de las entradas 8 a 15 (1). Puede considerarse como el bit 3 del color del borde en modo radastaniano.
- RADPALQUARTER: dos bits que indican qué sección de la paleta de ULAplus se va a usar para el modo radastaniano. 00 para usar la sección de entradas 0 a 15. 01 para las entradas 16 a 31, 10 para las entradas 32 a 47 y 11 para usar las entradas 48 a 63.

\$80 HOFFS48K

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	\$38

Valor de ajuste de centrado horizontal de la pantalla para la ULA de 48K.

\$81 VOFFS48K

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	\$01

Valor de ajuste de centrado vertical de la pantalla para la ULA de 48K.

\$82 HOFFS128K

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	\$3A

Valor de ajuste de centrado horizontal de la pantalla para la ULA de 128K.

\$83 VOFFS128K

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	\$01

Valor de ajuste de centrado vertical de la pantalla para la ULA de 128K.

\$84

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
HOFFSPEN	Lectura/Escritura	No cambia	No cambia

Valor de ajuste de centrado horizontal de la pantalla para la ULA de Pentagon.

\$85 VOFFSPEN

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	\$00

Valor de ajuste de centrado vertical de la pantalla para la ULA de Pentagon.

\$A0 DMACTRL

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$A1 DMASRC

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$A2 DMADST

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$A3 DMAPRE

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$A4 DMALEN

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$A5 DMAPROB

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$A6 DMASTAT

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$C6 UARTRDATA

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$C7 UARTRSTAT

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$C8 - \$DF RESERVED

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	Definido por el usuario	Definido por el usuario	Definido por el usuario

Rango de registros de ZXUNO reservados para experimentos o uso privado.

\$F0 SRAMADDR

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$F1 MADDRINC

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000 ===== \$F2 SRAMDATA [align="center",options="header"]

| Sentido | Valor tras reset de usuario | Valor tras reset maestro | Valor tras poweron
| Lectura/Escritura | No cambia | 00000000 | 00000000

\$F3 VDECKCTRL

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$F7 AUDIOMIX

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	10011111

Control de la mezcla de los canales izquierdo/derecho partiendo de los 4 canales del AY-8912. Se definen los siguientes grupos de 2 bits cada uno:

CHANNELA	CHANNELB	CHANNELC	BEEPDRUM
----------	----------	----------	----------

- BEEPDRUM : Canal para Beeper y Specdrum.
- El significado de cada grupo de 2 bits es: 00 = Silencio, 01 = Canal derecho, 10 = Canal izquierdo, 11 = Ambos

\$FB AD724

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	00000000

Control del modo de funcionamiento del chip encoder AD724. Se definen los siguientes bits:

Resv	MODE						
------	------	------	------	------	------	------	------

- Resv : el significado de estos bits está reservado y no debe alterarse.
- MODE : modo de funcionamiento del AD724. 0 = codifica norma PAL. 1 = codifica norma NTSC

\$FC COREADDR

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	No cambia	\$058000

Almacena la dirección, dentro de la memoria SPI, de inicio del core a arrancar. Para almacenar una dirección diferente, se han de realizar tres escrituras a este registro, conteniendo los tres bytes de la dirección, en orden de más a o menos significativo. En lectura, cada acceso a este registro devuelve una parte de la última dirección almacenada, de la parte más significativa a la menos significativa.

\$FD COREBOOT

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Escritura	No cambia	No cambia	No cambia

Registro de control de arranque. Escribiendo un 1 en el bit 0 de este registro (el resto de bits están reservados y deben quedarse a 0) hace que se desencadene el mecanismo interno de la FPGA que permite arrancar otro core. La dirección de comienzo de este segundo core será la última que se escribiera usando el registro COREADDR.

\$FE SCRATCH

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura/Escritura	No cambia	00000000	00000000

\$FF COREID

Sentido	Valor tras reset de usuario	Valor tras reset maestro	Valor tras poweron
Lectura	Ver descripción	Ver descripción	Ver descripción

Cada operación de lectura proporciona el siguiente carácter ASCII de la cadena que contiene la revisión actual del core del ZX-Uno. Cuando la cadena termina, lecturas posteriores emiten bytes con el valor 0 (al menos se emite uno de ellos) hasta que vuelve a comenzar la cadena. Este puntero vuelve a 0 automáticamente tras un reset o una escritura en el registro **\$FC3B**. Los caracteres entregados que forman parte de la cadena son ASCII estándar imprimibles (códigos [32-127](#)). Cualquier otro valor es indicativo de que este registro no está operativo.

Enlaces

[ZX-Uno](#)

[ZX-Uno FAQ](#)

[Guía rápida del ZX-Uno](#)

[Core ZX Spectrum](#)

[The ZX Spectrum +3e Homepage](#)

[Sharing a +3e disk with PC \(FAT\) partitions](#)

[Layouts de teclado](#)

[Firmware de teclado para ZX Go+](#)

[zxunops2](#)

[Almost \(In-\) Complete List of esxDOS DOT-Commands](#)

[Cargando Leches 2.0](#)

[WiFi \(RetroWiki\)](#)

[WiFi en ZX-Uno](#)

[Core de ZX-Uno Test UART \(WiFi\)](#)

[Network tools for ZX-Uno pack](#)

[ESP8266 AT Instruction Set](#)

[Vídeos Radastanianos](#)

[Nuevo core zx48](#)

[ZX 48 para ZX DOS+ \(Kyp\)](#)

[Core ZXNEXT en ZX DOS](#)

[ZX Spectrum Next en ZX DOS](#)

[Core MSX](#)

[MSX1FPGA](#)

[MSX Pack](#)

[Nextor para MSX](#)

[Nextor User Manual](#)

[MSX-DOS](#)

[Atom Software Archive en carpeta ATOM](#)

[Teclado Core Atom](#)

[Core de NES para ZX-Uno](#)

[ColecoFPGA en GitHub](#)

[Core de Videopac para ZX DOS](#)

[VideoPac-ZX DOS](#)

[TOSEC: Magnavox Odyssey 2 \(2012-04-23\)](#)

[Videopac G7000 / Odyssey2 for FPGA](#)

[Odyssey Speech And Sound Effects Module Manual](#)

[Maxduino - guía de uso](#)

[MakeTSX](#)

[RetroConverter](#)

[Lince Script](#)

[Hardware files to support STM32 based boards on Arduino version 1.8.x](#)

[Programming a Spartan 6 with a Raspberry Pi](#)

[Tutorial para desbriquiar el ZX-Uno con una Raspberry](#)

[Como programar un UnAmiga con la Raspberry Pi \(o Linux\) con el USB-Blaster y OpenOCD](#)