

# ZXOS+

## Manual



# Manual de ZXDOS+ y gomaDOS+

kounch

Version 1.1.6

# Índice

Introducción .....	1
Puertos y Conectores .....	2
ZXDOS+ .....	2
gomaDOS+ .....	3
Descripción .....	3
Configuración Inicial .....	4
Formato de la tarjeta microSD .....	4
Windows .....	4
MacOS .....	4
Linux .....	5
esxdos .....	6
Modos de teclado en gomaDOS+ .....	8
BIOS .....	11
Main .....	11
ROMs .....	12
Upgrade .....	12
Boot .....	13
Advanced .....	14
Exit .....	15
ZX Spectrum .....	16
Teclado .....	17
Español .....	17
Inglés .....	17
Spectrum .....	17
Teclas especiales y botones .....	18
ROMs .....	19
esxdos .....	20
Guía básica .....	20
Comandos para ZXDOS+ .....	22
Wi-Fi .....	23
Network tools for ZX-Uno pack .....	23
FTP-Uno .....	23
UART Terminal .....	24
Creación de vídeos RDM (RaDastan Movie) .....	25
Actualizaciones .....	26
BIOS .....	26
ROMs .....	26
Cores .....	26

esxdos .....	27
Otros cores .....	28
ZX Spectrum Next .....	28
Formato de Tarjeta microSD .....	29
Teclado .....	29
Teclas especiales y botones .....	29
Guía básica .....	30
MSX .....	32
Formato de Tarjeta microSD .....	32
Teclado .....	33
Teclas especiales y botones .....	33
Guía básica .....	34
MSXCTRL .....	35
Otros .....	35
Amstrad CPC .....	36
Formato de Tarjeta microSD .....	36
Teclado .....	36
Teclas especiales y botones .....	36
Guía básica .....	37
Acorn Atom .....	38
Formato de Tarjeta microSD .....	38
Teclado .....	39
Teclas especiales y botones .....	39
Guía básica .....	40
Commodore 64 .....	41
Formato de Tarjeta microSD .....	41
Teclado .....	41
Teclas especiales y botones .....	41
Guía básica .....	42
Phoenix .....	43
Formato de Tarjeta microSD .....	43
Teclado .....	43
Teclas especiales y botones .....	43
Guía básica .....	43
Pong .....	44
Formato de Tarjeta microSD .....	44
Teclado .....	44
Teclas especiales y botones .....	44
Guía básica .....	45
NES .....	46
Formato de Tarjeta microSD .....	46

Teclado .....	46
Teclas especiales y botones .....	46
Guía básica .....	47
Colecovision .....	48
Formato de Tarjeta microSD .....	48
Teclado .....	48
Teclas especiales y botones .....	48
Guía básica .....	49
Solución de problemas .....	50
Recuperación del firmware .....	50
Recuperación usando una Raspberry Pi .....	50
Referencias .....	57

# 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.

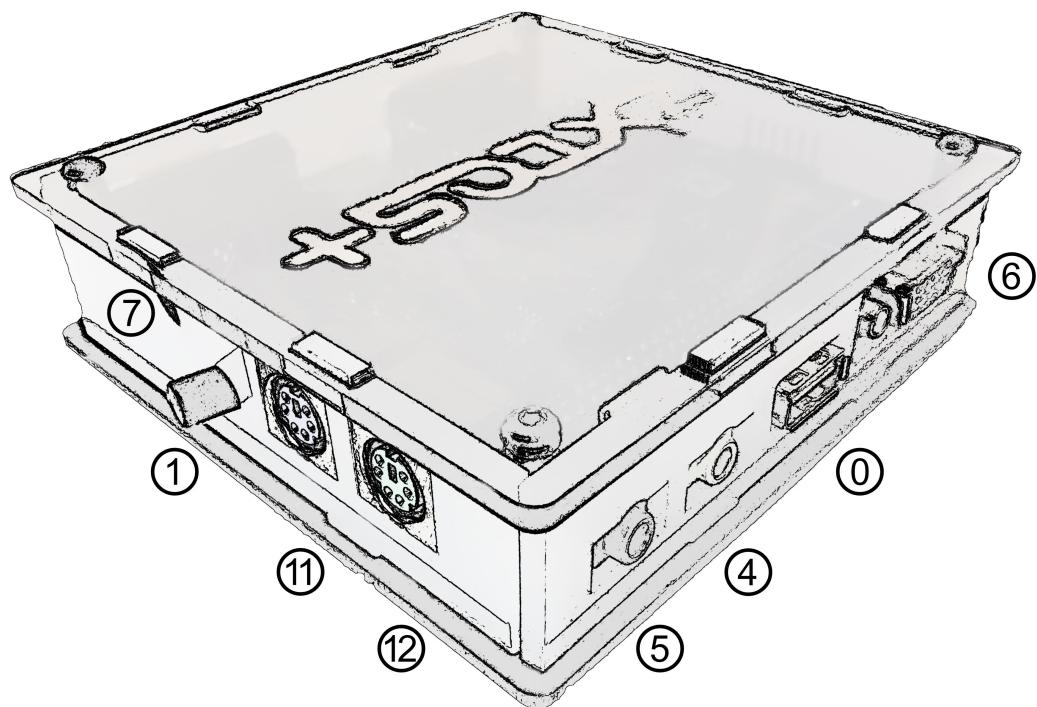
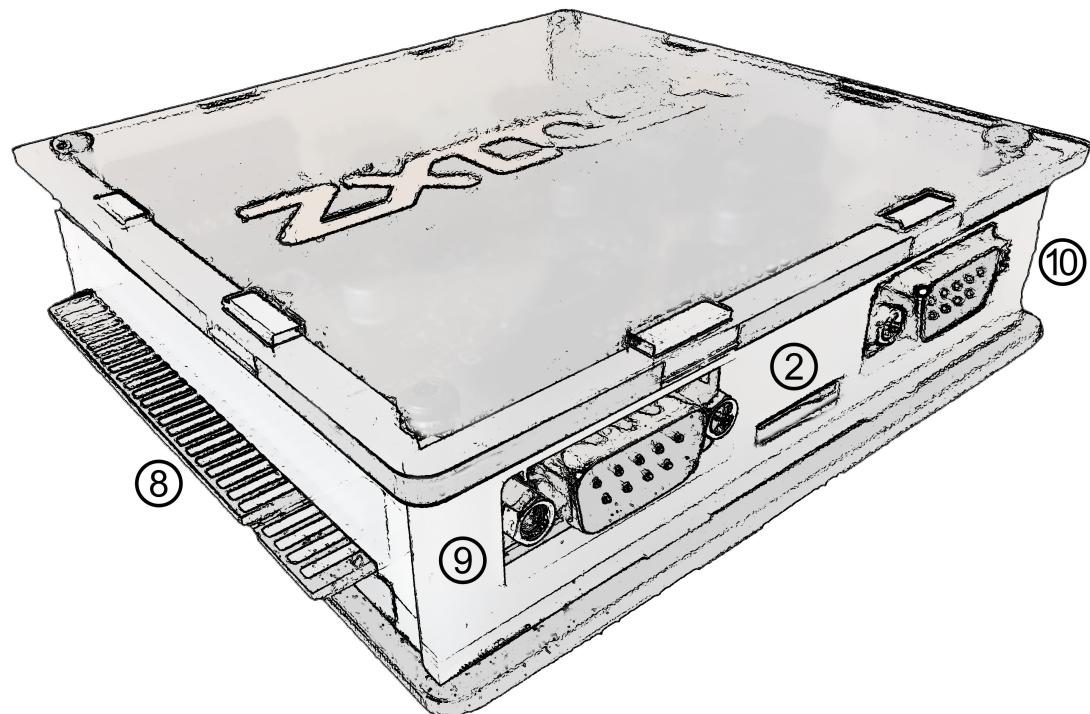
Con el paso del tiempo, el proyecto ha ido creciendo, de forma que es posible instalar distintas configuraciones de software (cores) en la memoria flash de la FPGA, y que trabajan como otros sistemas distintos del ZX Spectrum, pudiendo elegir arrancar el ZXDOS+ con la configuración que se desee de entre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

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

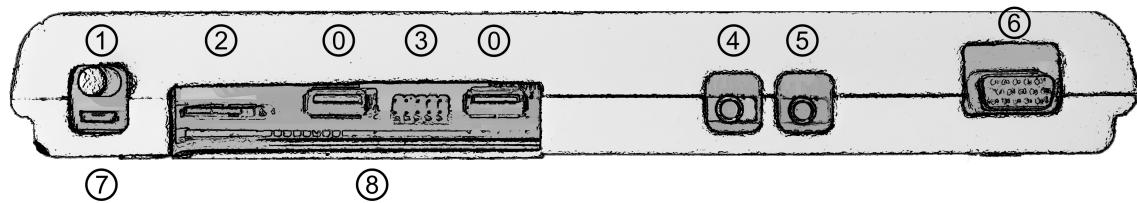
La mayoría de las funciones y características de ZXDOS+ y gomaDOS+ son las mismas, así que, en este documento, se hablará, en general, de ZXDOS+, indicando las diferencias con gomaDOS+ donde sea necesario.

# 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, no necesariamente muy grande
- 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+)

## Formato de la tarjeta microSD

Para poder utilizar una tarjeta microSD, esta debe tener, al menos, una partición (la primera en el caso de haber varias) en formato FAT16 o FAT32 (según el caso, se recomienda uno u otro formato para compatibilidad con distintos cores de terceros).



El tamaño máximo de una partición FAT16 son 4GB

### Windows

Para configuraciones sencillas, y tarjetas del tamaño adecuado (menos de 2GB para FAT16 o menos de 32GB 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.

### MacOS

Para configuraciones sencillas, y tarjetas del tamaño adecuado (menos de 2GB para FAT16 o menos de 32GB 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" ZXDOSPLUS 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" ZXDOSPLUS 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% ZXDOSPLUS 4G %DOS_FAT_16% EXTRA 4G
"MS-DOS FAT32" DATA R
sudo newfs_msdos -F 16 -v ZXDOSPLUS -b 4096 -c 128 /dev/rdisk6s1
sudo newfs_msdos -F 16 -v EXTRA -b 4096 -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) 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
```

## Linux

Existen multitud de herramientas en Linux que permiten formatear y particionar el contenido de una tarjeta SD (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.

## 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.



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](#) y [aquí](#)), y son los siguientes:

```
back16m  
back32m  
corebios  
dmaplayw  
esprst  
iwconfig  
joyconf  
keymap  
loadpxz  
playmid  
playrmov  
romsback  
romsupgr  
upgr16m  
upgr32m  
zxuc  
zxunocfg
```

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

# Modos de teclado en gomaDOS+

El teclado de gomaDOS+ al ser similar al teclado del ZX Spectrum original, carece de algunas de las 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	<b>0</b>
Amstrad CPC	<b>1</b>
MSX	<b>2</b>
Commodore 64	<b>3</b>
Atari 800XL	<b>4</b>
BBC Micro	<b>5</b>
Acorn Electron	<b>6</b>
Apple (I y II)	<b>7</b>
Commodore VIC 20	<b>8</b>
PC XT	<b>9</b>
Oric Atmos	<b>A</b>
SAM Coupé	<b>B</b>
Jupiter ACE	<b>C</b>

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 video compuesto 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:

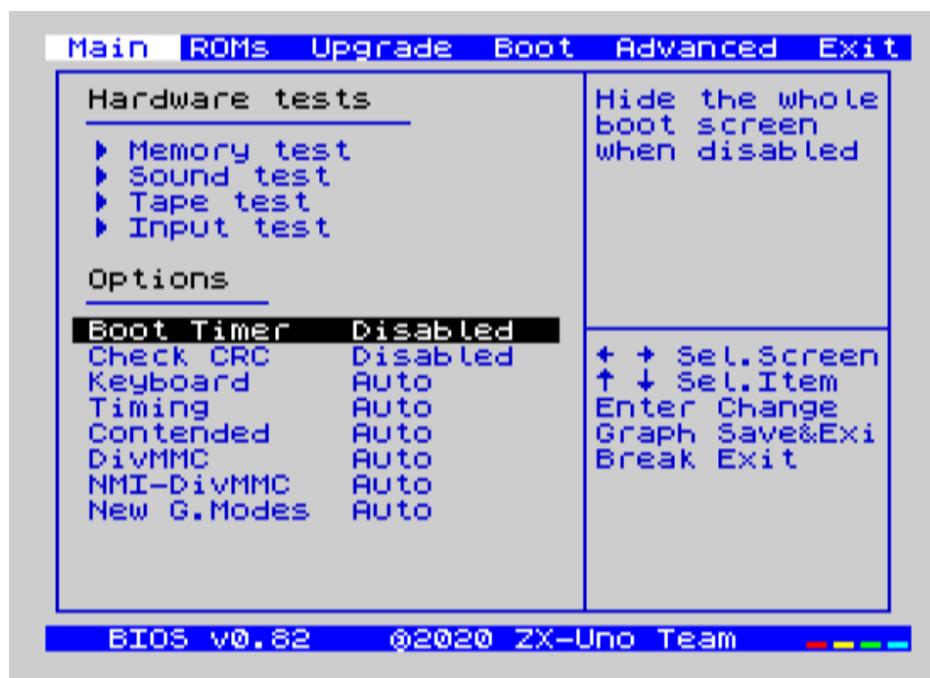
Caps S.+Symbol S.	Modo	Acción
1	Todos	F1
2	Todos	F2
3	Todos	F3
4	Todos	F4
5	Todos	F5
6	Todos	F6
7	Todos	F7
8	Todos	F8
9	Todos	F9
0	Todos	F10
Q	Todos	F11
W	Todos	F12
S	C64	Ctrl+F12
E	Acorn/CPC	Re Pág
R	Acorn	PgAbajo
U	Todos	Modo
G	ZX/MSX/C64	B1qDs
X	Todos	Guarda
C	PC	OPQA
V	Todos	Versión
B	ZX	Ctrl+Alt+Bcksp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

# 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 ZXDOS+. 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.

## Main



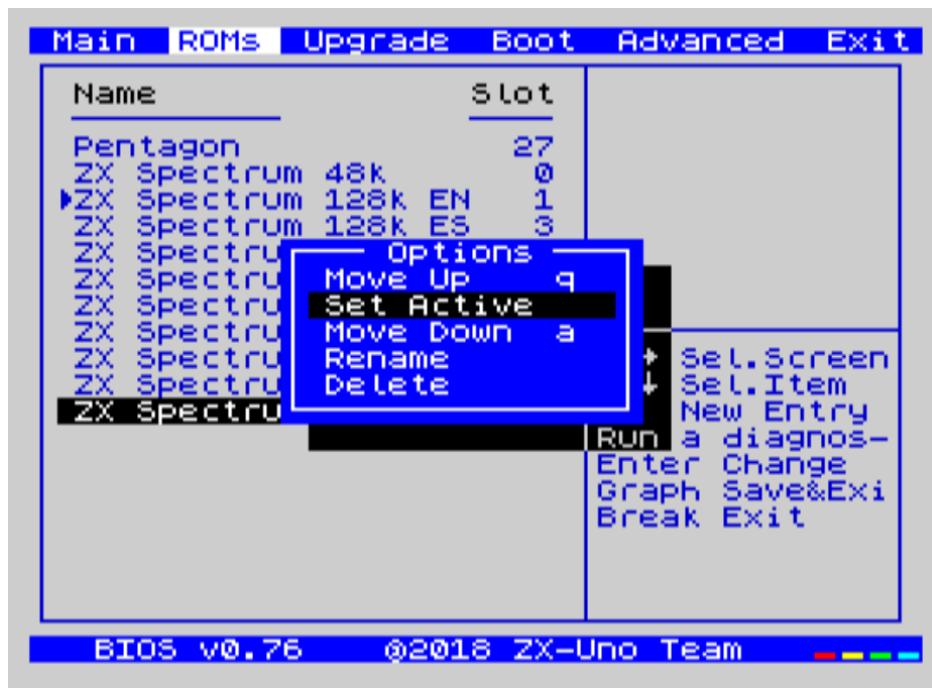
En la primera pantalla de configuración, además de poder ejecutar distintas pruebas, se puede definir el comportamiento por defecto para lo siguiente:

- 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).

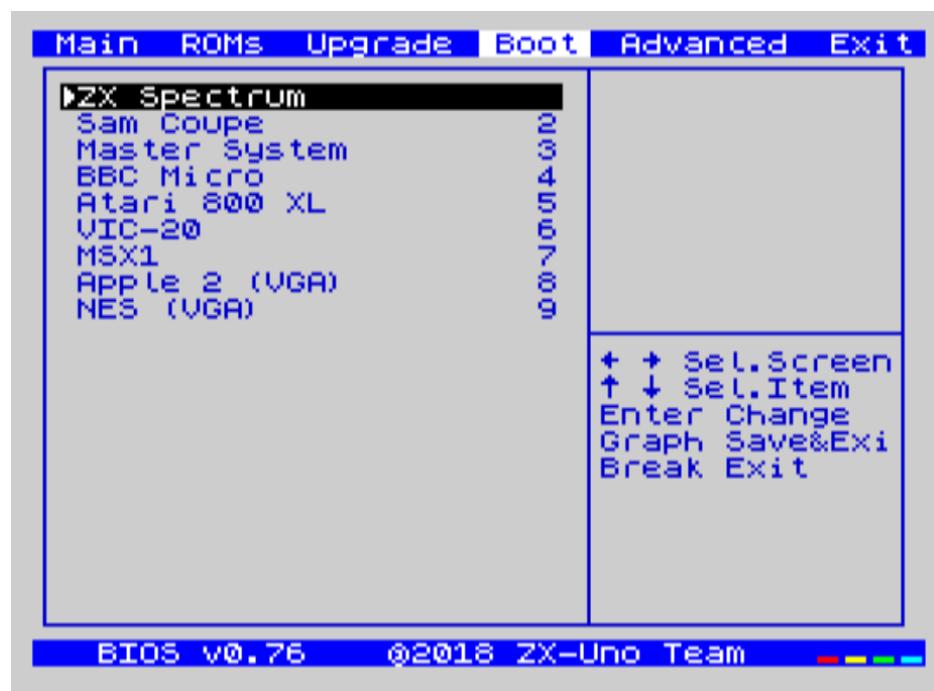
## Upgrade



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

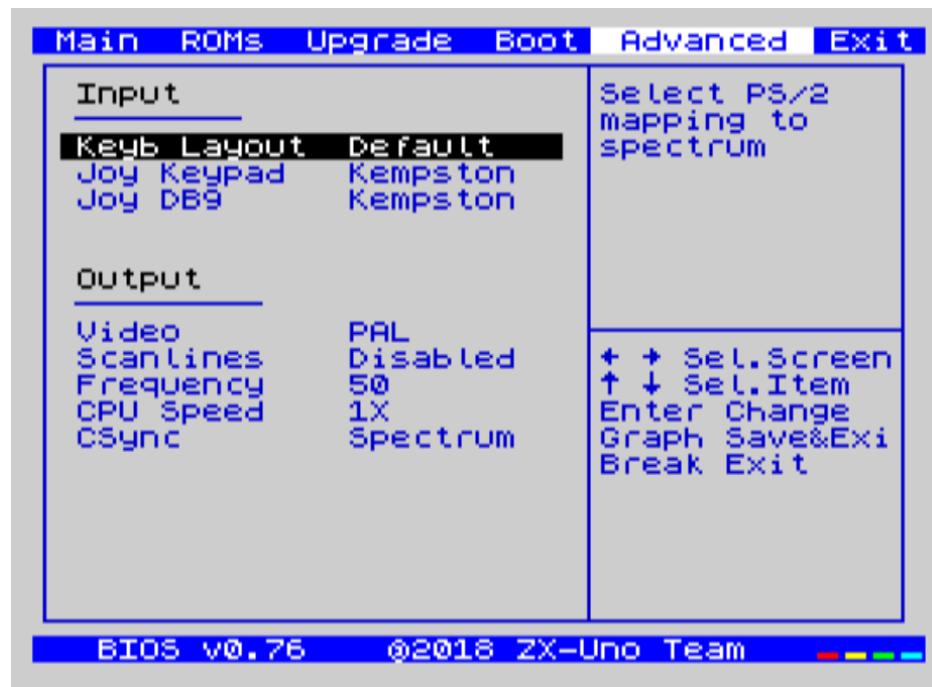
más información).

## Boot



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

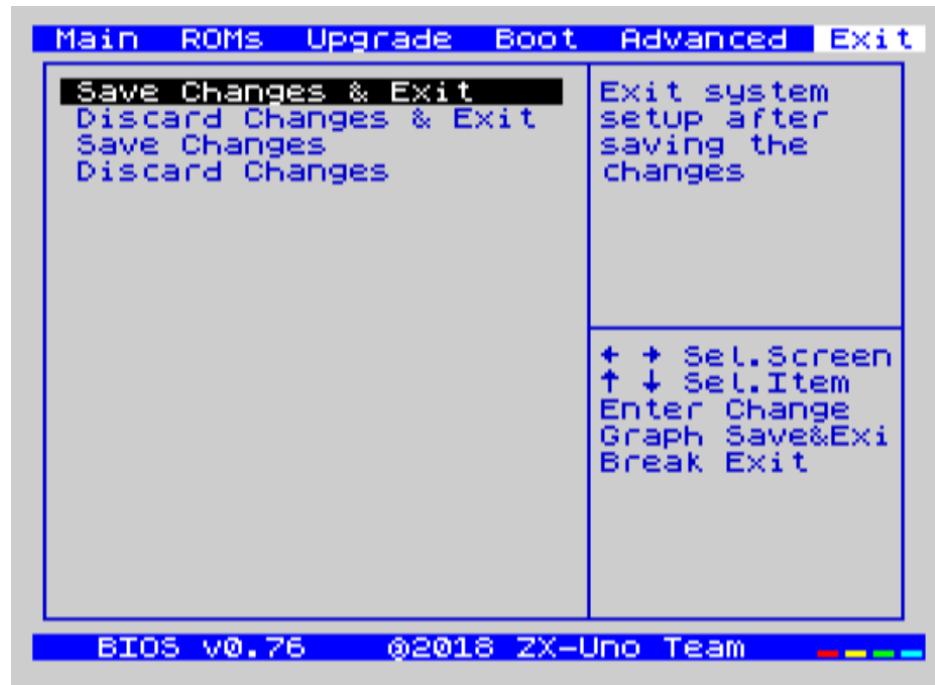
## Advanced



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 para +3e y soporte DIVMMC para 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 por el usuario desde el propio Spectrum.
- Soporte de ratón PS/2 emulando el protocolo Kempston Mouse.
- Posibilidad de salida de video en modo de video compuesto, RGB 15kHz, o VGA.
- Frecuencia de refresco vertical seleccionable por el usuario para mejorar 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.

# Teclado

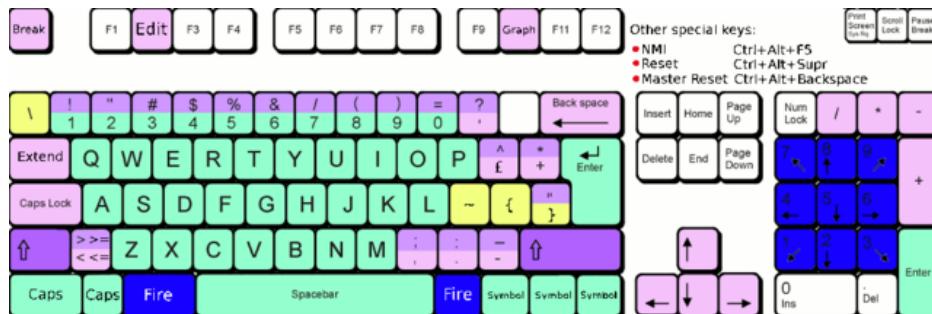
El mapa de teclado (asignación de las teclas físicas del teclado con las pulsaciones que se presentan a los distintos cores) se cambia desde el menú **Advanced** de la BIOS. Existen tres mapas distintos a elegir: Español (por defecto), inglés, y Spectrum (avanzado).

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.

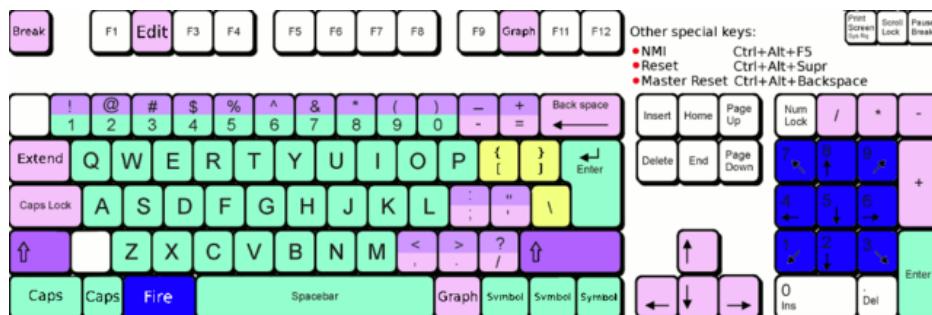
Para que el mapa se conserve después de un master reset, hay que tener seleccionado **Default** en la configuración de BIOS.

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

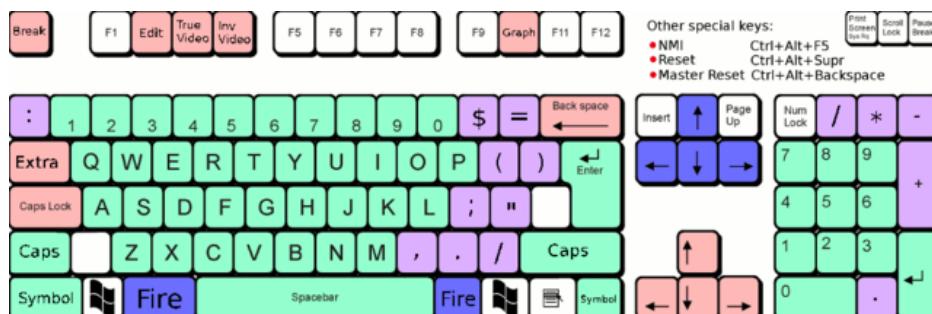
## Español



## 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+): Menú de selección de cores
- **Esc** (**Caps Shift+Espacio** en gomaDOS+): 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): Carga la ROM 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
- **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 video compuesto a VGA y viceversa.

## 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.

# 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 SD.
- **cp**: Copiar un archivo.
- **divideo**: Reproduce un archivo de video 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.

## 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 SD 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.
- **back32m**: Genera un fichero **FLASH.ZX2** en el directorio raíz de la tarjeta SD 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.
- **playmid**: Reproduce archivos musicales .MID en el addon MIDI.
- **playrmov**: Reproduce videos en **formato radastaniano (ficheros .RDM)**. Este comando no funciona en modo 48K.
- **romsback**: Copia a un fichero **ROMS.ZX1** en el directorio raíz de la tarjeta microSD todas las ROMS del core ZX Spectrum almacenadas en la memoria SPI Flash. Se debe ejecutar desde una ROM en modo "root".
- **romsupgr**: Copia el contenido de un fichero **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 SD a una memoria SPI Flash de 16 megas. Se debe ejecutar desde una ROM en modo "root".
- **upgr32m**: Copia el contenido de un fichero **FLASH.ZX2** 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.
- **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.

# 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
```

## Network tools for ZX-Uno pack

Se trata de un conjunto de programas, desarrollados por Nihirash y que se pueden [descargar de su web](#).

- [netman](#): Utilidad sencilla para configurar la conexión Wi-Fi para el resto de programas. No funciona en modo 48K.
- [uGophy](#): Cliente de [Gopher](#). No funciona en modo 48K.
- [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 la web de [IRATA.ONLINE](#).

## FTP-Uno

Cliente de FTP desarrollado 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 ZX DOS+ y cargar el archivo de cinta [ftpUno.tap](#)

## UART Terminal

Se trata de un programa de ejemplo 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 videos en formato radastaniano. Para poder convertir nuestros propios videos, 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 video (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.

# 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

Para actualizar las ROM instaladas para ZX Spectrum se ha de obtener un fichero con el nombre **ROMS.ZX1**, y se tiene que copiar en la tarjeta MicroSD. Arrancar el ZX DOS+ usando una ROM "rooted", 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 ROM en un fichero **ROMS.ZX1**), se puede usar el comando **.romsback**.

Los ficheros **ROMS.ZX1** 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 slots disponibles para almacenar cores (el número depende del tamaño de la SPI Flash del modelo de ZX DOS), estando reservado el primer slot para el de ZX Spectrum principal (esto no impide tener más cores de ZX Spectrum en otros slot 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 ZX DOS+ con placa FPGA LX16) o **COREnn.ZXD** (para un ZX DOS+ con placa FPGA LX25), donde **nn** es el número de slot donde realizar la instalación (por

ejemplo **CORE2.ZX2** o **CORE2.ZXD** para el slot 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**.

## 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, renombrándolo como **ESXDOS.ZX2** (para un ZX DOS+ con placa FPGA LX16) o **ESXDOS.ZXD** (para un ZX DOS+ con placa FPGA LX25).

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.

# Otros cores

## 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 ZX DOS+.

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

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

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

## 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

## 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.).



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 Mount  
Drive M K dir Rename Copy Erase Unmount



En el momento de escribir estas líneas, el core de ZX Spectrum Next para ZX DOS+ no soporta el uso del acelerador basado en Raspberry Pi, así que no es posible 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)

## Formato de Tarjeta microSD

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

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**.

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

El core para ZX DOS+ de Amstrad CPC 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, de 4GB de tamaño 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:

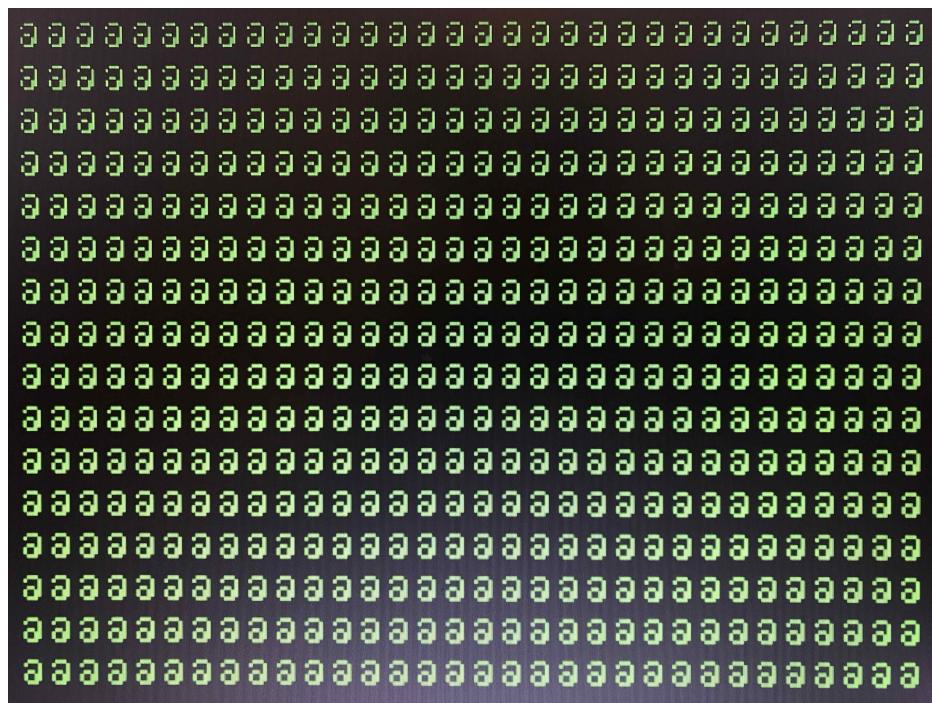
- **Mayús+F10**: Muestra el menú de Atom Software Archive
- **F10 (Caps Shift+Symbol Shift+0 en gomaDOS+)**: Soft Reset
- **F1 (Caps Shift+Symbol Shift+1 en gomaDOS+)**: Modo turbo 1Mhz
- **F2 (Caps Shift+Symbol Shift+2 en gomaDOS+)**: Modo turbo 2Mhz
- **F3 (Caps Shift+Symbol Shift+3 en gomaDOS+)**: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

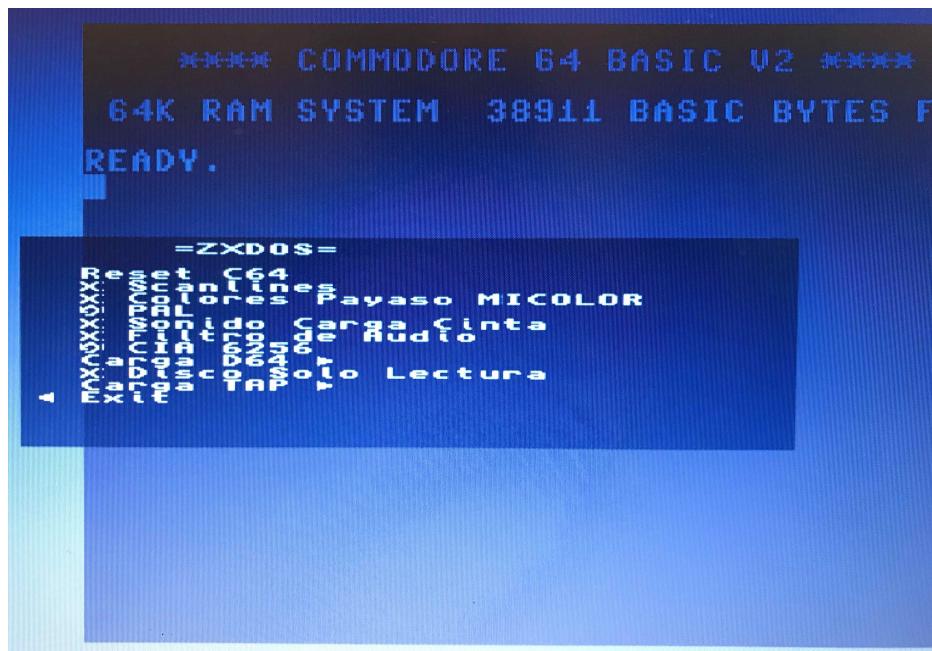
Las combinaciones específicas de gomaDOS+ que se indican a continuación se corresponden con el modo de teclado [Commodore 64](#). Véase el [apartado dedicado a los modos de teclado](#) de gomaDOS+ para más información. También se pueden utilizar las equivalentes en el modo de teclado [PC XT](#).

Durante la ejecución del core:

- **F12** ([Caps Shift+Symbol Shift+W](#) en gomaDOS+): Muestra menú de opciones
- **Bloq. Despl.** ([Caps Shift+Symbol Shift+G](#) en gomaDOS+): cambia de modo video compuesto 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 video compuesto 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 cinta, se puede escribir **LOAD** y pulsar **Enter**, o bien pulsar **Mayús+Esc** (**Mayús+RUN/STOP**).

# 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)

## Formato de Tarjeta microSD

Este core no utiliza la tarjeta microSD.

## Teclado

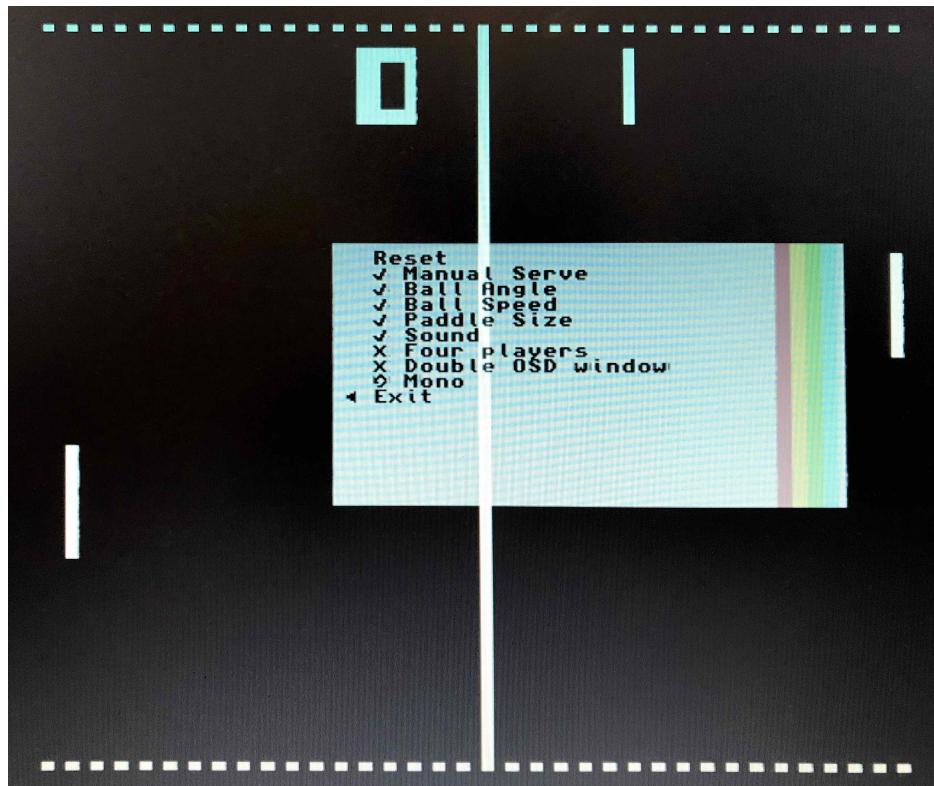
### Teclas especiales y botones

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 video compuesto 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 manual (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)
- Duplicar el tamaño del menún (Double OSD Window)
- Salir del menú (Exit)

# 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 SD
- 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 un ROM para cargar desde la SD (Load ROM)
- Salir del menú (Exit)

# 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

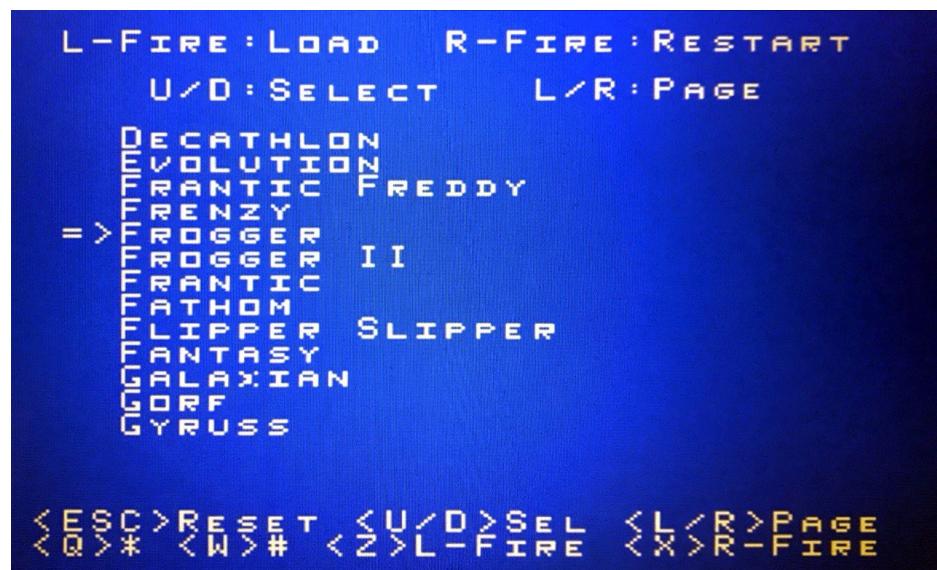
### Teclas especiales y botones

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 SD, 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.

# Solución de problemas

## 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.

### 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:

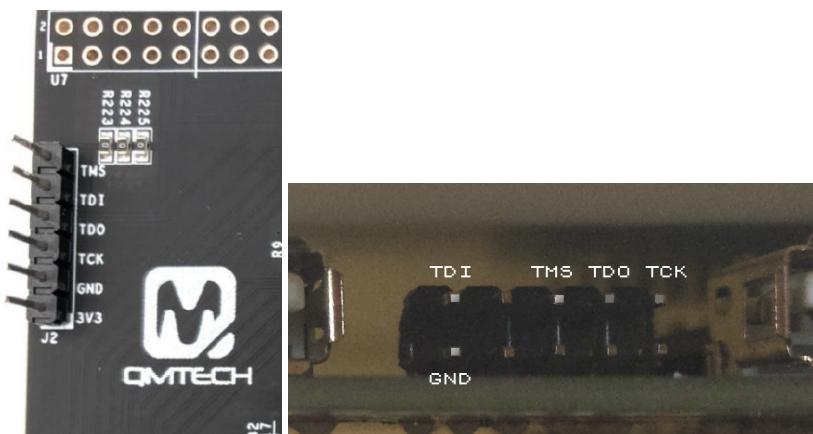
- Imagen Flash y recovery para ZX DOS+ (LX25), del [repositorio oficial, en este enlace](#)

## 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
```

3. Conectar el USB Blaster o los cables puente para GPIO. 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 **GPIO** de la Raspberry Pi.

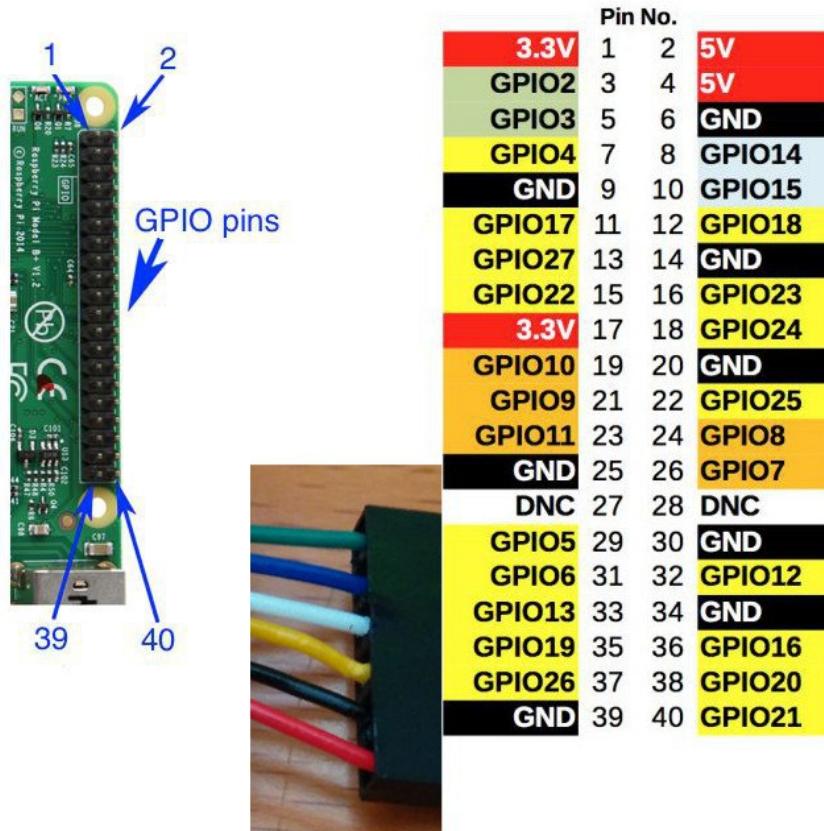


**NO** se ha de conectar la línea de 3V



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

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

4. 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:

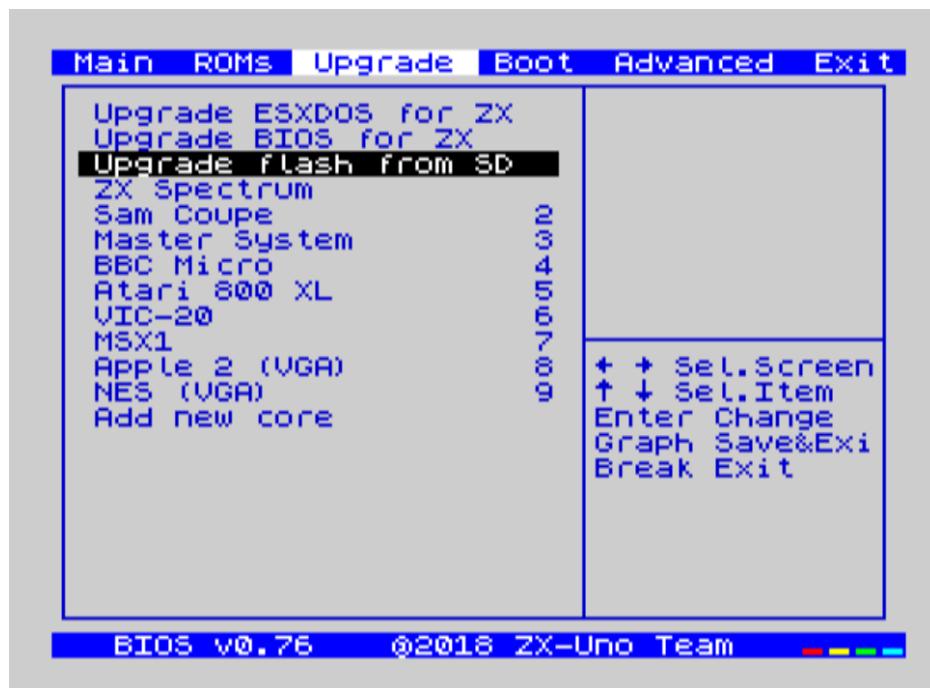
```
cd /home/pi/zxdosplus/unbrick  
sudo openocd -f /home/pi/zxdosplus/unbrick/raspberrypi2-native.cfg -f  
/usr/local/share/openocd/scripts/cpld/xilinx-xc6s.cfg -c "init; xc6s_program xc6s.tap;  
pld load 0 recovery.zxd.bit ; exit"
```

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"
```

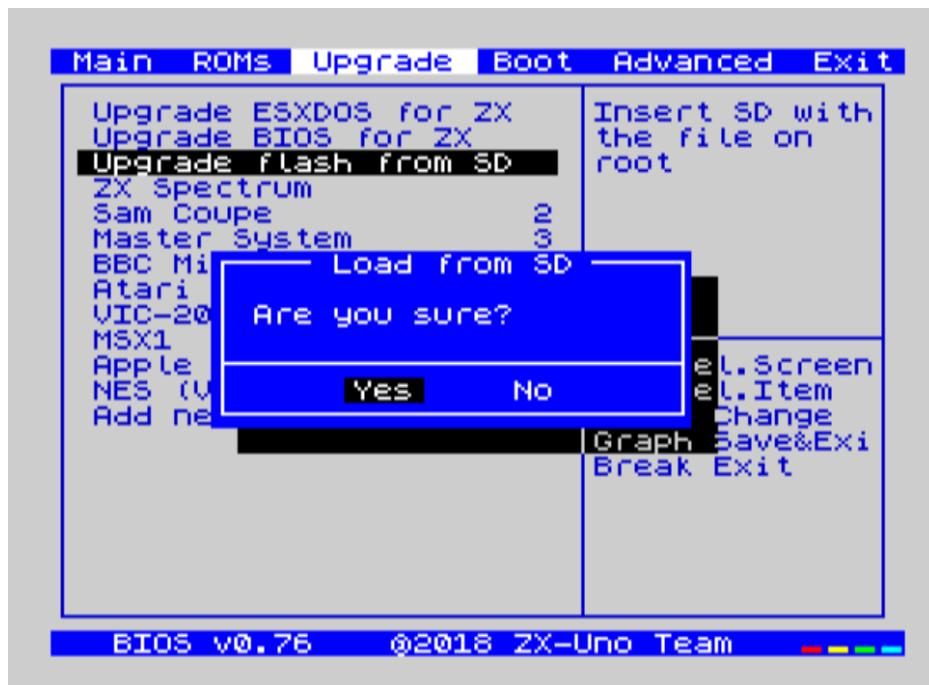
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** en gomaDOS+): para cambiar entre modo de video compuesto 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 por los que haya en la imagen, así como las ROMs de ZX Spectrum.



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 video compuesto 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 video compuesto 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 video compuesto (**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

[ZX-Uno](#)

[ZX-Uno FAQ](#)

[Guía rápida del ZX-Uno](#)

[Core ZX Spectrum](#)

[Layouts de teclado](#)

[Firmware de teclado para ZX Go+](#)

[zxunops2](#)

[Almost \(In-\) Complete List of esxDOS DOT-Commands](#)

[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](#)

[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](#)

[ColecóFPGA en GitHub](#)

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