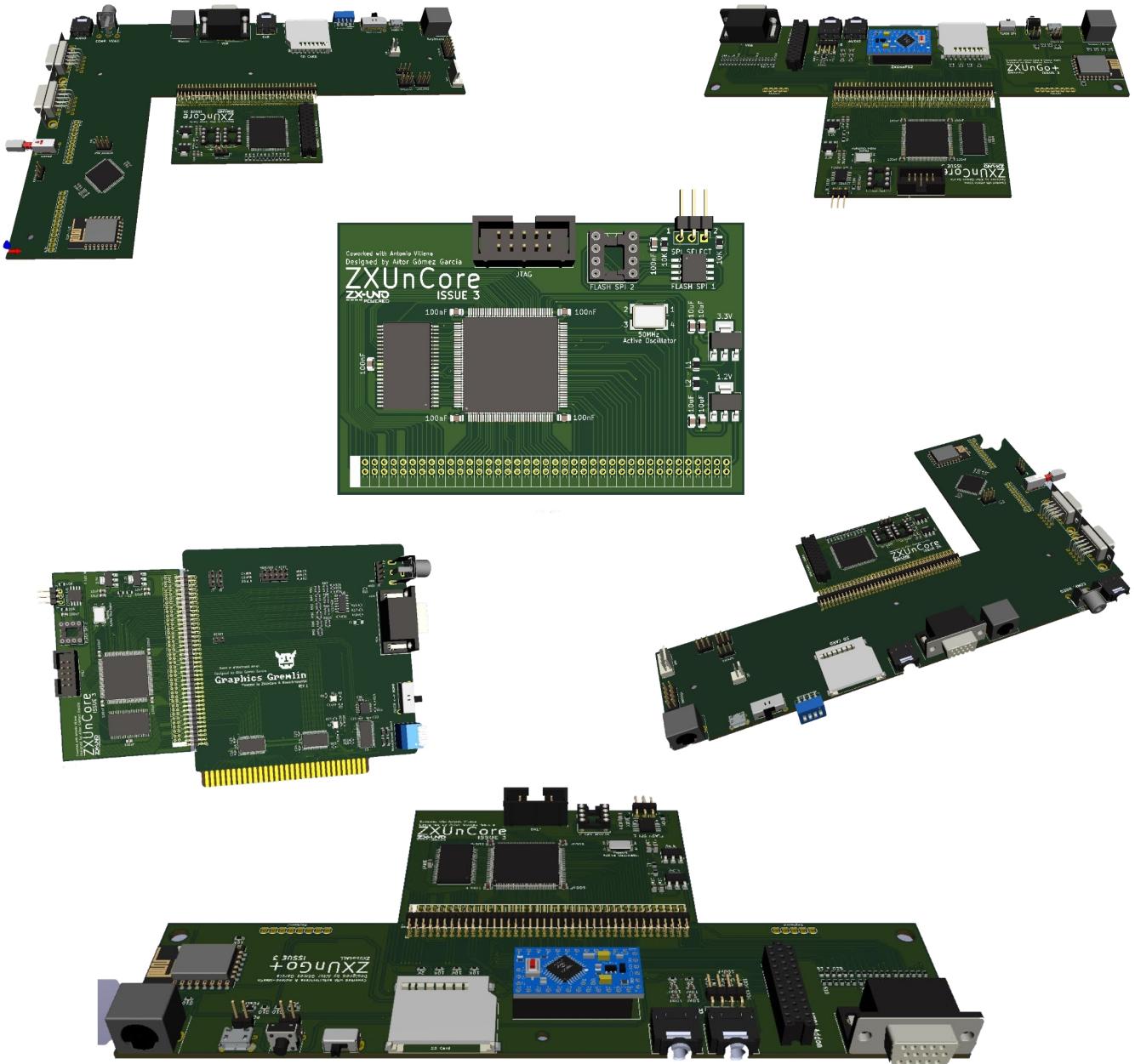


# ZXUno4ALL

## Manual



# Índice

Introducción .....	1
Agradecimientos .....	1
Módulos .....	2
ZXUnCore .....	2
ZXUnGo+ .....	3
Puertos y Conectores .....	4
Descripción .....	4
Configuración Inicial .....	5
Formato de la tarjeta SD .....	6
Windows .....	7
MacOS .....	8
Linux .....	10
Teclado .....	11
Teclado de membrana .....	11
Modos de teclado .....	12
Funciones .....	14
Actualización de firmware .....	15
Teclado PS/2 .....	18
Teclas especiales y botones .....	19
esxdos .....	21
BIOS .....	23
Main .....	24
ROMs .....	25
Upgrade .....	25
Boot .....	26
Advanced .....	27
Exit .....	28
ZX Spectrum .....	29
ROMs .....	31
DerbyPro .....	31
CargandoLeches .....	32
POKES .....	32
Preparación de cintas de carga ultrarrápida .....	33
SE Basic IV .....	34
Otras ROMs .....	35
Formato avanzado de la tarjeta SD (+3e) .....	36
Windows .....	36
MacOS .....	37

Linux .....	40
+3e .....	42
Comandos de esxdos .....	43
Guía básica .....	43
Comandos para ZX-Uno .....	45
Wi-Fi .....	46
Network tools for ZX-Uno pack .....	46
FTP-Uno .....	46
UART Terminal .....	47
Creación de vídeos RDM (RaDastan Movie) .....	48
CP/M .....	49
FUZIX .....	50
Cómo compilar .....	50
Cómo utilizarlo .....	51
Actualizaciones .....	55
BIOS .....	55
ROMs .....	55
Cores .....	56
esxdos .....	56
Memoria Flash .....	57
Otros cores .....	58
ZX Spectrum EXP27 sin PZX .....	58
ZX Spectrum 48K (Kyp) .....	58
Formato de Tarjeta SD .....	58
Teclado .....	58
Teclas especiales y botones .....	58
ZX Spectrum 128K (Kyp) .....	59
Formato de Tarjeta SD .....	59
Teclado .....	59
Teclas especiales y botones .....	59
ZX Spectrum (Kyp/azesmbog) .....	60
Formato de Tarjeta SD .....	60
Teclado .....	60
Teclas especiales y botones .....	60
ZX Spectrum TBBBlue .....	62
Formato de Tarjeta SD .....	63
Teclado .....	63
Teclas especiales y botones .....	63
Guía básica .....	64
Acorn Atom .....	65
Formato de Tarjeta SD .....	65

Teclado .....	66
Teclas especiales y botones.....	66
Guía básica .....	67
Acorn Electron .....	68
Formato de Tarjeta SD .....	68
Teclado .....	69
Teclas especiales y botones.....	69
Guía básica .....	69
Amstrad CPC 464 .....	71
Formato de Tarjeta SD .....	71
Teclado .....	71
Teclas especiales y botones.....	71
Guía básica .....	72
Amstrad CPC 6128 .....	73
Formato de Tarjeta SD .....	73
Teclado .....	73
Teclas especiales y botones.....	73
Guía básica .....	74
Apple I .....	75
Formato de Tarjeta SD .....	75
Teclado .....	75
Guía básica .....	76
Apple ][ .....	78
Formato de Tarjeta SD .....	78
Windows.....	78
MacOS y Linux.....	79
Teclado .....	79
Teclas especiales y botones.....	79
Arcades .....	80
Teclado .....	80
Teclas especiales y botones.....	80
Atari 800XL .....	81
Formato de Tarjeta SD .....	81
Teclado .....	81
Teclas especiales y botones.....	81
Guía básica .....	82
Atari 2600 .....	83
Formato de Tarjeta SD .....	83
Teclado .....	83
Teclas especiales y botones.....	83
Personalización de los controles .....	84

Guía básica .....	84
BBC Micro .....	85
Formato de Tarjeta SD .....	85
Teclado .....	86
Teclas especiales y botones .....	86
Guía básica .....	87
Computers Lynx .....	89
Formato de Tarjeta SD .....	89
Teclado .....	89
Teclas especiales y botones .....	89
Guía básica .....	90
CHIP-8 .....	91
Formato de Tarjeta SD .....	91
Teclado .....	91
Teclas especiales y botones .....	91
Guía Básica .....	92
Colecovision .....	93
Formato de Tarjeta SD .....	93
Teclado .....	93
Teclas especiales y botones .....	93
Guía básica .....	94
Colour Genie .....	95
Formato de Tarjeta SD .....	95
Teclado .....	95
Teclas especiales y botones .....	95
Guía básica .....	96
Commodore 16 .....	98
Formato de Tarjeta SD .....	99
Windows .....	99
MacOS y Linux .....	99
Teclado .....	100
Teclas especiales y botones .....	100
Guía básica .....	101
Commodore 64 .....	102
Formato de Tarjeta SD .....	103
Windows .....	103
MacOS y Linux .....	103
Teclado .....	104
Teclas especiales y botones .....	104
Guía básica .....	105
Commodore PET .....	106

Formato de Tarjeta SD .....	106
Teclado .....	106
Teclas especiales y botones .....	106
Guía básica .....	107
Commodore VIC-20 .....	108
Formato de Tarjeta SD .....	108
Teclado .....	108
Teclas especiales y botones .....	108
Guía básica .....	109
Galaksija .....	110
Formato de Tarjeta SD .....	110
Teclado .....	111
Teclas especiales y botones .....	111
Guía básica .....	112
HT-1080Z .....	113
Formato de Tarjeta SD .....	113
Teclado .....	114
Teclas especiales y botones .....	114
Guía básica .....	115
Jupiter ACE .....	117
Formato de Tarjeta SD .....	117
Teclado .....	117
Teclas especiales y botones .....	117
Guía básica .....	118
MSX .....	119
Formato de Tarjeta SD .....	119
Teclado .....	120
Teclas especiales y botones .....	120
Guía básica .....	121
MSXCTRL .....	122
Otros .....	122
NES .....	123
Formato de Tarjeta SD .....	123
Teclado .....	124
Teclas especiales y botones .....	124
Personalización de los controles .....	124
Guía básica .....	125
Oric Atmos .....	126
Formato de Tarjeta SD .....	126
Windows .....	126
MacOS y Linux .....	126

Teclado .....	127
Teclas especiales y botones .....	127
Guía básica .....	128
Oric Atmos (Kyp) .....	129
Formato de Tarjeta SD .....	129
Teclado .....	130
Teclas especiales y botones .....	130
Guía básica .....	131
Pong .....	133
Formato de Tarjeta SD .....	133
Teclado .....	134
Teclas especiales y botones .....	134
Guía básica .....	135
PC XT .....	136
Formato de Tarjeta SD .....	136
Windows .....	137
Linux .....	137
MacOS .....	138
Teclado .....	138
Teclas especiales y botones .....	138
Guía Básica .....	139
SAM Coupé .....	140
Formato de Tarjeta SD .....	140
Teclado .....	140
Teclas especiales y botones .....	140
Guía básica .....	141
Sega Master System .....	142
Formato de Tarjeta SD .....	142
Teclado .....	142
Teclas especiales y botones .....	142
Guía básica .....	143
SmartROM .....	144
Formato de Tarjeta SD .....	144
Vectrex .....	145
Formato de Tarjeta SD .....	145
Teclado .....	145
Teclas especiales y botones .....	145
Guía básica .....	146
Videopac .....	147
Formato de Tarjeta SD .....	147
Teclado .....	147

Teclas especiales y botones .....	147
Guía básica .....	148
Cambio del charset de la ROM de VDC .....	149
ZX81 .....	150
Formato de Tarjeta SD .....	150
Teclado .....	150
Guía básica .....	150
ZX81 y ZX80 .....	151
Formato de Tarjeta SD .....	151
Teclado .....	152
Teclas especiales y botones .....	153
Guía básica .....	153
Otro Hardware .....	155
Joy2PS2 .....	155
Actualización de firmware .....	156
Carga desde cinta .....	157
Reproductor de cassette .....	157
Ordenador .....	157
PlayTZX .....	157
Teléfono móvil, tableta, reproductor de sonido MP3, etc. ....	159
Conversión a fichero de audio .....	160
Solución de problemas .....	161
Gestión de imágenes de firmware .....	161
zx123_tool .....	161
Recuperación del firmware .....	165
Preparación del cableado .....	165
Recuperación usando una Raspberry Pi .....	166
Recuperación usando MacOS y Cable USB-Blaster .....	172
Recuperación usando Windows y Cable USB-Blaster .....	177
Referencias .....	181
Scan Codes .....	181
Enlaces .....	182

# Introducción

ZXUno4ALL es una variante, creada por spark2k06, 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 ZX-Uno ha ido creciendo, de forma que es posible instalar distintas configuraciones (cores) en la memoria flash de la FPGA, y que trabajan como otros sistemas distintos del ZX Spectrum, pudiendo elegir arrancar el ZX-Uno con la configuración que se desee de entre todas las instaladas.

La página oficial de ZX-Uno es <https://zxuno.speccy.org>.

La principal idea de ZXUno4ALL es facilitar y poner al alcance de cualquiera los desarrollos de cores y software creado para el ZX-Uno, mediante el uso de diferentes formatos de backplanes que hacen uso del modulo principal [ZXUnCore](#). Dadas las carecterísticas de éste, es posible crear proyectos derivados que no guarden relación con el ZX-Uno. Este manual está principalmente orientado al uso de [ZXUnGo+](#) o ZXUnSugarless, junto con [ZXUnCore](#).

En este documento, se hablará, en general, de ZXUnGo+, indicando las diferencias con otros donde sea necesario.

Los ficheros correspondientes a ZXUno4ALL están compartidos en el repositorio <https://github.com/spark2k06/ZXUno4ALL>.

También hay información sobre la evolución del proyecto en los foros Va de Retro:

<https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=9201>

<https://www.va-de-retro.com/foros/viewtopic.php?f=63&t=9248>

## Agradecimientos

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

- En el [foro de ZX-Uno](#)
- En [foroFPGA](#)
- En los varios FAQ existentes, principalmente la versión original [de @uto\\_dev](#), y la versión más moderna [de @desUBIKado](#)
- En el [canal oficial de Telegram de ZX-Uno](#)

Un agradecimiento especial a desUBIKado por la intensa y continua labor recopilando y recabando información sobre los distintos cores y funcionalidades.

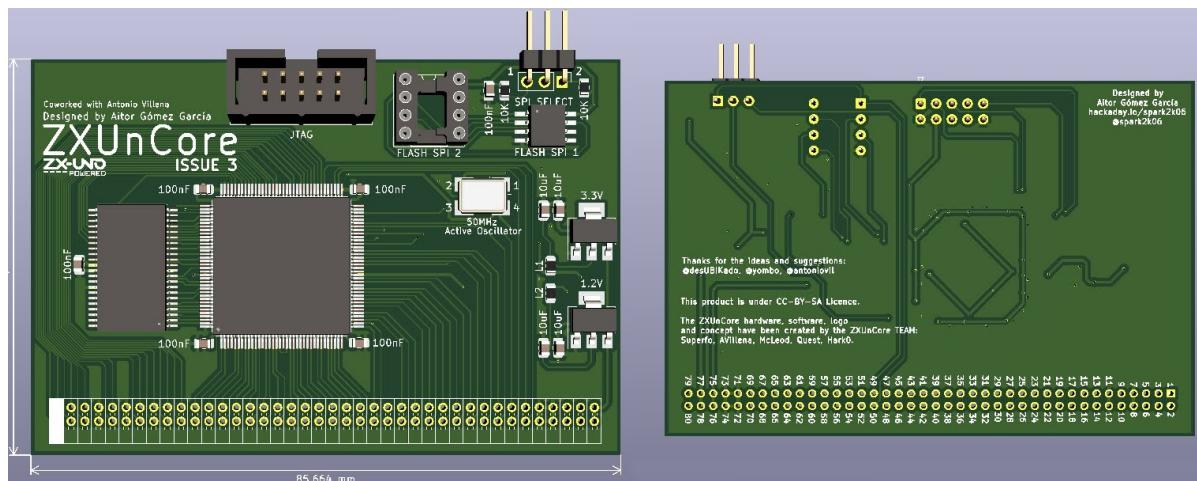
Sin el trabajo previo de todas estas personas (y más), ese manual no podría existir.

# Módulos

ZXUnCore

Módulo principal, y base principal para el resto. Se trata de un ZX-Uno reducido a la mínima expresión, con lo más básico y en formato módulo. Sus características principales son:

- Soporte para un DAC de Vídeo RGB666
  - Memoria SRAM integrada (512Kb o 2Mb).
  - Doble Flash SPI seleccionable en caliente, y una de ellas en formato Slot



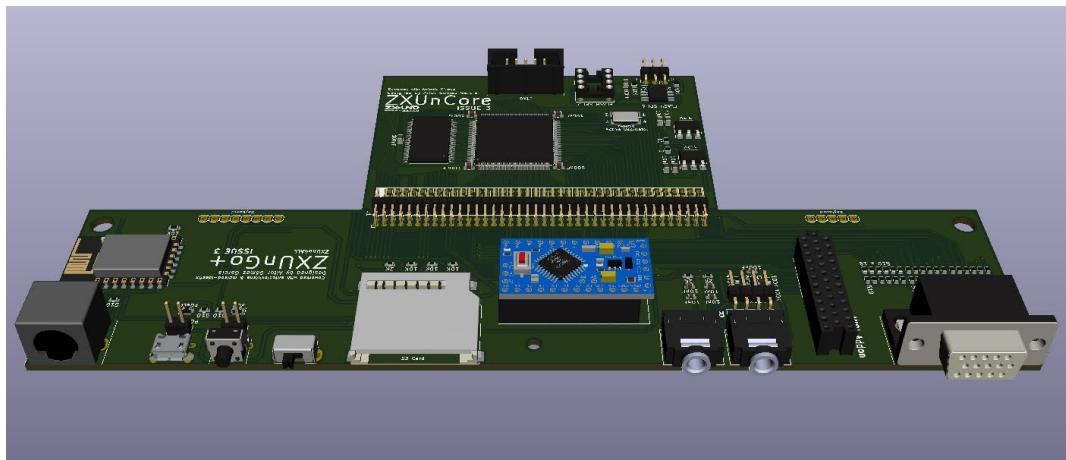
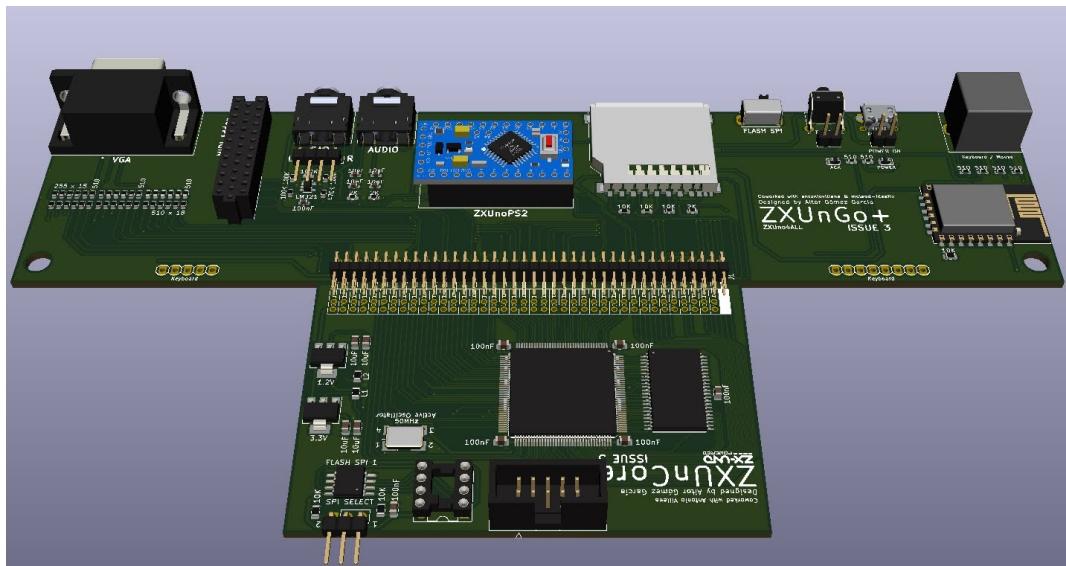
Este es el esquema de pines del módulo (ZXUNO RGB666 / GRAPHICS GREMLIN):

J1		J4	
	Conn_02x40_Counter_Clockwise		Conn_02x40_Counter_Clockwise
R4	1	2	R3
G3	3	4	R5
G5	5	6	G4
GND	7	8	GND
CSYNC	9	10	VSYNC
GND	11	12	GND
B4	13	14	B3
EAR	15	16	B5
P52_PIN2	17	18	P52_PIN6
P52_CLK	19	20	P52_DATA
JOY_UP	21	22	JOY_FIRE1
JOY_DOWN	23	24	JOY_LEFT
JOY_RIGHT	25	26	JOY_FIRE2
AUDIO_RIGHT	27	28	AUDIO_LEFT
LED	29	30	EXT35 WIFI TXD
EXT34	31	32	EXT33
WIFI_RXD	33	34	EXT31
WIFI_RST	35	36	EXT29 WIFI GPIO
R1	37	38	EXT27 R0
G0	39	40	EXT25 R2
G2	41	42	EXT23 G1
B1	43	44	EXT21 B0
EXT20	45	46	EXT19 B2
RST	47	48	FLASH_CS
FLASH_CS1	49	50	FLASH_CS2
JOY_FIRE3	51	52	EXT18
EXT17	53	54	EXT16 MIDI IN
EXT15	55	56	EXT14
EXT13	57	58	EXT12
EXT11	59	60	EXT10 MIDI CLE
EXT19	61	62	EXT8
MIDI_DABDO	63	64	EXT6
MIDI_WSBD	65	66	STD_N
SD_CS	67	68	SD_MOSI
STD_NB	69	70	SD_MISO
SD_SCK	71	72	GND
GND	73	74	GND
+1.2V	75	76	+1.2V
+3.3V	77	78	+3.3V
+5V	79	80	+5V
BUSRESET / AUDIO LEFT		AUX1	1
OSC10M		2	AUX2
ALE		3	1
GND		4	DSC14M31B
A0		5	AEN
GND		6	GND
A2		7	8
A4		9	OWS
GND		10	GND
A6		11	12
A8		13	14
A10		15	16
A12		17	18
A14		19	20
A16		21	22
A18		23	24
DATA_DIR		25	A11
IOR		26	A13
SMEMR		27	A15
D0		29	A17
D2		31	A19
D4		33	10W
D6		35	SMEMW
VGA_VSYNC		36	36
RDY		37	38
B1		39	D1
B3		40	D3
B5		41	D5
G1		43	D7
G3		45	VGA_HSYNC
G5		47	48
B0		49	50
B2		51	52
B4		53	54
B6		55	56
G0		57	58
G2		59	60
G4		61	62
G6		63	64
R1		65	66
R3		67	68
R5		69	70
GND		71	72
GND		73	74
+1V2		75	76
+3V3		77	78
+5V		79	80
+5V			+5V

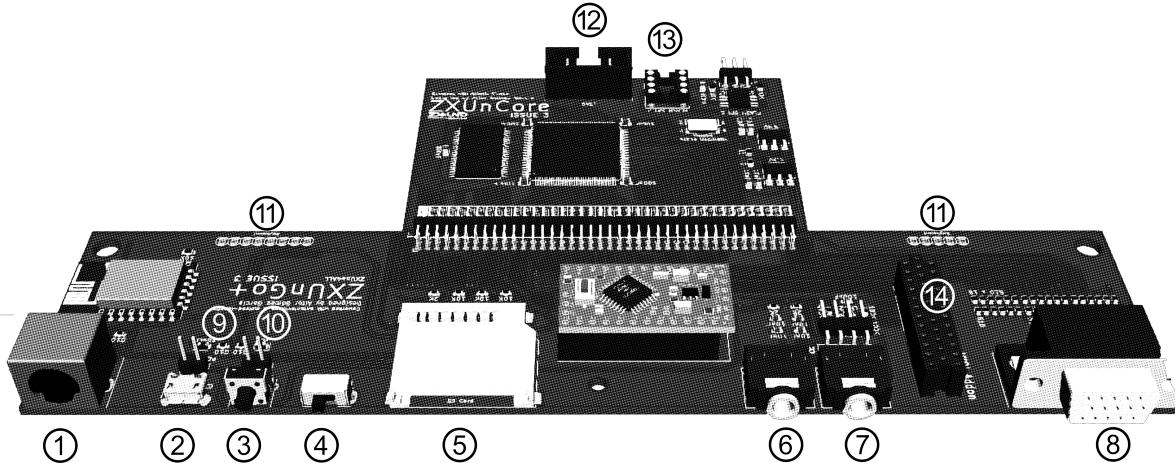
## ZXUnGo+

Placa en formato ZX-Spectrum 16K/48K (ZX Spectrum original y ZX Spectrum+). Hay que llevar a cabo una única modificación en la carcasa para facilitar el acceso al conector VGA, simplemente haciendo más grande salida original de TV (eliminando los tornillos y la chapa del conector VGA, esta ampliación es mínima). Sus características principales son:

- Salida de vídeo VGA
- Interruptor selector de SPI Flash
- Botón de reset
- Nuevo circuito mejorado de EAR diseñado por mcleod-ideafix, que es más sensible a la carga y funciona bien incluso cargando desde un móvil
- Soporte para conectar teclado de membrana de ZX Spectrum, usando ZXUnoPS2 y un Arduino Mini Pro basado en Atmega 328P en la versión de 5V
- Soporte para módulo Wi-Fi
- Conector para addon MIDI (compatible con el addon desarrollado por McLeod)



## Puertos y Conectores



## Descripción

1	Puerto de Teclado y ratón (PS/2)
2	Alimentación (USB Mini-A)
3	Botón de Reset
4	Selector de SPI Flash
5	Ranura SD
6	Salida de Sonido
7	Entrada de Sonido
8	Salida de vídeo VGA
9	Puente para añadir interruptor de encendido
10	Puente para añadir botón de reset
11	Conectores para membrana de teclado de ZX Spectrum
12	Conector JTAG
13	Conector para SPI Flash externa
14	Conector para módulo MIDI

# Configuración Inicial

Para poder poner en marcha un ZXUnGo+ hace falta, al menos, lo siguiente:

- Un cargador USB, una TV u otro dispositivo que ofrezca alimentación USB. Normalmente con 500 mA es suficiente.
- Un cable y un monitor o TV con entrada VGA
- Un teclado de membrana de ZX Spectrum (o ZX Spectrum+), o bien un teclado PS/2

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

- Una tarjeta SD, 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
- Un ratón PS/2 (se necesita adaptador PS/2 en "Y")
- 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, 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).

## Formato de la tarjeta SD

Para poder utilizar una tarjeta SD con el core principal de Spectrum, esta debe tener, al menos, una partición (la primera en el caso de haber varias) en formato FAT16, FAT32 u otros (según el caso, se puede necesitar, un formato específico para compatibilidad con distintos cores de terceros). 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.



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 (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.

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=ZXUNCORE
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=ZXUNCORE 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=ZXUNCORE unit=4k quick
create part primary
format fs=FAT32 label=EXTRA quick
exit
```

## 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" ZXUNCORE 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" ZXUNCORE 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% ZXUNCORE 4G %DOS_FAT_16% EXTRA 4G
"MS-DOS FAT32" DATA R
sudo newfs_msdos -F 16 -v ZXUNCORE -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" ZXUNCORE 4G "MS-DOS FAT32" EXTRA
R
```

Para el core de Amstrad CPC 6128, 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 ZXUNCORE -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 SD se llama **ZXUNCORE**):



```
mdutil -i off /Volumes/ZXUNCORE
cd /Volumes/ZXUNCORE
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 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.

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)

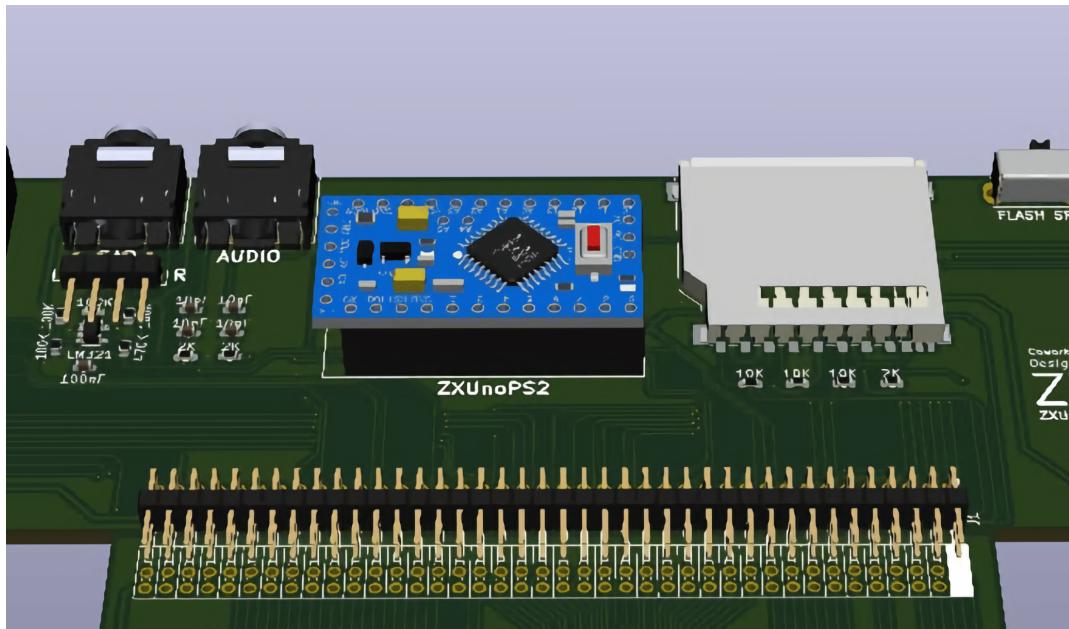
```
sudo mkfs.fat -F 16 -n ZXUNCORE -s 128 /dev/sdc1
```

# Teclado

## Teclado de membrana

Con el módulo ZXUnGo+, se puede conectar un teclado de membrana de ZX Spectrum original, utilizando una placa Arduino, que se encarga de transformar las pulsaciones al protocolo PS/2 de teclado, usando [el firmware zxunops2](#).

Esta placa está programada de manera que pueda comportarse de distintas maneras (modos de teclado) según sea necesario.



## Modos de teclado

Por defecto, está configurado en modo ZX Spectrum. Para cambiar a otro modo, se debe pulsar **Caps Shift+Symbol Shift+F**, luego **U**, y finalmente la tecla correspondiente. Al hacerlo, se tecleará automáticamente un texto indicando el modo seleccionado (por ejemplo: **.zx** si se pulsa **Caps Shift+Symbol Shift+F, U**, y luego **1**).

Modo	Tecla
ZX Spectrum	<b>1</b>
Amstrad CPC	<b>2</b>
MSX	<b>3</b>
Commodore 64	<b>4</b>
Atari 800XL	<b>5</b>
BBC Micro	<b>6</b>
Acorn Electron	<b>7</b>
Apple (I y II)	<b>8</b>
Commodore VIC 20	<b>9</b>
PC XT	<b>0</b>
Oric Atmos	<b>A</b>
SAM Coupé	<b>B</b>
Jupiter ACE	<b>C</b>
ZX81	<b>D</b>
Commodore 16 / Plus 4	<b>E</b>
HT-1080Z / Video Genie / TRS-80 Model I G – Galaksija	<b>F</b>
Galaksija	<b>G</b>
Multicomp	<b>H</b>
ZX80	<b>I</b>
Atom	<b>J</b>
Commodore PET	<b>K</b>
Colour Genie	<b>L</b>
Lynx	<b>M</b>
Enterprise	<b>N</b>

El modo de teclado ZX puede funcionar de dos formas distintas, entre las que se puede alternar usando **Caps Shift+Symbol Shift+F** y luego **D**:

- En modo convencional, con acceso a **Caps Shift** y **Symbol Shift** originales del teclado pero sin acceso a las teclas especiales ni a las teclas modificadoras.
- En modo completo, sin acceso a **Caps Shift** y **Symbol Shift** originales del teclado pero con acceso a las teclas especiales y a las teclas modificadoras, como el resto de modos de teclado.

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

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>0</b>
<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>	<b>F5</b>	<b>F6</b>	<b>F7</b>	<b>F8</b>	<b>F9</b>	<b>F10</b>
<b>Q</b>	<b>W</b>	<b>E</b>	<b>R</b>	<b>T</b>	<b>Y</b>	<b>U</b>	<b>I</b>	<b>O</b>	<b>P</b>
<b>F11</b>	<b>F12</b>	*	-	Impr. Pant	Bloq. Desp	Pausa	Bloq. Mayús.	([ y ^) ])	+ (* y ])
<b>A</b>	<b>S</b>	<b>D</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>Enter</b>
Mayús.	< y >	+		Ins.	Inicio	Re Pág	Ñ	' ({ y "} )	
<b>CShift</b>	<b>Z</b>	<b>X</b>	<b>C</b>	<b>V</b>	<b>B</b>	<b>N</b>	<b>M</b>	<b>SShift</b>	<b>Space</b>
	<b>Ctrl</b>	<b>Win</b>	<b>Alt</b>	<b>Supr</b>	<b>Fin</b>	<b>Av Pág</b>	<b>Alt Gr</b>		



Al pulsar una combinación de tecla modificadora con otra (por ej. **Mayús.+F10**), no es necesario soltar **Caps Shift+Symbol Shift** entre pulsaciones (en el ejemplo anterior, **Caps Shift+Symbol Shift+A**, y **Caps Shift+Symbol Shift+0**).



Pulsar **Caps Shift+Symbol Shift+F**, luego **U**, y una tecla no válida, teclea automáticamente el texto indicando el modo actual.



Se puede obtener el código fuente, binarios y más instrucciones sobre el firmware de teclado en [el repositorio oficial](#).



La combinación **Caps Shift+Symbol Shift+I** se comporta de forma diferente en los siguientes modos:

- Commodore 16 y VIC-20: **Bloq. Num**
- Commodore Pet: Tecla **Menu** (derecha de **Alt Gr**), que se utiliza como **Bloq Mayús**



No se recomienda usar el modo de teclado **PC XT**, a la vez que un teclado de PS/2

## Funciones

Pulsar **Caps Shift+Symbol Shift+F** y luego una tecla asociada, permite activar una serie de funciones especiales.

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

Tecla	Acción	Teclados compatibles
V	Imprime versión	Todos
X	Guarda configuración en EEPROM	Todos
B	<b>Ctrl+Alt+Bcksp</b> (Hard Reset)	Todos
N	<b>Ctrl+Alt+Supr</b> (Soft Reset)	Todos
Y	<b>Ctrl+Alt+F5</b> (NMI)	ZX, CPC, Jupiter, SAM, ZX80, ZX81
O	<b>F12</b> seguido de <b>Mayús</b> con pausa de 1 seg.	BBC
H	<b>F8+F10</b>	Atari
U	Cambio de modo de teclado	Todos
C	Mapeo de las teclas <b>0, P, Q, A</b> a bloque numérico (Atari) o cursores	PC, MSX, SMS y Atari
D	Activa y desactiva el modo completo en teclado ZX	ZX
G	Cambiar el modo de vídeo	ZX, Atari, BBC, VIC, ORIC y HT-1080Z

La función **C** activa o desactiva el mapeo de los cursores a **0, P, Q, A** y la tecla **M** se intercambia con la **Espacio**. Funciona en todos los tipos de teclado, excepto:

- Teclados C64 y Atari 800XL: en lugar de los cursores, se mapean a las teclas **8, 4, 6** y **2** del teclado numérico, y la **M** a la **0** (disparo)
- Teclado Jupiter ACE: en lugar de los cursores se mapea a las teclas **8, 4, 6** y **2** del teclado numérico, y la **M** y 'Espacio' se intercambian.

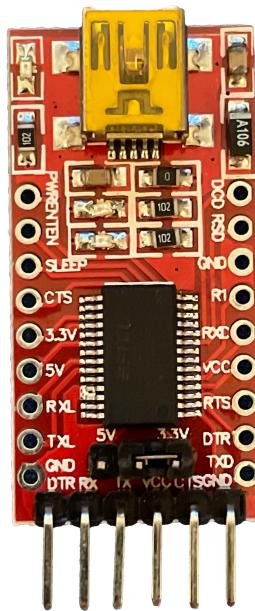
En todos los modos de teclado se ha intentado que coincidan la mayoría de los símbolos con los que hay impresos en el teclado de Spectrum. Así **Symbol Shift+M** obtiene “.” o **Symbol Shift+K** obtiene +. Incluso en algunos se han sustituido algunos símbolos cuando eran más prácticos. Por ejemplo en el teclado MSX Espanol versión 1.3 la combinación **Symbol Shift+E** obtiene “í” y **Symbol Shift+I** obtiene “¿”.

## Actualización de firmware

Para poder instalar el firmware, se necesita un ordenador (Windows, Mac, Linux).

### Material necesario:

- Ordenador (Windows, Mac, Linux)
- 6 [cables puente para prototipos](#) (idealmente, hembra en los dos extremos)
- Adaptador USB Serie (FTDI)

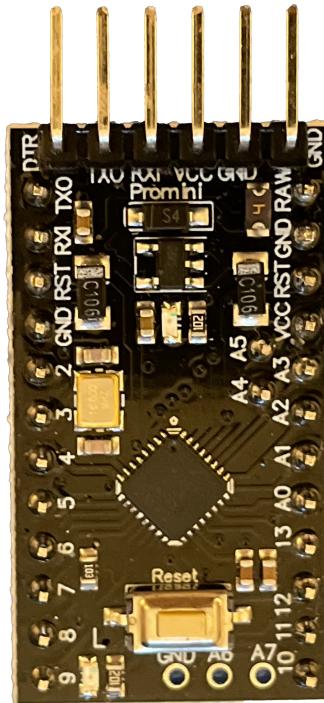


### Software necesario:

- [XLoader](#) (sólo para Windows) o [AVRDUDE](#) (es posible utilizar la versión incluida con el entorno [Arduino IDE](#)).
- Fichero **.hex** binario con la versión a instalar, que se puede obtener en [el repositorio oficial](#) (normalmente, el fichero deseado es uno de los terminados en **-328.hex**).

## Preparación

Desconectar la placa Arduino (si fuera necesario), e identificar sus conexiones.



Conectar el adaptador serie con la placa Arduino, de la siguiente forma:

USB FTDI	Arduino Mini
DTR	DTR
RX	TXO
TX	RXI
VCC (3.3V)	VCC
CTS	GND
GND	GND



Si la placa arduino utiliza 5V, se puede realizar la conexión con ese voltaje en vez de 3,3V

Conectar el adaptador USB al ordenador, e identificar el puerto donde se ha conectado (**COM**, **/dev/usb...**, etc.).

**xLoader**

Seleccionar los siguientes parámetros:

- Device: Duemilanove/Nano (ATMega328)
- Baud rate: 57600
- Port: Puerto COM correspondiente al adaptador

Pulsar "Upload" y esperar a que se muestre el mensaje "xx bytes uploaded".

**AVRDUEDE**

Usar un comando de la forma:

```
avrdude -U flash:w:<fichero.hex>:i -e -p atmega328p -b 57600 -c arduino -P <puerto USB>
```

En el caso de utilizar la versión de **avrdude** incluida con Arduino IDE, se debe indicar también el fichero **.conf** incluido. Por ejemplo, en MacOS:



```
/Applications/Arduino.app/Contents/Java/hardware/tools/avr/bin/avrdude -U  
flash:w:zxunops2-25092021-328.hex:i -e -p atmega328p -b 57600 -c arduino -P  
/dev/cu.usbserial-A50285BI -C  
/Applications/Arduino.app/Contents/Java/hardware/tools/avr/etc/avrdude.conf
```

## Teclado PS/2

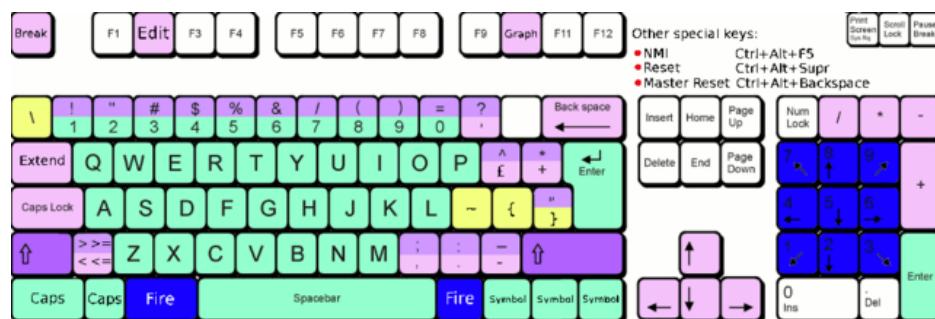
El mapa de un teclado conectado al puerto PS/2 (asignación de las teclas físicas del teclado con las pulsaciones que se presentan a los distintos cores) se cambia desde el menú **Advanced** de la BIOS. Existen tres mapas distintos a elegir: Español (por defecto), inglés, y Spectrum (avanzado).

También se puede cambiar con la utilidad **keymap**. Dentro de **/sys** 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.

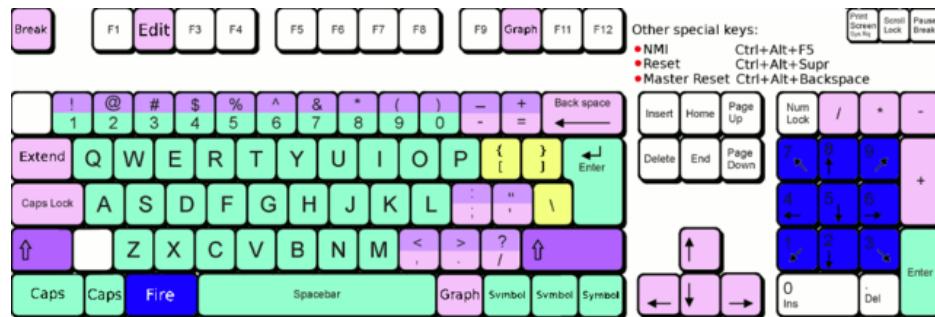
Para que el mapa se conserve después de un master reset, hay que tener seleccionado **Default** en la configuración de BIOS.

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

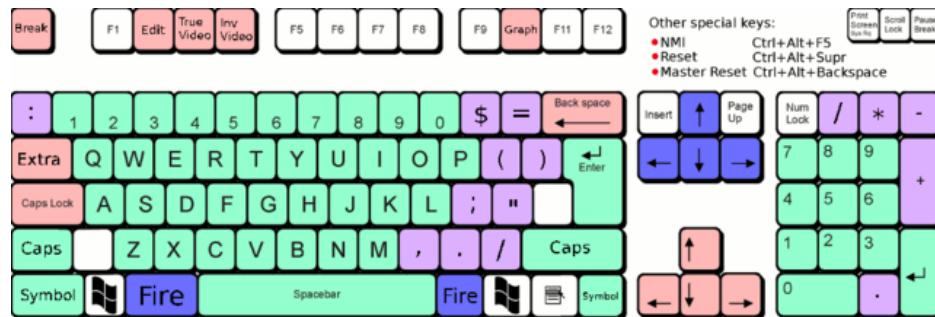
### Español



### Inglés



### Spectrum



## Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **ZX**. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Teclas especiales durante el arranque:

- **F2** (**Caps Shift+1** en ZXUnGo+) Entrar en la BIOS
- **Bloq. Mayús o Cursor abajo** (**Caps Shift+2** en ZXUnGo+) o **Caps Shift+6**. o, si hay un joystick conectado, pulsar la dirección hacia abajo: Menú de selección de cores
- **Esc** (**Caps Shift+Espacio** en ZXUnGo+) 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 (**Symbol Shift+V** en ZXUnGo+): 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

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

- **Esc** (**Caps Shift+Espacio** en ZXUnGo+): BREAK
- **F2** (**Caps Shift+1** en ZXUnGo+): Edit
- **F3** (**Caps Shift+3** en ZXUnGo+): True Video
- **F4** (**Caps Shift+4** en ZXUnGo+): Inverse Video
- **F5** (**Caps Shift+Symbol Shift+5** o **Caps Shift+Symbol Shift+Y** en ZXUnGo+): NMI
- **F6** (**Caps Shift+Symbol Shift+6** en ZXUnGo+): Pone una marca personalizada en la posición actual de un PZX que se haya cargado (sólo en la versión con soporte para PZX)
- **F7** (**Caps Shift+Symbol Shift+7** en ZXUnGo+): Reproducir o Pausa en la reproducción de archivos .PZX
- **F8** (**Caps Shift+Symbol Shift+8** en ZXUnGo+): Mueve la posición de PZX a la marca personalizada
- **CTRL+ F8** (**Caps Shift+Symbol Shift+Z** y, a continuación, **Caps Shift+Symbol Shift+8** en ZXUnGo+): Rebobina el PZX hasta una marca del tipo "tag BRWS". Estas marcas sirven para señalar, por ejemplo, el comienzo de una multicarga (sólo en la versión con soporte para PZX)
- **F9** (**Caps Shift+Symbol Shift+9** en ZXUnGo+): Para la reproducción del PZX y rebobina hasta el principio
- **F10** (**Caps Shift+9** en ZXUnGo+): Graph
- **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+): Turbo Boost. Alterna la CPU entre 28MHz y 3,5MHz
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**.
- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+F** y luego **N** en ZXUnGo+): Soft reset.
- **Fin** (**Caps Shift+Symbol Shift+B** en ZXUnGo+): Alterna la visualización entre color, monocromo verde, monocromo ámbar o blanco y negro

## esxdos

**esxdos** es un firmware para la interfaz the DivIDE/DivMMC, que el ZX-Uno implementa, y que permite el acceso a dispositivos de almacenamiento como la tarjeta SD. 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 SD.

En el momento de escribir este documento, última versión es la 0.8.9, 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-Uno se verá cómo esxdos detecta la tarjeta y carga los componentes necesarios para funcionar.



```

esxdos
v0.8.6-DivMMC
© 2005-2016
Papaya Dezign

Detecting Devices...
sda: card

Mounting drives...
hd0: ZX , FAT16, 128M

Loading ESXDOS.SYS... [OK]
Loading RTC.SYS... [OK]
Loading NMI.SYS... [OK]
Loading BETADISK:SYS... [OK]

```

Es recomendable, además, añadir los comandos específicos para ZX-Uno. Estos se pueden obtener en la página con el código fuente del proyecto ([aquí](#), [aquí](#) y [aquí](#) - Usuario **guest**, contraseña **zxuno**), y son los siguientes:

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

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

# BIOS



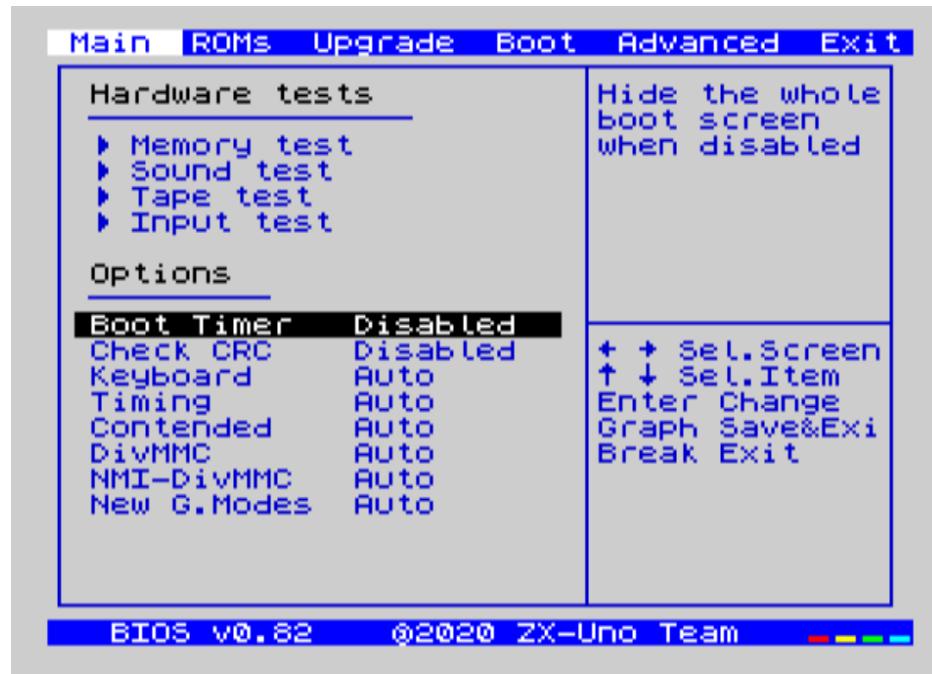
Si se pulsa la tecla **F2** (**Caps Shift+1** en ZXUnGo+) durante el arranque, se tendrá acceso a la configuración de BIOS. El firmware de BIOS es el primer programa que se ejecuta cuando se enciende el ZX-Uno. El propósito fundamental del software de BIOS es iniciar y probar el hardware y cargar uno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

Usando las teclas de cursor izquierda y derecha (**Caps Shift+5** y **Caps Shift+8** en ZXUnGo+), 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 ZXUnGo+) 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 ZXUnGo+) 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 Shif+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

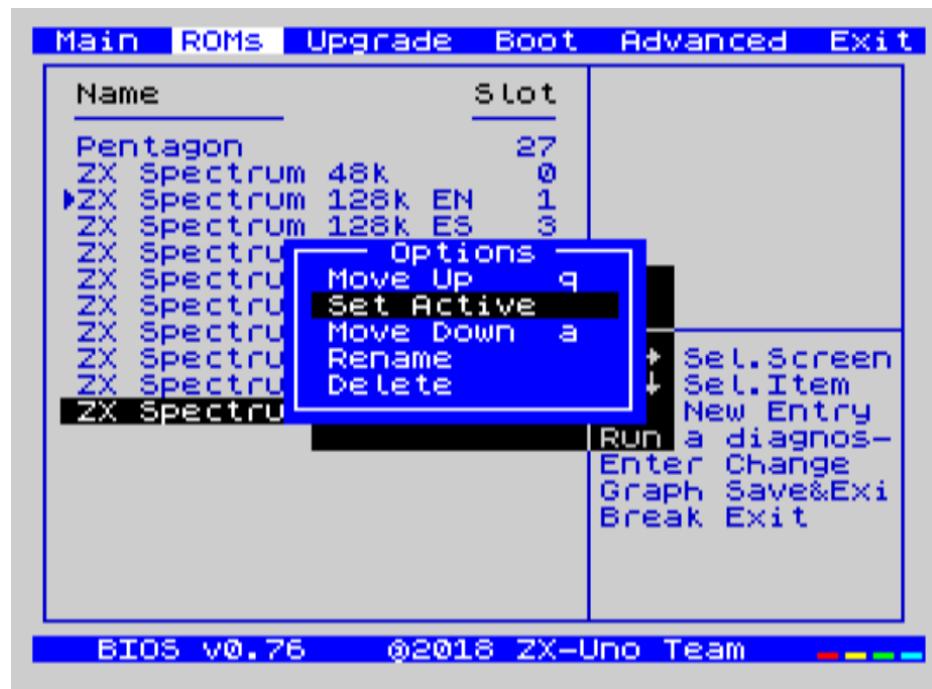


En la primera pantalla de configuración, además de poder ejecutar distintas pruebas, se puede definir el comportamiento por defecto para lo siguiente:

- 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)

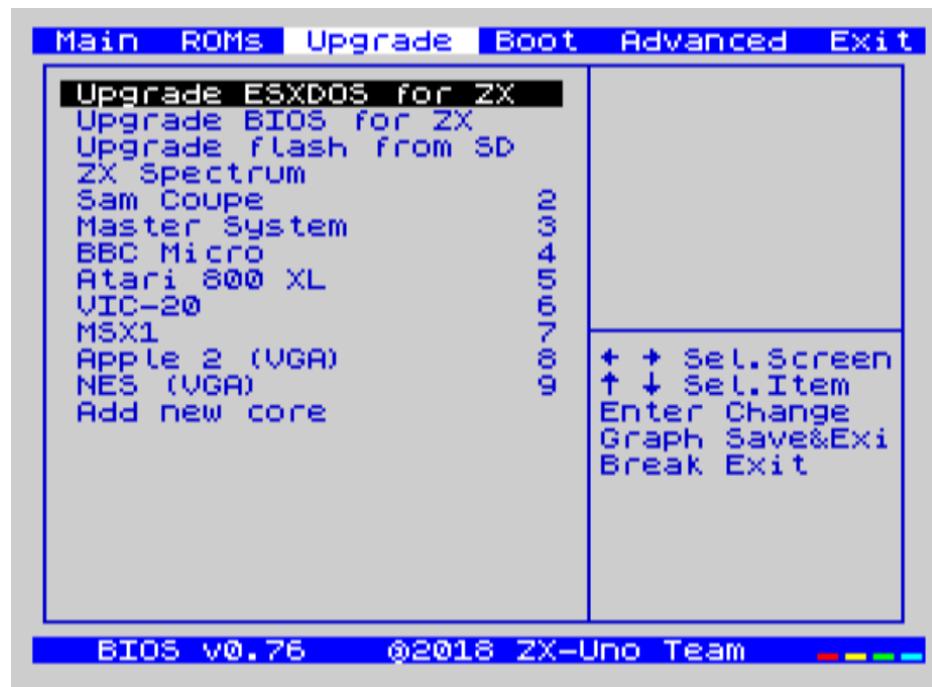
Se puede consultar información más técnica en [la Wiki de ZX-Uno](#).

ROMs



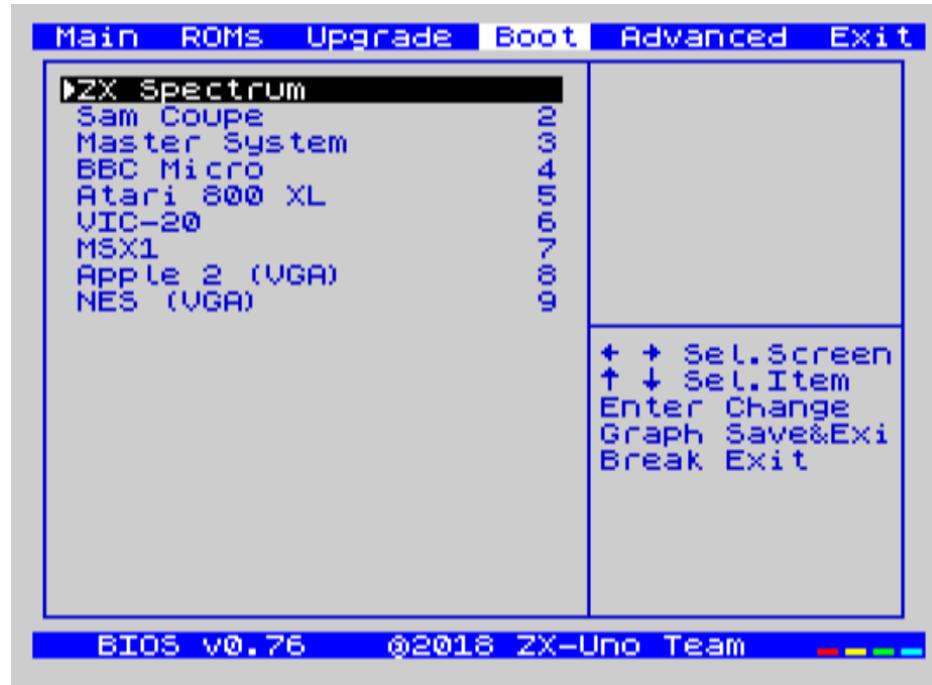
La segunda pantalla muestra las ROMs de ZX Spectrum instaladas y permite reordenar (Move Up, Move Down), renombrar (Rename) o borrar (Delete) cada una de ellas, así como elegir la que se cargará por defecto en el arranque (Set Active).

## 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

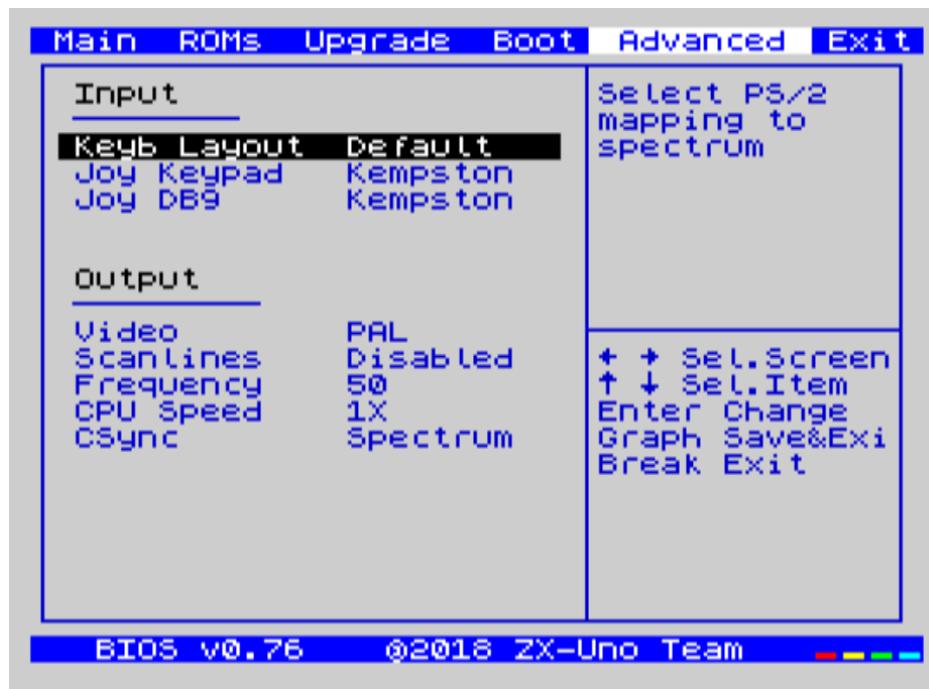


En la pantalla *Boot* se puede elegir qué core de los instalados se desea que cargue por defecto en el arranque.



Nunca se ha de elegir como arranque un core de tipo Spectrum distinto del principal (Primer core de la flash). Esto provoca un bucle de reinicios constantes del mismo.

## 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 de un teclado PS/2 (Joy Keypad): Kempston, Sinclair Joystick 1, Sinclair Joystick 2, Protek o Fuller
- Comportamiento de un joystick conectado a una expansión de ZX-Uno (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

La siguiente tabla resume las distintas asignaciones de botones de joystick:

Botón	Kempston	Sinclair 1	Sinclair 2	Protek	Fuller
Izquierda	Bit 0	1	6	5	Bit 2
Derecha	Bit 1	2	7	8	Bit 3
Abajo	Bit 2	3	8	6	Bit 1
Arriba	Bit 3	4	9	7	Bit 0
Disparo 1	Bit 4	5	0	0	Bit 4
Disparo 2	Bit 5	X	Z	9	Bit 5

## 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-Uno lo utiliza para su funcionamiento. La versión instalada por defecto es una variación del core EXP27-160820 de mcleod\_ideafix.

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
- Soporte Covox
- 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 QAOOPSpCm.
- 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 (PAL y NTSC), 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.
- Soporte Wi-Fi (UART) vía addon
- Soporte MIDI (General MIDI) vía addon
- Carga de ficheros de ficheros **PZX** desde la SD

La versión incluida con ZXUnCore tiene las siguientes mejoras:

- Soporte para el DAC de Vídeo RGB666
- Visualización de la imagen en Color, monocromo verde, monocromo ámbar o en blanco y negro pulsando la tecla **Fin** (**Caps Shift+Symbol Shift+B** en ZXUnGo+)
- Modo Turbo Boost: Se ha modificado el funcionamiento de la tecla **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+) para alternar la velocidad de la CPU entre 28Mhz (4x) y 3,5Mhz (1x).
- Se ha mejorado la funcionalidad de las cargas de ficheros .PZX, recuperando la posibilidad de poner una marca personalizada

Tal y como se podía hacer en el primer core T24 no oficial sintetizado por Neuro con la carga de ficheros .PZX se pueden usar las siguientes teclas:

- **F6** (**Caps Shift+Symbol Shift+6** en ZXUnGo+): Pone una marca personalizada en la posición actual
- **F7** ('**Caps Shift+Symbol Shift+7** en ZXUnGo+') : Play/Pausa
- **F8** ('**Caps Shift+Symbol Shift+8** en ZXUnGo+') : Poner la cinta en la marca personalizada
- **Ctrl+F8** (**Caps Shift+Symbol Shift+Z** y luego **Caps Shift+Symbol Shift+8** en ZXUnGo+): Rebobinar hasta una marca del tipo **tag BRWS**. Estas marcas sirven para señalar, por ejemplo, el comienzo de una multicarga.
- **F9** (**Caps Shift+Symbol Shift+9** en ZXUnGo+): Detiene la cinta y la rebobina hasta el principio.

Cuando termina la reproducción completa del fichero PZX, el reproductor se para automáticamente. También se parará si se encuentra en el fichero una marca STOP, o si se encuentra una marca STOP IF IN 48K y el ZX-Uno está en modo de 48K (sin paginación de 128K).



Es posible añadir tags BRWS o STOP a un fichero PZX usando el programa [ZX-Blockeditor](#), perteneciente al conjunto de utilidades [ZX-Modules](#).

# 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-Uno, 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.

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

Esta ROM de 64K tiene soporte para nuevo hardware, incluyendo el uso de comandos de esxdos desde 128 BASIC, así que se pueden usar estas opciones al [añadirla a la SPI flash](#):

Ajuste	Significado
<b>d</b>	Habilitar DivMMC
<b>n</b>	Habilitar NMI DivMMC (menú de esxdos)
<b>t</b>	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 ultarrápida 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).

## SE Basic IV

[SE Basic IV](#) es un intérprete de Microsoft BASIC, gratuito y de código abierto. SE Basic IV se ha diseñado para funcionar en un [Chloe 280SE](#) pero también lo hace en ZX-Uno y similares.

SE Basic se pensó en sus orígenes como un nuevo firmware para el [ZX Spectrum SE](#). Las primeras versiones eran modificaciones aplicadas sobre la ROM original del ZX Spectrum, pero posteriormente se ha reescrito basándose en la [ROM mejorada y de código abierto TS1000 / ZX81](#).

La versión 3, también conocida como [OpenSE BASIC](#), todavía se mantiene como un firmware de código abierto para el Spectrum. Se incluye, por ejemplo, en el [repositorio principal de Debian](#) para su uso en emuladores.

La versión IV es una nueva rama de la versión anterior, creada principalmente porque no quedaba espacio para añadir nuevas características a la ROM de 16K. La primera versión (4.0 Anya) añadió una segunda ROM de 16K con soporte para el modo hi-res de Timex. La sintaxis era aún bastante compatible con Sinclair BASIC. La versión 4.2 fue reconstruida específicamente para el Chloe 280SE, eliminó el soporte para dispositivos antiguos como las cintas, añadió soporte integrado y compatibilidad total con el kernel de esxdos, y migró a la sintaxis de Microsoft BASIC.

Aunque guarda una base común de código con muchas versiones de Sinclair BASIC (la ROM TS1000), hay varias diferencias significativas:

- Soporte para páginas de códigos(8-bit ASCII).
- Traducción de los mensajes de error.
- 38 tokens nuevos.
- Presentación en una terminal estándar (80 x 24).
- Introducción de caracteres de terminal (CTRL, META).
- Buffer de teclado.
- Soporte para teclados completos.
- Soporte completo de las características adicionales del core de Spectrum de ZX-Uno

Las principales diferencias con Microsoft BASIC son:

- Abreviación de tokens.
- Comprobación de sintaxis en cada entrada.
- Normalmente los paréntesis son opcionales.
- Introducción de número con el estilo Motorola: % – binario @ – octal \$ – hexadecimal
- Evaluación de expresiones siempre activa.
- Operadores lógicos y bit a bit separados.
- Tipado automático de datos.

Se puede encontrar mucha más información, incluyendo el manual de usuario, etc. en [la wiki oficial](#).

## 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 SD (+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 ZXUNCORE -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 ZXUNCORE 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 (1 primary, 0 extended, 3 free)
e extended (container for logical partitions)
Select (default p): p
Partition number (2-4, 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 2 of type 'Linux'

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

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

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

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1		128	2017152	2017025	984,9M	7f	unknown
/dev/sda2 *		2017280	7626751	7812504	2,7G	b	FAT16
/dev/sda3		9829784	7626751	21286504	21G	b	W95 FAT32

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

```
sudo mkfs.fat -F 16 -n ZXUNCORE -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 IDEDOS según se deseé.

El primer paso consiste en determinar la geometría de la SD. Con la tarjeta insertada en el ZX-Uno, 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 necesitarán son  $32768/16=2048$ . Por tanto, podemos formatear la partición IDEDOS 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 IDEDOS 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 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 o BAS.
- **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 SD 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.



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-Uno

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

- **backup**
- **back16m**: Copia a un fichero **FLASH.ZX1** en el directorio raíz de la tarjeta SD el contenido de la 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 SD. 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 RomPack, llamado **ROMS.ZX1**, en el directorio raíz de la tarjeta SD 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 RomPack, llamado **ROMS.ZX1**, en el directorio raíz de la tarjeta SD con todas las ROMS para el core ZX Spectrum a la memoria SPI Flash. Se debe ejecutar desde una ROM en modo "root".
- **upgrade**
- **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".
- **zxuc**: Configura todas las opciones de la BIOS, permitiendo grabar en la SD 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

Se puede incorporar un módulo ESP 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
```



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

## 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-Uno 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 vídeos, se debe obtener la utilidad **makevideoradas** desde el **Repositorio SVN** (Usuario **guest**, contraseña **zxuno**).

En el caso de Windows, en el propio repositorio hay un ejecutable (**makevideoras.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.

## CP/M

En el ZX-Uno existe la posibilidad de usar el core Multicomp que es un CP/M 2.2, o bien correr el CP/M 3.0 de Spectrum con la ROM +3e y una SD especialmente preparada, pero solo se presentarán 51 columnas en pantalla, y es necesario estar intercambiando la parte izquierda y derecha para poder ver las 80 columnas, lo cual es una incomodidad grande.

Multicomp

[multicomp\\_z80\\_CPM.ZX1](multicomp_z80_CPM.ZX1) <https://obsolescence.wixsite.com/obsolescence/multicomp-fpga-cpm-demo-disk> <http://searle.x10host.com/Multicomp/index.html> <http://searle.x10host.com/Multicomp/cpm/fpgaCPM.html#UsingTheMachine>

+3

<https://www.zxuno.com/forum/viewtopic.php?f=12&t=1427&hilit=core#p16169>

<https://www.zxuno.com/forum/viewtopic.php?f=39&t=4099>

# 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 todos los cores de ZX Spectrum del ZX-Uno. 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:

```
./build-filesystem/rootfs
```

## Cómo utilizarlo

Hace falta que la tarjeta SD 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).

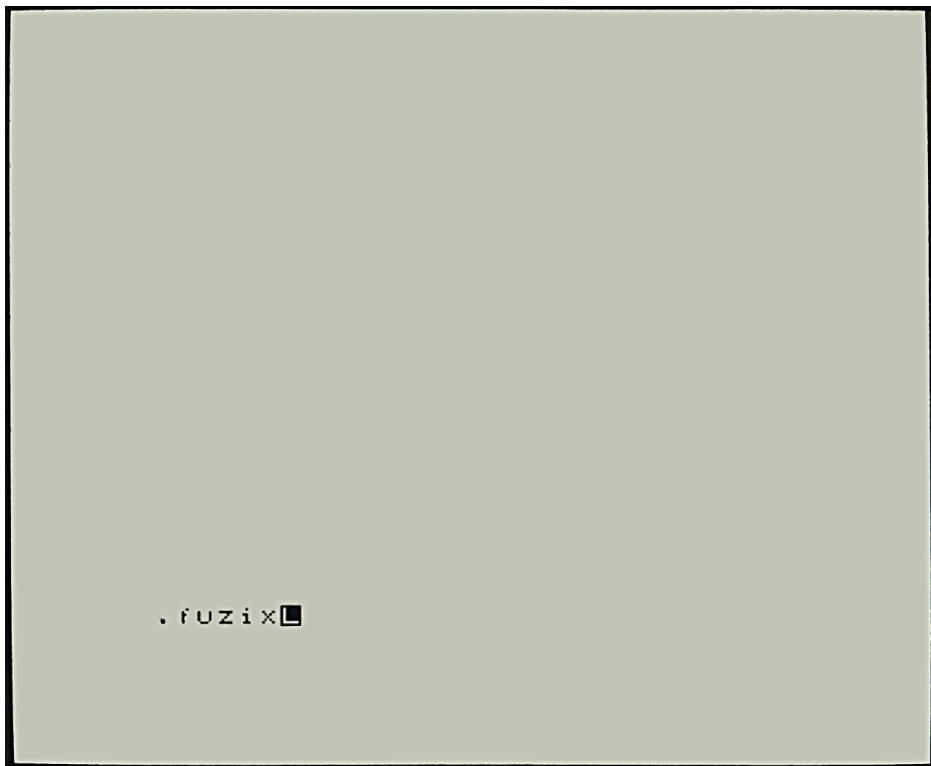
Volcar el contenido del sistema de archivos raíz en la partición [7E](#). Para ello, se puede usar la utilidad [dd](#), incluida en Linux, MacOS, etc. (y también [convertida para Windows](#)).

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
```

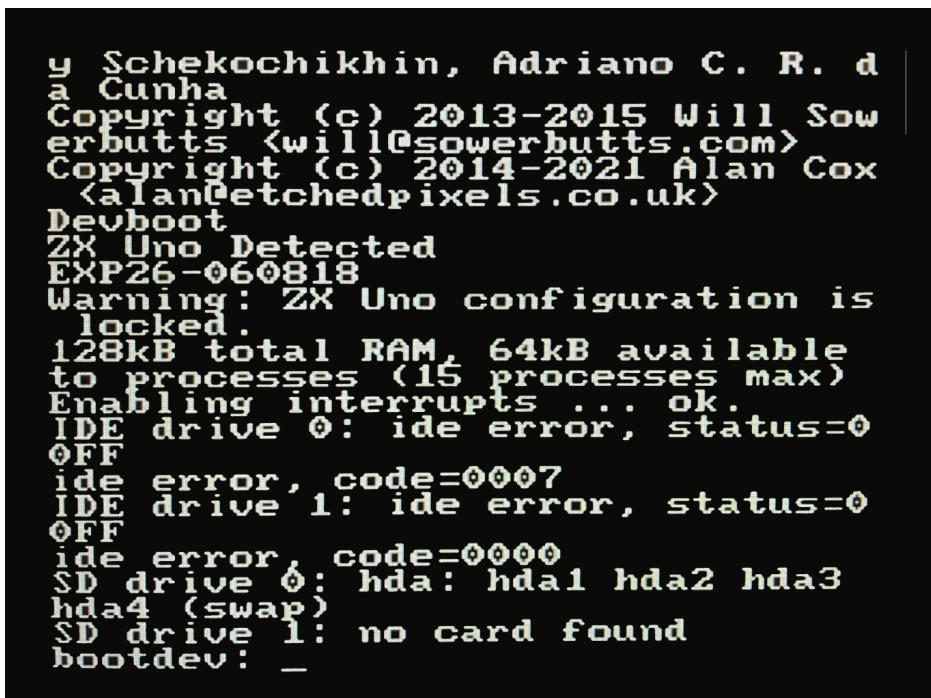
Copiar el fichero [FUZIX](#) dentro del directorio [BIN](#) y copiar [FUZIX.BIN](#) al directorio raíz de la particion esxdos.

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 SD y encontrar las particiones. En este ejemplo, la raíz está en la tercera partición de SD 0.



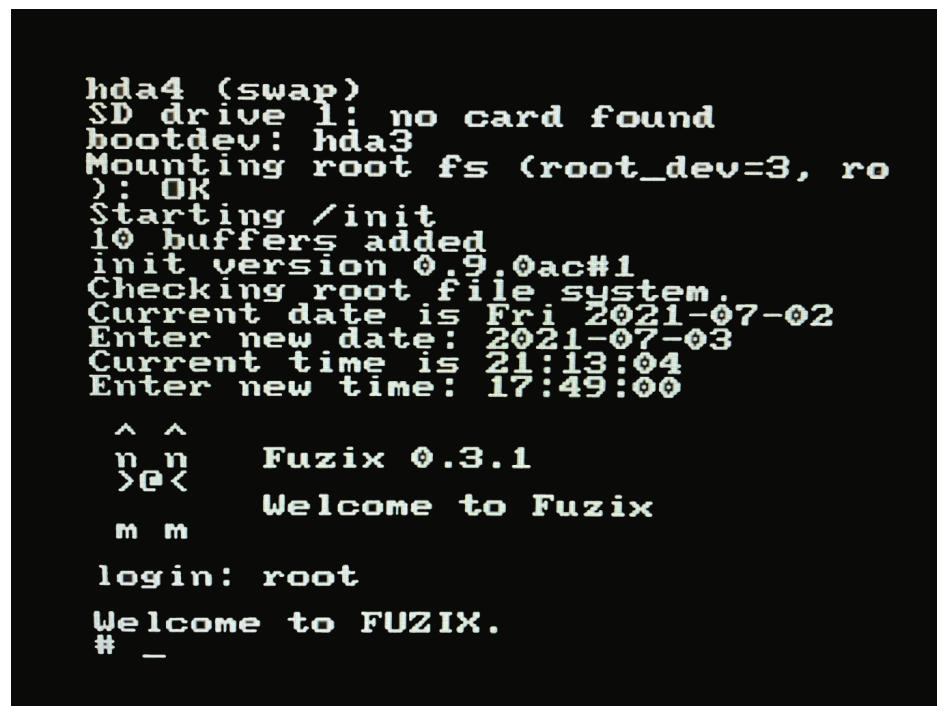
Escribir el identificador de la partición raíz (por ejemplo, **hd3**) y pulsar **Enter**.

```
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
 $\rangle$ : 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
 $\rangle$ : 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.ZX1**. 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-Uno tenga alimentación ininterrumpida (como un SAI o un USB de portatil con batería).

Copiar el fichero en la raíz de la tarjeta SD, 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.ZX1** y avisará cuando esté actualizado.

## ROMs

La memoria flash del ZX-Uno 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 SD. Arrancar el ZX-Uno 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-Uno pulsando la tecla / del teclado numérico \(Symbol Shift+V en ZXUnGo+\)](#), 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**.

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

## Cores

Hay un total de 45 espacios disponibles para almacenar cores, 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.ZX1`, donde `nn` es el número de espacio donde realizar la instalación (por ejemplo `CORE2.ZX1` para el espacio 2).



A partir de la versión 0.80 de BIOS, los ficheros se nombran usando la convención `COREXXy.ZX1` donde `XX` *siempre* es un número de dos dígitos. Así, un antiguo fichero `CORE4.ZX1` ha de renombrarse como `CORE04.ZX1`. La parte `y` del nombre se ignora, así que se pueden usar nombres más largos y descriptivos (como, por ejemplo, `CORE04_ejemplo.ZX1`).

Copiar el fichero en la raíz de la tarjeta SD, 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.ZX1`, ha de ser un fichero llamado `SPECTRUM.ZX1`.

## 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` sobre escribiendo los existentes (para preservar los comandos exclusivos de ZX-Uno).

Copiar `ESXMMC.BIN` (o `ESXMMC.ROM`, según la versión) en la raíz de la tarjeta SD.

Iniciar el ZX-Uno 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 `ESXMMC…`, 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-Uno 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 entera de la FPGA.

Copiar el archivo de imagen (de 16MiB) **FLASH.ZX1** en la raíz de la tarjeta SD.

Iniciar el ZX-Uno con la tarjeta insertada y pulsar **F2** 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 EXP27 sin PZX

Este core se basa en el core por defecto, pero añadiendo un chip de audio SAA1099 y sin soporte para PZX.

### 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:

- Specdrum
- Turbosound (dos chips AY) con posibilidad de elegir mix ACB/ABC
- DivMMC con esxdos 0.8.8
- Salida de vídeo compuesto/RGB y VGA 50Hz seleccionables vía teclado

### Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD 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

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [ZX](#). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

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

# ZX Spectrum 128K (Kyp)

[Core alternativo](#), cuyo objetivo es ser una implementación de un Spectrum 128K que sea lo más exacta posible en cuanto a la configuración de los tiempos (timing), contención de memoria, etc.

Sus características principales son:

- Specdrum
- Turbosound (dos chips AY) con posibilidad de elegir mix ACB/ABC
- DivMMC con esxdos 0.8.8

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD 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

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [ZX](#). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

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

## ZX Spectrum (Kyp/azesmbog)

Versión combinada, sintetizada por azesmbog, de los dos cores <<,#\_zx\_spectrum\_48k\_kyp,48K>> y [128K](#) de Kyp, añadiendo además también un modo Pentagon.

Sus características principales son:

- Specdrum
- Turbosound (dos chips AY) con posibilidad de elegir mix ACB/ABC
- Chip SAA1099
- DivMMC con esxdos 0.8.9
- Salida de vídeo compuesto/RGB y VGA 50Hz seleccionables vía teclado

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD con la primera partición en formato FAT16 o FAT32, y que tenga instalada la distribución de esxdos 0.8.9 (ver [el apartado correspondiente de esxdos](#) para más información).

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [ZX](#). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [Esc](#) ([Caps Shift+Espacio](#) en ZXUnGo+): BREAK
- [F5](#) ([Caps Shift+Symbol Shift+5](#) o [Caps Shift+Symbol Shift+Y](#) en ZXUnGo+): NMI
- [F8](#) ([Caps Shift+Symbol Shift+8](#) en ZXUnGo+): Alternar la configuración de mezcla de Turbosound entre ACB y ABC.
- [F10](#) ([Caps Shift+Symbol Shift+0](#) en ZXUnGo+): Modo Pentagon
- [F11](#) ([Caps Shift+Symbol Shift+Q](#) en ZXUnGo+): Modo 48K
- [F12](#) ([Caps Shift+Symbol Shift+W](#) en ZXUnGo+): Modo 128K
- [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#).
- [Ctrl+Alt+Supr](#) ([Caps Shift+Symbol Shift+F](#) y luego [N](#) en ZXUnGo+) o [F12](#) ([Caps Shift+Symbol Shift+W](#) en ZXUnGo+): Soft reset.

El core arranca por defecto en modo 128K.



Para ver el menú de opciones en modo 128K se debe usar el comando de esxdos:  
**.128.**

## ZX Spectrum TBBBlue

El firmware [TBBBlue](#), creado en 2016 por Victor Trucco y Fabio Belavenuto, lleva el nombre de dos de los creadores y el color de la máscara de soldadura de la placa base. Aunque inicialmente se pretendía usar el chip Z80 del ZX Spectrum original, el diseño se modificó posteriormente para usar una FPGA Xilinx Spartan-6, para permitir otras características avanzadas. Este core es la base sobre la que se construyó posteriormente el [ZX Spectrum Next](#).

El core TBBBlue más reciente para ZX-Uuno es una versión sintetizada por azesmbog sobre últimas versiones del repositorio y tiene, entre otras, las siguientes características:

- Salida de vídeo compuesto/RGB y VGA
- Soporte SD, vía DivMMC
- Modo turbo
- Modo Timex
- ULAPlus
- Soporte multiface
- Soporte para joysticks
- Turbosound
- ROM de Spectrum de arranque configurable
- Soporte para 3 chips de sonido AY (como en el ZX Spectrum Next), añadido por azesmbog
- Soporte para chip de sonido SID (como el que lleva el Commodore 64), añadido por azesmbog

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD con la primera partición en formato FAT16 o FAT32, opcionalmente, con la distribución de esxdos correspondiente a la configuración actual de BIOS (ver [el apartado correspondiente de esxdos](#) para más información).

Obtener la distribución de TBBlue correspondiente al core instalado, por ejemplo, consultando en [los foros de ZX-Uno](#). Normalmente la estructura necesaria para el core es una como la siguiente (en la raíz de la tarjeta SD):

```
/  
+-TBBLUE/  
| +-128.rom  
| (...)  
| +-config.ini  
| (...)  
| +-esxmmc.rom  
| (...)  
|  
+-TBBLUE.FW
```



En el caso de tener en la tarjeta SD instalado [esxdos](#), el archivo [esxmmc.rom](#) debe ser la ROM con el mismo número de versión.

Si no estuviera ya, [instalar el core TBBlue](#) en el ZX-Uno.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [ZX](#). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

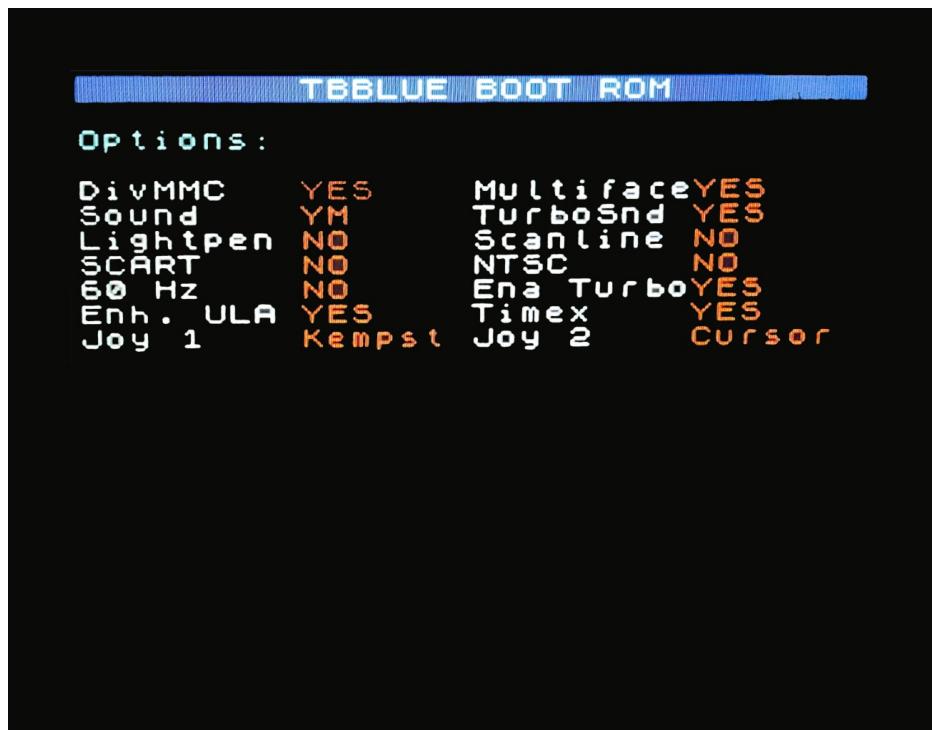
- [F1 \(Caps Shift+Symbol Shift+1 en ZXUnGo+\)](#): Hard Reset
- [F1 \(Caps Shift+Symbol Shift+1 en ZXUnGo+\)](#) y, a continuación, [Espacio](#): Menú de arranque
- [F2 \(Caps Shift+Symbol Shift+2 en ZXUnGo+\)](#): Activa o desactiva Scandoubler
- [F3 \(Caps Shift+Symbol Shift+3 en ZXUnGo+\)](#): Cambia entre 50Hz y 60Hz
- [F4 \(Caps Shift+Symbol Shift+4 en ZXUnGo+\)](#): Soft reset
- [F5 \(Caps Shift+Symbol Shift+5 en ZXUnGo+\)](#): Pausa CPU
- [F9 \(Caps Shift+Symbol Shift+9 en ZXUnGo+\)](#): Multiface
- [F10 \(Caps Shift+Symbol Shift+Q en ZXUnGo+\)](#): NMI

## Guía básica

Pulsando **F1** (**Caps Shift+Symbol Shift+1** en ZXUnGo+) y, a continuación, **Espacio**, se fuerza un reset del core y la carga del menú de arranque, donde se mostrarán las opciones de inicio existentes en el fichero **config.ini**.



A continuación, pulsando **E**, se accede al resto de opciones de configuración del core. Estas se pueden editar usando las teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en ZXUnGo+) y la tecla **Espacio**. Pulsar **Enter** para finalizar.



# Acorn Atom

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

Características principales:

- Soporte para cargar software con tarjeta SD
- Sólo VGA

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD 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 SD 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 SD. 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 ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **ATOM** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **J**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

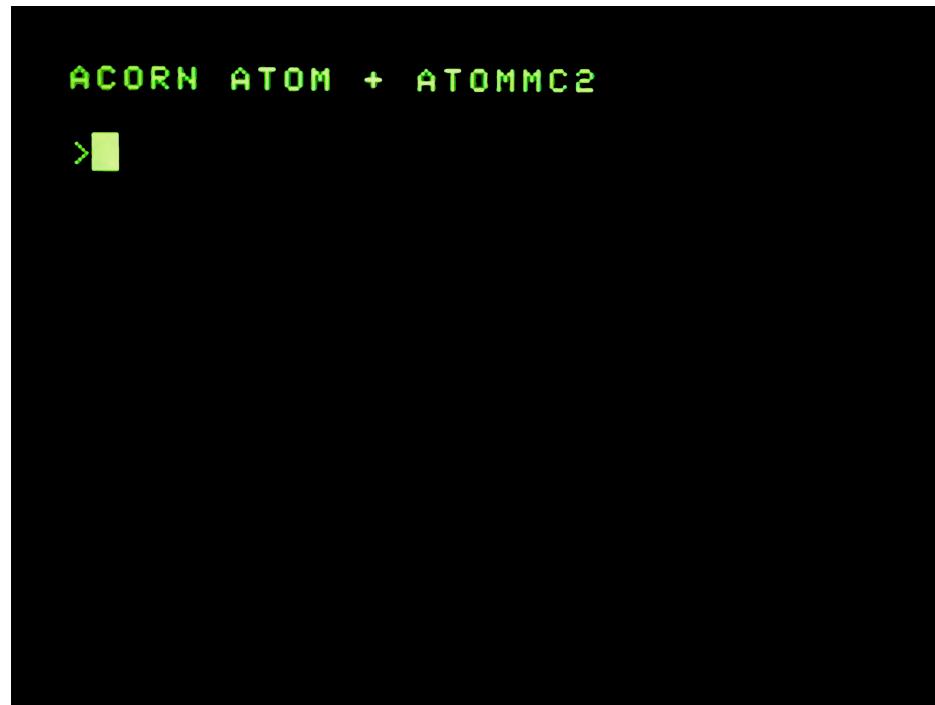
Durante la ejecución del core:

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

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



## Guía básica



Una vez iniciado, pulsar **Mayús+F10** (**Caps Shift+Symbol Shift+A** y, a continuación, **Caps Shift+Symbol Shift+0** en ZXUnGo+), 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.

# Acorn Electron

El [Acorn Electron](#) fue una versión barata del BBC Micro de Acorn Computers. El core está basado en el [trabajo original de David Banks \(hoglet\)](#).

Características principales:

- Salida de vídeo compuesto/RGB y VGA 50Hz seleccionables vía teclado
- Soporte SD/MMC, vía archivos de imagen ".MMB"
- Carga de software vía el puerto de entrada de audio del ZX-Uno
- Teclado PS/2 (mapeado a teclado inglés)

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD con la primera partición en formato FAT16 o FAT32. El core usa una ROM especial (Smart SPI) que lee de la SD un archivo que contiene imágenes de disquete.

El archivo debe llamarse **BEEB.MMB** y estar en el directorio raíz. Se puede crear con la utilidad [MMBImager](#) para Windows, disponible en [el repositorio SVN de ZX-Uno](#) (Usuario **guest**, contraseña **zxuno**) o con la utilidades MMB/SSD en perl, disponibles en [GitHub](#).

El fichero debe estar sin fragmentar en la SD. Se puede utilizar algún programa que defragmente ficheros o sistemas de archivos FAT (como [Defraggler para Windows](#)), o bien usar el siguiente método:

1. Formatear la primera partición de la SD en FAT16 o FAT32, pero **NO en formato rápido** (en Windows, desmarcar la casilla de formato rápido).
2. Copiar a la SD el archivo **BEEB.MMB**, de modo que este sea el **PRIMER archivo** que se copia a la SD.
3. Si se desea, ya se pueden copiar otros archivos a la SD para usar con otros cores, pero **SIEMPRE debe mantenerse el BEEB.MMB** como el primero que se copió a la SD.



Como el fichero **MMB** se llama igual que el que utiliza el [core de BBC Micro](#), se puede usar otro core con acceso completo a la tarjeta SD (Como uno de ZX Spectrum con esxDOS), teniendo dos ficheros **MMB** distintos, y renombrar el de Acorn Electron como **BEEB.MMB** antes de ejecutar este core

## Teclado

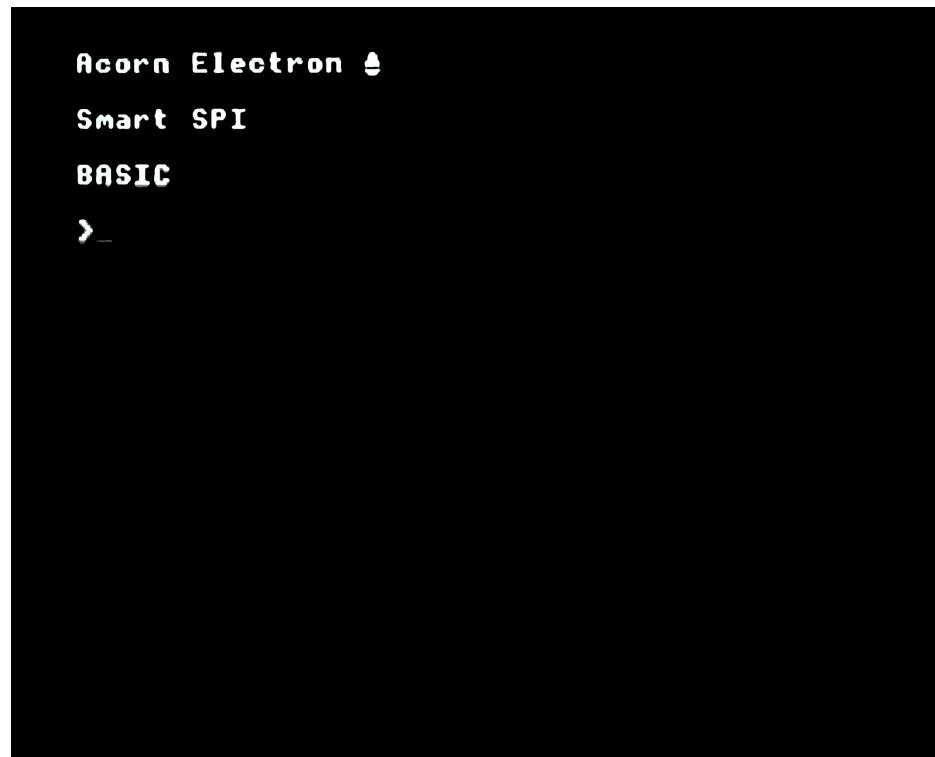
### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado ELECTRON (**Caps Shift+Symbol Shift+F**, luego **U** y luego **7**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Ctrl+Bloq May+1 - Ctrl+Bloq May+4** (se necesita un teclado externo PS/2): Configurar otros modos gráficos (60Hz, etc)
- **F10** (**Caps Shift+Symbol Shift+0** en ZXUnGo+) y **Ctrl+F10** (**Caps Shift+Symbol Shift+Z** y luego **Caps Shift+Symbol Shift+0** en ZXUnGo+): Soft Reset
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**.

## Guía básica



Una vez introducida la SD en el ZX-Uno y arrancado el core, si el archivo de **BEEB.MMB** está correctamente creado, al arrancar debería aparecer:

```
Acorn Electron  
Smart SPI  
BASIC  
>
```

Automáticamente se monta el disco 0 de la imagen, y se puede ver su contenido con el comando:

```
*CAT
```

Para cargar, por ejemplo, el menú de juegos que viene en algunas imágenes disponibles en internet, usar el comando:

```
CHAIN"MENU"
```

Para cargar desde la entrada de audio:

```
*TAPE  
CHAIN""
```

Y entonces comenzar la reproducción.

Para ver una lista de los discos disponibles en el fichero **BEEB.MMB**:

```
*DCAT
```

Para insertar un disco concreto en una unidad virtual concreta:

```
*DIN numerodisco numerounidad
```

# Amstrad CPC 464

El [Amstrad CPC 464](#) fue un ordenador doméstico creado y comercializado por la empresa británica Amstrad Consumer Plc a partir del año 1984. La versión para ZX-Uno ha sido [creada por McLeod](#).

Características del core:

- Amstrad CPC 464 completo: 64KB de RAM, 32Kb de ROM, interfaz de cassette, teclado y joystick.
- Soporte RGB/video compuesto y VGA (a 50Hz)
- Scanlines en modo VGA
- Soporta el joystick del primer jugador

La versión incluida con ZXUnCore tiene las siguientes mejoras:

- Soporte para el DAC de Vídeo RGB666
- Visualización de la imagen en Color, monocromo verde, monocromo ámbar o en blanco y negro  
Pulsando la tecla **Fin** ()



Existe una versión (Amstrad CPC6128) de este core. sintetizada por Jepalza. que añade la paginación de memoria 128K. Sería como un CPC6128 al que no le funcionase la unidad de discos

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

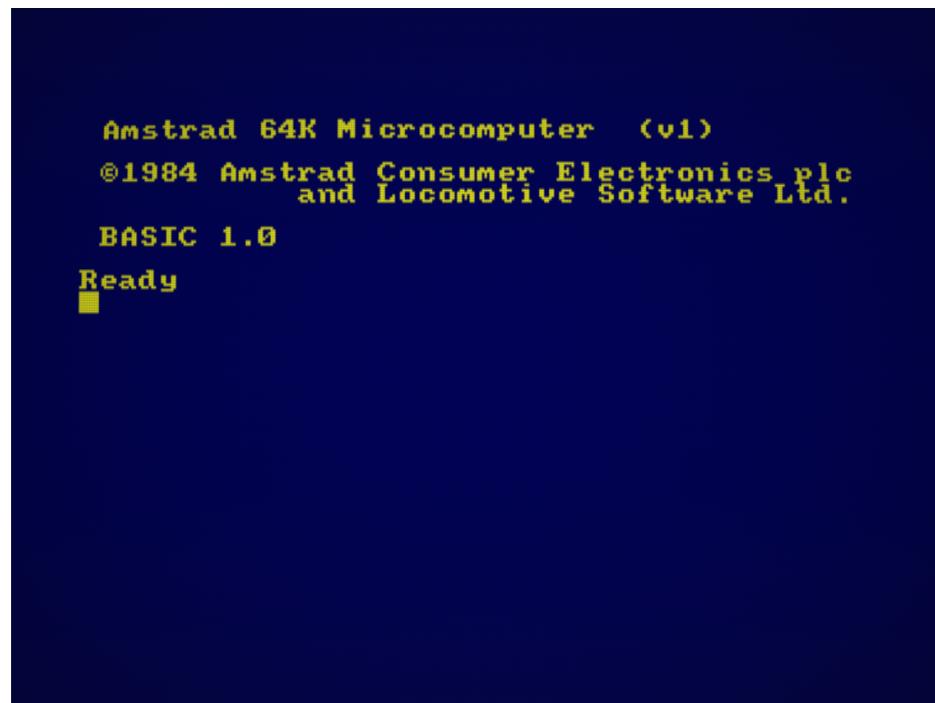
### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **CPC** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **2**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Supr** (**Caps Shift+Symbol Shift+V** en ZXUnGo+): **CLR**.
- **Impr. Pant** o **Windows Izquierda** (**Caps Shift+Symbol Shift+T** en ZXUnGo+): **COPY**
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**.
- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+F** y luego **N** en ZXUnGo+): Soft reset.
- **Fin** (**Caps Shift+Symbol Shift+B** en ZXUnGo+): Alterna la visualización entre color, monocromo verde, monocromo ámbar o blanco y negro

## Guía básica



Desde BASIC, se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando **RUN"**. Al contrario que en la máquina original, durante la reproducción se puede escuchar el audio de la cinta.

# Amstrad CPC 6128

El [Amstrad CPC 6128](#) fue un ordenador doméstico, sucesor del Amstrad CPC 664 (que solo duró 6 meses en el mercado), y este, a su vez, era sucesor del Amstrad CPC 464.

El core para ZX-Uno 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 SD

Se debe de utilizar una tarjeta SD 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 se debe 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 ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **CPC** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **2**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Re Pág** (**Caps Shift+Symbol Shift+J** en ZXUnGo+): 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** (**Caps Shift+Symbol Shift+J** en ZXUnGo+) para hacer reset y cargar el siguiente archivo **DSK** en orden alfabético.

# Apple I

El [Apple I](#) fue uno de los primeros computadores personales, y el primero en combinar un microprocesador con una conexión para un teclado y un monitor.

El core de ZX-Uno ha sido creado por Subcritical, basado en el proyecto [Apple-One](#).

Algunas de sus características son:

- Teclado PS/2 (mapeado a teclado inglés)
- Sólo funciona con la salida VGA
- Incluye Integer Basic

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

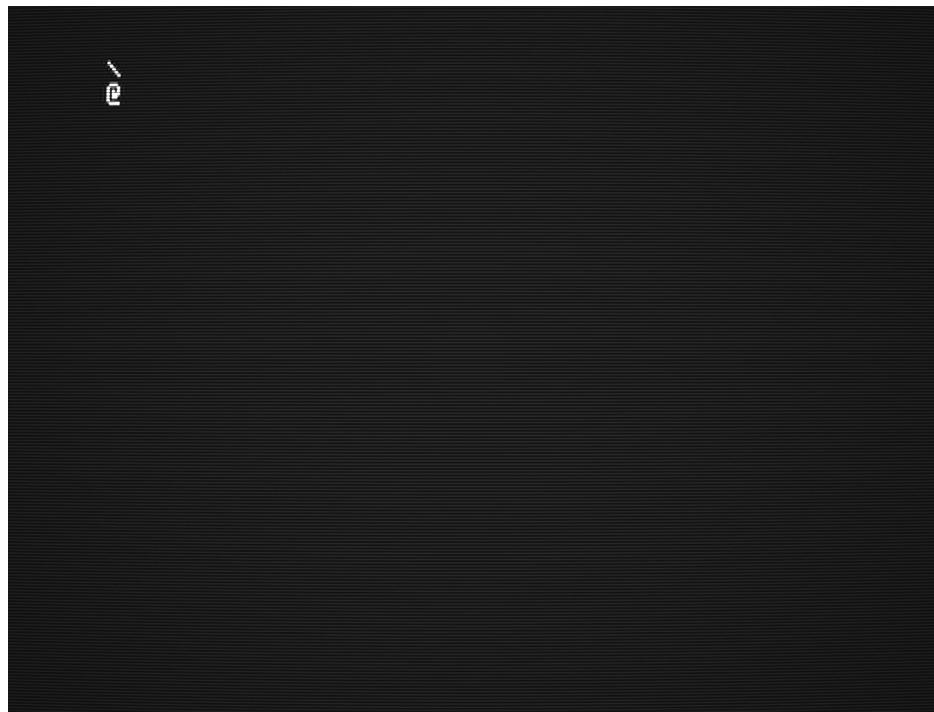
El teclado está con la distribución en inglés.

Las combinaciones específicas de ZXUnGo+ que se indiquen se corresponden con el modo de teclado **APPLE** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **8**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Notar que, en Apple I, no existe tecla de borrado. Debido a la forma en que se muestra el vídeo, no es posible hacer que el cursor retroceda para borrar un carácter, así que se utiliza uno especial: **\_**. Este significa que se ha retrocedido para borrar en la entrada de datos (por ejemplo **PRN\_INT** es interpretado como **PRINT**).

## Guía básica

Según se inicia, el Apple I arranca con [Woz Monitor](#) (ROM cargada en `0xFF00`)



El intérprete BASIC incluido con el core de Apple I es [Integer BASIC](#). Para iniciararlo, se debe invocar la dirección de memoria correspondiente ([0xE000](#)):

```
\  
E000R
```

Una vez cargado el intérprete, la consola cambia para indicar que está lista para recibir [comandos de BASIC](#):

```
\  
E000R  
  
E000:4C  
>@
```

```
\  
E000R  
E000: 4C  
>10 PRINT "HELLO"  
>RUN
```

## Apple ][

La familia de computadores [Apple II](#) fue la primera serie de microcomputadoras de producción masiva hecha por la empresa Apple Computer.

El core de ZX-Uno está basado en el [original de Stephen A. Edwards](#) y en [la adaptación de vlast a las placas Papilio](#).

Algunas de sus características son:

- Soporta Joystick (hasta dos botones de disparo)
- Tarjetas de expansión de RAM. 128K Saturn RAM (slot 5) + 16K Language card (slot 0).
- Scanlines en modo VGA
- Cambio de tipo de monitor entre color y blanco y negro

Para más información consultar [el foro oficial de ZX-Uno](#).

## Formato de Tarjeta SD

La tarjeta SD ha de ser en un formato exclusivo, y por tanto no puede ser utilizada con otros cores. Está basada en imágenes de disco **NIB** concatenadas.

Para convertir imágenes de disco en otro formato (**DSK** o **D0**), se puede utilizar la utilidad **dsk2nib**, disponible en [el repositorio SVN de ZX-Uno](#) (Usuario **guest**, contraseña **zxuno**) y en [GitHub](#).



Este proceso borra el contenido que hubiera antes en la tarjeta SD, y no se puede deshacer.

## Windows

Concatenar las imágenes (máximo 20) usando **COPY**:

```
COPY /B imagen1.nib + imagen2.nib + (...) + imagen20.nib apple2_20discos.img
```

Volcar la imagen en la SD, por ejemplo, usando [HDD Raw Copy Tool](#).

## MacOS y Linux

Concatenar las imágenes (máximo 20) usando **cat**:

```
cat imagen1.nib imagen2.nib (...) imagen20.nib > apple2_20discos.img
```

Volcar la imagen en la SD, usando **dd**:

```
sudo umount /dev/...
sudo dd if=apple2_20discos.img of=/dev/...
```

## Teclado



En algunas versiones de este core, el teclado no se inicializa correctamente si sólo hay conectado un teclado de membrana. Es necesario iniciar el ZXUnGo+ (en frío) con un teclado PS/2 conectado, a la vez, para poder utilizarlo.

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **APPLE** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **8**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- - (teclado numérico) (**Caps Shift+Symbol Shift+R** en ZXUnGo+): Activar o desactivar scanlines en salida VGA.
- \* (teclado numérico) (**Caps Shift+Symbol Shift+E** en ZXUnGo+): Alternar entre monitor en color y monitor en blanco y negro.
- **F1** a **F10** (**Caps Shift+Symbol Shift+1** a **Caps Shift+Symbol Shift+0** en ZXUnGo+): Insertar la imagen de disco 1 a 10 en la tarjeta SD. Pulsar **F12** a continuación.
- **Mayús+F1** (**Caps Shift+Symbol Shift+A** y luego **Caps Shift+Symbol Shift+1** en ZXUnGo+) a **Mayús+F10** (**Caps Shift+Symbol Shift+A** y luego **Caps Shift+Symbol Shift+0** en ZXUnGo+): Insertar la imagen de disco 11 a 20 en la tarjeta SD. Pulsar **F12** a continuación.
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**.
- **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+): Soft reset y carga del disco insertado actualmente.



El core necesita tener insertada una tarjeta SD con el formato correcto, o si no, no finaliza el arranque. Esto es porque el Apple ][ necesita un disco de arranque para funcionar.

# Arcades

Originalmente pensados para el [Addon Jamma para ZX-Uno](#) (para conectar en una máquina recreativa), se trata de cores de [máquinas recreativas \(o Arcade\)](#). Posteriormente también se crearon versiones adaptadas para ser compatibles con joysticks (por ejemplo con el addon VGA+DB9 para ZX-Uno).

Existen principalmente tres tipos:

- Verticales (la máquina original usaba un monitor girado 90º)
- Verticales invertidos (la máquina usaba un monitor girado 270º)
- Horizontales (la máquina original usaba un monitor en horizontal, sin girar)

En el [hilo original del foro de ZX-Uno](#) se pueden obtener enlaces de descarga para las distintas versiones e información más detallada de cada uno de los mismos.

Señalar que existe una versión vertical de BIOS, y una [ROM de Spectrum](#) pensada para poder cargar directamente los distintos cores Arcade.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo **ZX** normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

En general, casi todos los cores tienen unas teclas y botones de control similares.

Durante la ejecución:

- **1 y 2**: Botones de Jugador 1 y Jugador 2
- **3 y 4**: Introducir moneda
- Teclas de cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en ZXUnGo+) o palanca de joystick: Joystick
- **Z y X** (o botones de disparo de joystick): Botones de disparo
- **0**: En cores verticales, activa o desactiva giro de 90º en las direcciones de los controles
- **F10** (**Caps Shift+Symbol Shift+0** en ZXUnGo+): Soft Reset
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

# Atari 800XL

El [Atari 800XL](#) era un ordenador personal lanzado por Atari en la década de los 80.

La versión para ZX-Uno tiene estas características:

- Memoria ampliada hasta 320K, seleccionable
- Soporte de unidad de disco (hasta 4 unidades), mediante tarjeta SD
- Soporte de cartuchos, también mediante carga por SD
- Soporte para vídeo compuesto y VGA
- Scanlines en modo VGA
- Soporte de joystick norma Atari

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, con la primera partición en formato FAT32. En la raíz de la tarjeta debe haber un directorio llamado [atari800](#), con dos subdirectorios: [rom](#) con distintas ROM a utilizar (ej: [ATARIXL.ROM](#)), y [user](#) con ficheros de disco, cartucho, etc. (ej: [ManicMin.xex](#))

Si no estuviera ya, [instalar el core de Atari 800XL](#) en el ZX-Uno.

## Teclado

### Teclas especiales y botones

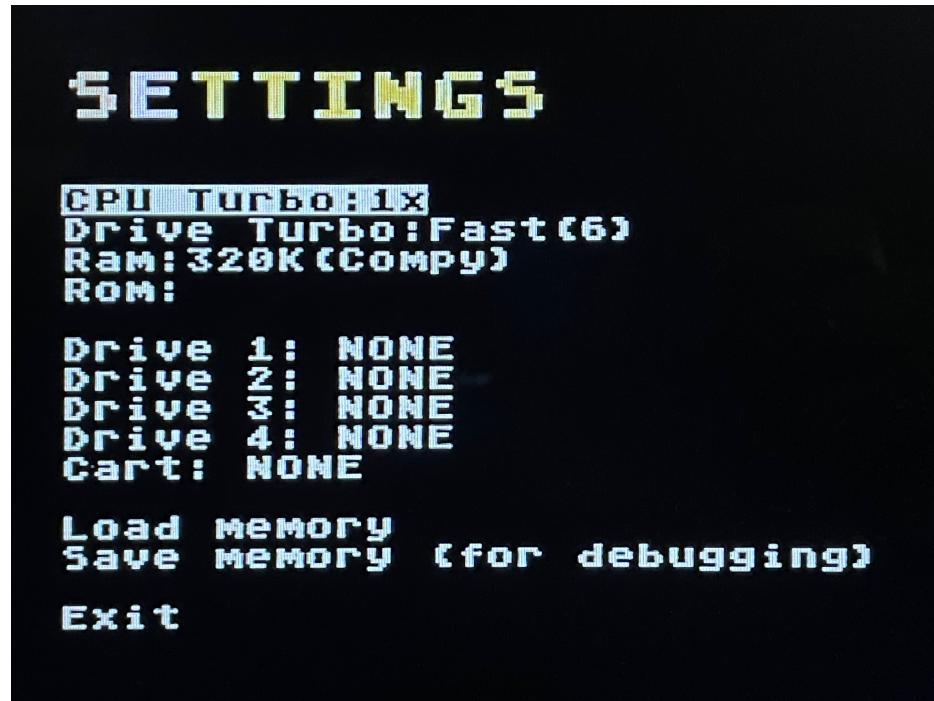
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **ATARI** ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [5](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- - ([Caps Shift+Symbol Shift+R](#) en ZXUnGo+): Activar o desactivar scanlines en modo VGA
- \* ([Caps Shift+Symbol Shift+E](#) en ZXUnGo+): Alternar el tipo de máquina entre PAL y NTSC
- [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)
- [F5](#) ([Caps Shift+Symbol Shift+5](#) en ZXUnGo+): [Help](#)
- [F6](#) ([Caps Shift+Symbol Shift+6](#) en ZXUnGo+): [Start](#)
- [F7](#) ([Caps Shift+Symbol Shift+7](#) en ZXUnGo+): [Select](#)
- [F9](#) ([Caps Shift+Symbol Shift+9](#) en ZXUnGo+): [Reset](#)
- [F11](#) ([Caps Shift+Symbol Shift+Q](#) en ZXUnGo+): Cargar un disco
- [F12](#) ([Caps Shift+Symbol Shift+W](#) en ZXUnGo+): Acceder al menú de opciones
- El teclado numérico emula un joystick. Las teclas [5](#) y [2](#) sirven indistintamente para la dirección [abajo](#) y [0](#) es el disparo

## Guía básica

Pulsando **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+) se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+) y **Enter** (o bien el joystick y el botón de disparo) para elegir y seleccionar en el menú.



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

- Acelerar la velocidad de la CPU (CPU Turbo)
- Acelerar la velocidad de lectura de los discos (Drive Turbo)
- Cambiar el tipo de Ram
- Cargar una Rom desde la SD
- Cargar un disco en la unidad 1 (Drive 1)
- Cargar un disco en la unidad 2 (Drive 2)
- Cargar un disco en la unidad 3 (Drive 3)
- Cargar un disco en la unidad 4 (Drive 4)
- Cargar un cartucho (Cart)
- Cargar la memoria desde una grabación anterior (Load Memory)
- Guardar el estado de la memoria (Save memory)
- Salir (Exit)

# Atari 2600

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

La versión para ZX-Uno está desarrollada por Quest y DistWave..

Algunas de las características del core son:

- Dos modos de vídeo seleccionables: RGB y VGA
- Compatible con joystick

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, 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-Uno.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **ATARI** ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [5](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **W, A, S, D** (personalizables) o el joystick 1: Controles de dirección del jugador 1
- **F** (personalizable) o el botón de disparo del joystick 1: Disparo del jugador 1
- **I, J, K, L** (personalizables) o el joystick 2: Controles de dirección del jugador 2
- **H** (personalizable) o el botón de disparo del joystick 2: Disparo del jugador 2
- **Ctrl+Alt+Backspace** ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)

## Personalización de los controles

Los controles por defecto del core con soporte de teclado se pueden modificar. Para ello, se ha de crear en la raíz de la tarjeta SD un fichero llamado **KEYSP1** para el primer jugador y **KEYSP2** para el segundo (se puede descargar uno de ejemplo desde [el foro de ZX-Uno](#)).

Estos ficheros se pueden crear con un editor hexadecimal (como [HxD para Windows](#)) y están formados por una secuencia de bytes indicando en este orden: **Arriba**, **Abajo**, **Izquierda**, **Derecha**, **Botón de disparo**. Los códigos a utilizar en la secuencia se pueden consultar en la tabla [Scan Codes](#) al final del manual (Columna **MAKE**), pero, teniendo en cuenta que, si el código está emparejado con **0xE0**, se ha de usar el otro valor, sumándole **0x80** (por ejemplo, para **E0 14**, sería **0x94**).



Por ejemplo, la secuencia **15 1C 44 4D 29** se correspondería con **Q A O P Espacio**

## Guía básica

Pulsando **Esc** ([Caps Shift+Espacio](#) en ZXUnGo+) o el botón 2 del joystick se muestra el menú de configuración. Se usan las teclas de cursor ([Caps Shift+6](#) y [Caps Shift+7](#) en ZXUnGo+) 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)
- Color (Color)
- Dificultad A (Difficulty A)
- Dificultad B (Difficulty B)
- Select
- Start
- Cargar ROM (Load ROM)
- Salir del menú (Exit)

# BBC Micro

El [BBC Micro](#), fue uno de los primeros ordenadores domésticos, diseñado y construido por Acorn Computers para la British Broadcasting Corporation (BBC).

El core para ZX-Uno ha sido creado por Quest y mejorado posteriormente por azesmbog y hoglet.

Algunas de sus características son:

- Salida vídeo RGB y VGA (con scanlines opcionales)
- Soporte SD/MMC, vía archivos de imagen [.MMB](#)
- Teclado PS/2 (mapeado a teclado inglés)
- Soporte de joystick en el puerto 1 (Emula Joy analógico)
- Implementación del chip de sonido sn76489 desde el [proyecto PACE](#) (Programmable Arcade Circuit Emulation)

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD con la primera partición en formato FAT16 o FAT32. El core usa una ROM especial (MMFS en los cores más recientes, Smart SPI en los más antiguos) que lee de la SD un archivo [BEEB.MMB](#), que contiene imágenes de disquete.

El archivo debe llamarse exactamente [BEEB.MMB](#) y estar en el directorio raíz. Se puede crear con la utilidad [MMBImager](#) para Windows, disponible en [el repositorio SVN de ZX-Uno](#) (Usuario [guest](#), contraseña [zxuno](#)) o con la utilidades MMB/SSD en perl, disponibles en Github [aquí](#) o [aquí](#).

El fichero [MMB](#) debe estar sin fragmentar en la SD. Se puede utilizar algún programa que defragmente ficheros o sistemas de archivos FAT (como [Defraggler para Windows](#)), o bien usar el siguiente método:

1. Formatear la primera partición de la SD en FAT16 o FAT32, pero **NO en formato rápido** (en Windows, desmarcar la casilla de formato rápido).
2. Copiar a la SD el archivo [BEEB.MMB](#), de modo que este sea el **PRIMER archivo** que se copia a la SD.
3. Si se desea, ya se pueden copiar otros archivos a la SD para usar con otros cores, pero **SIEMPRE debe mantenerse el BEEB.MMB** como el primero que se copió a la SD.



Como el fichero [MMB](#) se llama igual que el que utiliza el [core de Acorn Electron](#), se puede usar otro core con acceso completo a la tarjeta SD (Como uno de ZX Spectrum con esxDOS), teniendo dos ficheros [MMB](#) distintos, y renombrar el de BBC Micro como [BEEB.MMB](#) antes de ejecutar este core

## Teclado



### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado BBC ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [6](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- - del teclado numérico ([Caps Shift+Symbol Shift+R](#) en ZXUnGo+): Activa o desactiva scanlines
- **F12** ([Caps Shift+Symbol Shift+W](#) en ZXUnGo+): Reset
- **Mayús+F12** ([Caps Shift+Symbol Shift+A](#) y luego [Caps Shift+Symbol Shift+W](#) en ZXUnGo+): Soft Reset intentando cargar automáticamente el disco seleccionado en **BEEB.MMB**
- **Ctrl+Alt+Backspace** ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#).

## Guía básica



Una vez introducida la SD en el ZX-Uno y arrancado el core, si el archivo de **BEEB.MMB** está correctamente creado, al arrancar debería aparecer:

```
BBC Computer 32k
Model B MMFS
BASIC
>
```

O bien, si es un core con Smart SPI

```
BBC Computer 32k
Smart SPI
BASIC
```

Automáticamente se monta el disco 0 de la imagen, y se puede ver su contenido con el comando:

```
*CAT
```

Para cargar, por ejemplo, un programa llamado **MENU**, y que esté en el disco usar el comando:

```
*MENU
```

Para ver una lista de los discos disponibles en el fichero **BEEB.MMB**:

\*DCAT

Para insertar un disco concreto:

\*DIN numerodisco



Recordar, tras insertar un disco, que si este tiene arranque automático, se puede iniciar pulsando **Mayús+F12**.

# 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-Uno ha sido [desarrollada por Kyp069](#) y tiene las siguientes características:

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

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

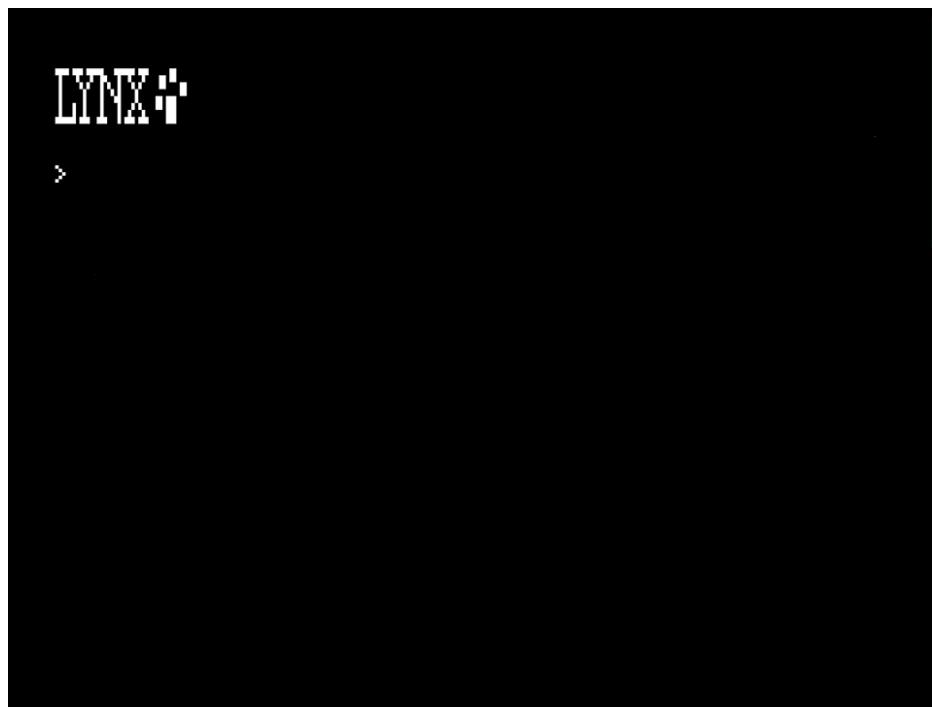
### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado LYNX ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [M](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [F6 \(Caps Shift+Symbol Shift+6 en ZXUnGo+\)](#): Alterna entre el modo 48kB y el modo 96kB (por defecto)
- [F7 \(Caps Shift+Symbol Shift+7 en ZXUnGo+\)](#): Alterna entre activar o desactivar la ROM Scorpio (activa por defecto)
- [F8 \(Caps Shift+Symbol Shift+8 en ZXUnGo+\)](#): 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.- [Ctrl+Alt+Supr](#): Reset
- [F11 \(Caps Shift+Symbol Shift+Q en ZXUnGo+\)](#): Reset
- [Ctrl+Alt+Backspace \(Caps Shift+Symbol Shift+F y luego B en ZXUnGo+\)](#): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [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**.

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** desde ficheros **TAP** de Lynx.

# 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-Uno 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 SD

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



Para algunas versiones antiguas de este core, la detección de teclado no se inicializa correctamente si sólo hay conectado un teclado de membrana. Es necesario iniciar el ZXUnGo+ (en frío) con un teclado PS/2 conectado, a la vez, para poder utilizarlo.

## Teclas especiales y botones

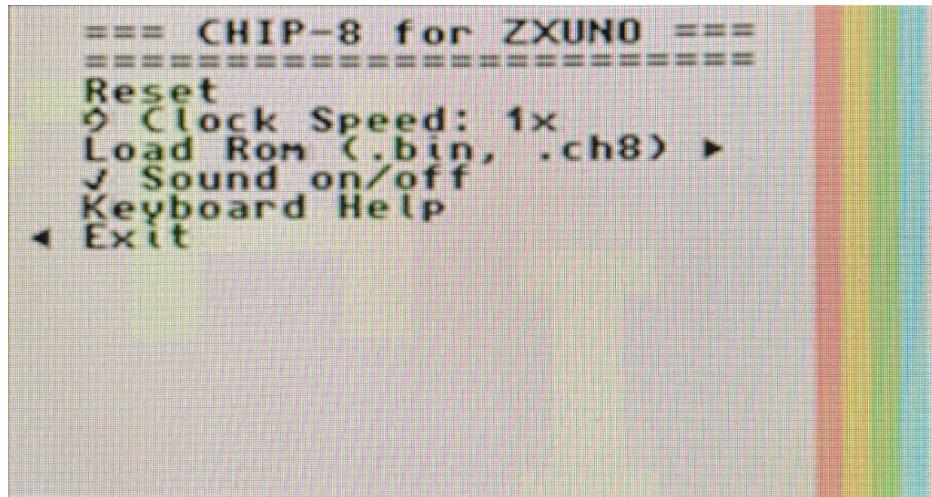
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo **ZX** normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

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

## Guía Básica

Pulsando **Esc** (**Caps Shift+Espacio** en ZXUnGo+) se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+), o bien **W** y **S**, 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 SD (Load Rom)
- Activar o desactivar el sonido (Sound On/Off)
- Ayuda sobre el uso del teclado (Keyboard Help)
- Salir del menú (Exit)

# ColecoVision

[ColecoVision](#) es una consola de videojuegos lanzada al mercado por la empresa Coleco.

La versión para ZUNO+ está basada el proyecto [de Fabio Belavenuto](#).

Algunas de las características del core son:

- La ROM de la BIOS se carga desde la tarjeta SD
- Soporta ROM multicartucho, que también se carga desde la SD
- Soporte para Joystick

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, 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 ZX-Uuno.

## Teclado

### Teclas especiales y botones

Durante la ejecución del core:

- Cursor ([Caps Shift+5](#), [Caps Shift+6](#), [Caps Shift+7](#) y [Caps Shift+8](#) en ZXUnGo+) 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' ([Caps Shift+Espacio](#) en ZXUnGo+): 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 ZXUnGo+) se reinicia el core y se vuelve a cargar el menú de selección de ROM.

# Colour Genie

El [EG 2000 Colour Genie](#) era un ordenador personal fabricado por la empresa EACA en la década de los 80.

La versión para ZX-Uno ha sido [desarrollada por Kyp069](#) y tiene las siguientes características:

- 32K de RAM
- Modos de texto y gráficos
- Carga de cinta usando la entrada de audio

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

El teclado está con la distribución en inglés. No está implementada la tecla [Clear](#).

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [eg2000](#) ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [L](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [Windows](#) ([Caps Shift+Symbol Shift+X](#) en ZXUnGo+): [MODSEL](#)
- [F5](#) ([Caps Shift+Symbol Shift+5](#) en ZXUnGo+): [NMI](#)
- [F11](#) ([Caps Shift+Symbol Shift+Q](#) en ZXUnGo+) o [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)
- [F12](#) ([Caps Shift+Symbol Shift+W](#) en ZXUnGo+) o [Ctrl+Alt+Supr](#) ([Caps Shift+Symbol Shift+F](#) y luego [N](#) en ZXUnGo+): Reset

## Guía básica

En el arranque, aparece una pregunta indicando el tamaño de memoria. Normalmente, basta con pulsar **Enter** para entrar al intérprete de BASIC.

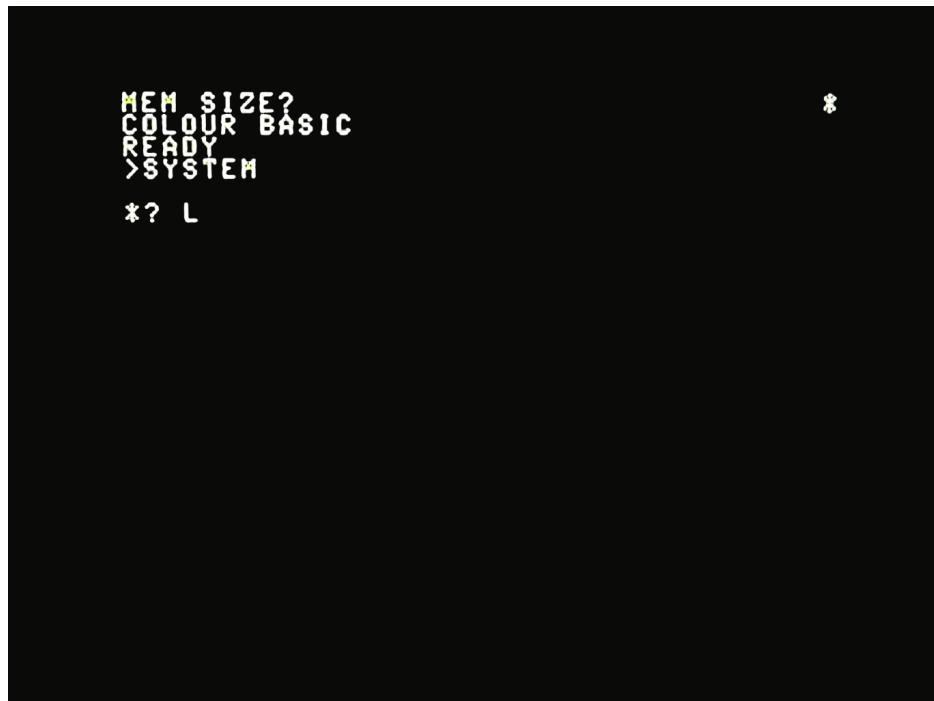


Desde BASIC, se puede cargar desde una cinta (u otro dispositivo externo de audio) con los comandos siguientes.

Para cargar ficheros de BASIC, usar **CLOAD** y **Enter**. Una vez haya terminado la carga, usar el comando **RUN**.

Para cargar código máquina, usar **SYSTEM** y **Enter**. A continuación, escribir la primera letra del nombre del programa y **Enter** otra vez. Si tras la carga el programa no comenzase automáticamente, escribir **/** y **Enter**.

Durante la carga, aparecen alternativamente uno **y dos \*** en la parte superior derecha de la pantalla.



## Commodore 16

El [Commodore 16](#) fue un ordenador doméstico fabricado por la empresa Commodore International en 1984.

El core de ZX-Uno está basado en el [proyecto FPGATED de István Hegedus](#), con algunos pequeños cambios y mejoras, como la lectura de cassetes reales por audio.

Características:

- Implementa un Commodore 16 PAL ampliado a 64K de RAM
- Disquetera 1541, redirigida a SD y en RAW. Sólo lectura (compatible con imágenes .D64)
- Chip TED de FPGATED
- Soporte de joystick, tanto real (conector DB9) como emulado en teclado numérico
- Switch VGA 50Hz / RGB-Compuesto
- Scanlines en modo VGA
- Carga de cinta usando la entrada de audio
- Cambio de polaridad de la señal de entrada de audio
- ROM Kernal PAL -5 modificada para evitar la espera de pulsación tras la cabecera en las cargas de cassette
- Mezcla de audio de salida compuesta por el feedback de audio del cassette (más bajo) y el audio principal del chip TED
- LED testigo de lectura de la 1541 y también del estado de la polaridad de EAR

## Formato de Tarjeta SD

La tarjeta SD ha de ser en un formato exclusivo, y por tanto no puede ser utilizada con otros cores. Está basada en imágenes de disco [D64](#) concatenadas, en bloques de 256K. Descargar el fichero [dummyto256.bin](#) disponible en [el repositorio oficial de ZX-Uno](#).

Para incluir varios ficheros [PRG](#) en una imagen de disco [D64](#), se puede utilizar el [programa DirMaster](#) para Windows, creando una imagen cuyo primer programa sea [FB16.PRG](#) (más detalles en [el foro de ZX-Uno](#)).



El formato RAW de SD es compatible con el que usa también el core de Commodore 64, así que se puede utilizar una misma tarjeta para los dos, incluyendo imágenes de disco con programas para los dos sistemas.



Este proceso borra el contenido que hubiera antes en la tarjeta SD, y no se puede deshacer.

### Windows

Concatenar las imágenes usando [COPY](#):

```
COPY /B imagen1.d64 + dummyto256.bin + imagen2.d64 + dummyto256.bin + (...)  
c16rawsd.img
```

Volcar la imagen en la SD, por ejemplo, usando [HDD Raw Copy Tool](#).

### MacOS y Linux

Concatenar las imágenes usando [cat](#):

```
cat imagen1.d64 dummyto256.bin imagen2.d64 dummyto256.bin (...) > c16rawsd.img
```

Volcar la imagen en la SD, usando [dd](#):

```
sudo umount /dev/...  
sudo dd if=c16rawsd.img of=/dev/...
```

Si no estuviera ya, [instalar el core de Commodore 16](#) en el ZX-Uno.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **C16** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **E**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Esc** (**Caps Shift+Espacio** en ZXUnGo+): Esc
- **Tab**: RUN/STOP
- **Windows izquierda** (**Caps Shift+Symbol Shift+x** en ZXUnGo+): = Commodore
- Teclado Numérico: Joystick emulado
- **F1** (**Caps Shift+Symbol Shift+1** en ZXUnGo+) a **F3** (**Caps Shift+Symbol Shift+3** en ZXUnGo+): F1 a F3
- **Bloq. Num** (**Caps Shift+Symbol Shift+I** en ZXUnGo+) o **F4** (**Caps Shift+Symbol Shift+4** en ZXUnGo+): HELP
- **Insert** (**Caps Shift+Symbol Shift+G** en ZXUnGo+): Seleccionar el primer disco de la tarjeta SD
- **Re.Pag** (**Caps Shift+Symbol Shift+J** en ZXUnGo+): Seleccionar el siguiente disco
- **Av.Pag** (**Caps Shift+Symbol Shift+N** en ZXUnGo+): Seleccionar el disco anterior
- **Ctrl+Re.Pag** (**Caps Shift+Symbol Shift+Z** y luego **Caps Shift+Symbol Shift+J** en ZXUnGo+): Avanzar 10 discos
- **Ctrl+Av.Pag** (**Caps Shift+Symbol Shift+Z** y luego **Caps Shift+Symbol Shift+N** en ZXUnGo+): Retroceder 10 discos
- + del teclado numérico (**Caps Shift+Symbol Shift+D** en ZXUnGo+): Cambiar la polaridad de la entrada de audio
- - del teclado numérico (**Caps Shift+Symbol Shift+R** en ZXUnGo+): Activar o desactivar scanlines en modo VGA
- **F11** (**Caps Shift+Symbol Shift+Q** en ZXUnGo+): Seleccionar la asignación del joystick al puerto 0 o al puerto 1
- **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+): Reset
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

## Guía básica



Desde BASIC, se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando **LOAD**. Una vez finalizada la carga, escribir **RUN** y pulsar **Enter** si fuese necesario.



Es posible utilizar programas como [Audiotap](#) o [Comodoro](#) con los ficheros **TAP** de cinta de Commodore.

Para listar el directorio del disco seleccionado, pulsar **F3** (**Caps Shift+Symbol Shift+3** en ZXUnGo+) o introducir el comando **DIRECTORY**. Para cargar un fichero del disco, usar el comando **DLOAD"<nombre de fichero>"** y luego, normalmente, **RUN**.

Para cargar rápidamente el primer fichero **PRG** de un disco pulsar **Mayús+TAB** o usar el comando **DLOAD"\***, y luego, normalmente, **RUN**.

## Commodore 64

Commodore 64 (C64, CBM 64/CBM64, C=64,C-64, VIC-641) era una [computadora doméstica de 8 bits](#) que fue desarrollada por Commodore International.

El core de ZX-Uno está desarrollado por Quest.

Características:

- Commodore 64 PAL con 64K de RAM
- Disquetera 1541 completa, redirigida a SD y en RAW. Sólo lectura.
- JiffyDOS opcional tanto para la 1541 como para el Commodore 64. Esto acelera muy notablemente las cargas.
- Chip de sonido SID
- Soporte de joystick, tanto real (norma Atari, SMS, megadrive...) como emulado en pad numérico
- Switch VGA 50Hz / RGB-Compuesto
- Scanlines en modo VGA
- Carga de cinta usando la entrada de audio
- Cambio de polaridad de la señal de entrada de audio
- Kernel modificado para evitar la espera de pulsación tras la cabecera en las cargas de cassette.
- LED testigo de lectura de la disquetera 1541.

## Formato de Tarjeta SD

La tarjeta SD ha de ser en un formato exclusivo, y por tanto no puede ser utilizada con otros cores. Está basada en imágenes de disco [D64](#) concatenadas, en bloques de 256K. Descargar el fichero [dummyto256.bin](#) disponible en [el repositorio oficial de ZX-Uno](#).

Para incluir varios ficheros [PRG](#) en una imagen de disco [D64](#), se puede utilizar el [programa DirMaster](#) para Windows, creando una imagen cuyo primer programa sea [FB64.PRG](#).



El formato RAW de SD es compatible con el que usa también el core de Commodore 16, así que se puede utilizar una misma tarjeta para los dos, incluyendo imágenes de disco con programas para los dos sistemas.



Este proceso borra el contenido que hubiera antes en la tarjeta SD, y no se puede deshacer.

### Windows

Concatenar las imágenes usando [COPY](#):

```
COPY /B imagen1.d64 + dummyto256.bin + imagen2.d64 + dummyto256.bin + (...)  
c64rawsd.img
```

Volcar la imagen en la SD, por ejemplo, usando [HDD Raw Copy Tool](#).

### MacOS y Linux

Concatenar las imágenes usando [cat](#):

```
cat imagen1.d64 dummyto256.bin imagen2.d64 dummyto256.bin (...) > c64rawsd.img
```

Volcar la imagen en la SD, usando [dd](#):

```
sudo umount /dev/...  
sudo dd if=c64rawsd.img of=/dev/...
```

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

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado C64 (**Caps Shift+Symbol Shift+F**, luego **U** y luego **4**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Esc** (**Caps Shift+Espacio** en ZXUnGo+): Start/Stop
- **Tab**: RUN/STOP
- **Alt** (**Caps Shift+Symbol Shift+C** en ZXUnGo+): = Commodore
- Teclado Numérico: Joystick emulado
- **F1** (**Caps Shift+Symbol Shift+1** en ZXUnGo+) a **F8** (**Caps Shift+Symbol Shift+8** en ZXUnGo+): F1 a F8
- **F9** (**Caps Shift+Symbol Shift+9** en ZXUnGo+): Tecla del símbolo de la libra
- **F10** (**Caps Shift+Symbol Shift+0** en ZXUnGo+): Tecla del símbolo +
- **Insert** (**Caps Shift+Symbol Shift+G** en ZXUnGo+): Seleccionar el primer disco de la tarjeta SD
- **Re.Pag** (**Caps Shift+Symbol Shift+J** en ZXUnGo+): Seleccionar el siguiente disco
- **Av.Pag** (**Caps Shift+Symbol Shift+N** en ZXUnGo+): Seleccionar el disco anterior
- **Ctrl+Re.Pag** (**Caps Shift+Symbol Shift+A** y luego **Caps Shift+Symbol Shift+J** en ZXUnGo+): Avanzar 10 discos
- **Ctrl+Av.Pag** (**Caps Shift+Symbol Shift+A** y luego **Caps Shift+Symbol Shift+N** en ZXUnGo+): Retroceder 10 discos
- + del teclado numérico (**Caps Shift+Symbol Shift+D** en ZXUnGo+): Cambiar la polaridad de la entrada de audio
- - del teclado numérico (**Caps Shift+Symbol Shift+R** en ZXUnGo+): Activar o desactivar scanlines en modo VGA
- **Fin** (**Caps Shift+Symbol Shift+B** en ZXUnGo+) = Cambiar paleta de colores
- **F11** (**Caps Shift+Symbol Shift+Q** en ZXUnGo+): Rotar entre puertos de joystick
- **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+): Reset modo JiffyDOS
- **Ctrl+F12** (**Caps Shift+Symbol Shift+Z** y luego **Caps Shift+Symbol Shift+W** en ZXUnGo+): Reset modo ROM C64 original (para carga de cintas)
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

## Guía básica



Desde BASIC, se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando **LOAD**. Una vez finalizada la carga, escribir **RUN** y pulsar **Enter** si fuese necesario.



Para cargar desde cinta se debe estar usando la ROM de C64 original que se activa con **Ctrl+F12** (**Caps Shift+Symbol Shift+Z**) y luego **Caps Shift+Symbol Shift+W** en ZXUnGo+).



Es posible utilizar programas como [Audiotap](#) o [Comodoro](#) con los ficheros **TAP** de cinta de Commodore.

Para cargar desde un disco, habitualmente, se puede pulsar **Mayus+Esc** (**Caps Shift+Symbol Shift+A**) y luego **Caps Shift+Espacio** en ZXUnGo+ o bien 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, usar **F1** (**Caps Shift+Symbol Shift+1** en ZXUnGo+) para listar el contenido, o bien 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 disco se debe estar usando la ROM de C64 con JiffyDOS que se activa con **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+).

## Commodore PET

El [Commodore PET \(Personal Electronic Transactor\)](#) fue una línea de ordenadores domésticos producida por Commodore a partir de 1977.

La versión para ZX-Uno ha sido [desarrollada por Jepalza](#), basada en el proyecto [pet2001fpga](#).

Características:

- Carga usando la entrada de audio
- Sólo funciona en VGA

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

El teclado está con la distribución en inglés.

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado CPET ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [K](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [F1](#) ([Caps Shift+Symbol Shift+1](#) en ZXUnGo+): Cancelar carga desde cinta
- [Alt](#) ([Caps Shift+Symbol Shift+C](#) en ZXUnGo+): Permite escribir en modo gráfico

## Guía básica



Se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando **LOAD**. Una vez finalizadada la carga, normalmente, se ha de escribir **RUN** y pulsar **Enter**.



Es posible utilizar programas como [Audiotap](#) o [Comodoro](#) con los ficheros **TAP** de cinta de Commodore.

## Commodore VIC-20

El **Commodore VIC-20** (Commodore VC20 en Alemania, VIC-1001 en Japón) fue un ordenador doméstico de 8 bits fabricado y diseñado por Commodore Business Machines, lanzado en junio de 1980, cerca de 3 años después del Commodore PET.

La versión para ZX-Uno ha sido desarrollada por McLeod y Quest

Características principales:

- Expansión de 32K de RAM. Para mayor compatibilidad se puede desactivar.
- Carga usando la entrada de audio
- Soporte para video compuesto y VGA
- Soporte de joystick

### Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

### Teclado



En algunas versiones de este core, el teclado no se inicializa correctamente si sólo hay conectado un teclado de membrana. Es necesario iniciar el ZXUnGo+ (en frío) con un teclado PS/2 conectado, a la vez, para poder utilizarlo.

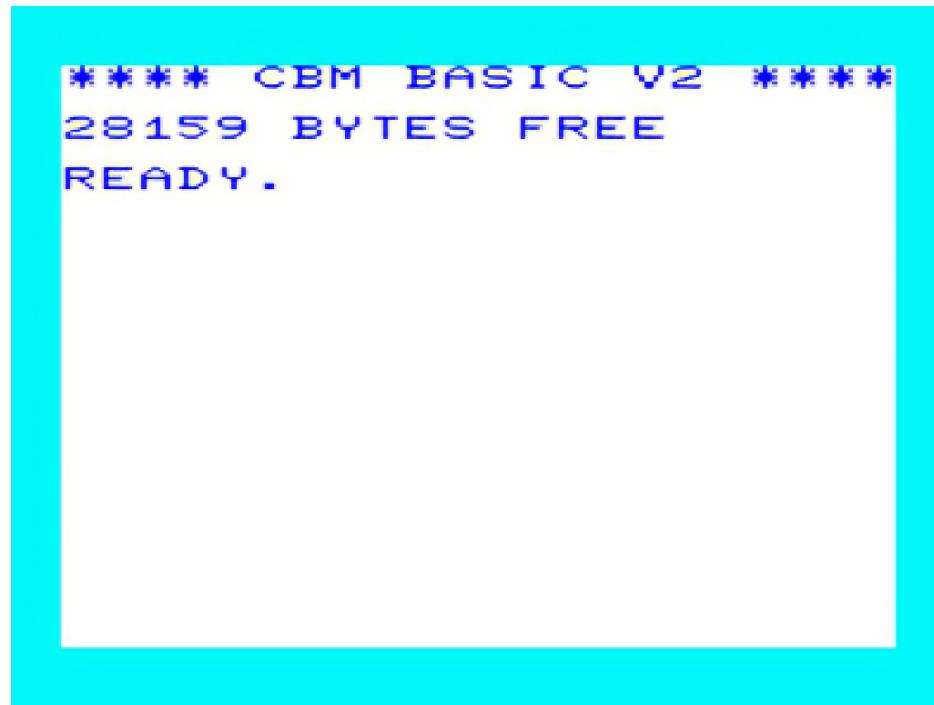
### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **VIC** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **9**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **F9** (**Caps Shift+Symbol Shift+9** en ZXUnGo+) a **F11** (**Caps Shift+Symbol Shift+Q** en ZXUnGo+): Cargar cartuchos incorporados al core (si los hubiera)
- - del teclado numérico (**Caps Shift+Symbol Shift+R** en ZXUnGo+): Cambiar la polaridad de la entrada de audio. El LED rojo se enciende cuando la polaridad está invertida
- **Bloq. Num** (**Caps Shift+Symbol Shift+I** en ZXUnGo+): Reset desactivando la ampliación de memoria de 32K (activa por defecto)
- **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo) : Reset activando la ampliación de memoria
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

## Guía básica



Se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando **LOAD**. Una vez finalizadada la carga, normalmente, se ha de escribir **RUN** y pulsar **Enter**.



Es posible utilizar programas como [Audiotap](#) o [Comodoro](#) con los ficheros **TAP** de cinta de Commodore.

## Galaksija

[Galaksija](#) (Galaxia en español) es un ordenador creado en 1983 por el inventor serbio Voja Antonić. Los esquemas fueron publicados en la primera revista de ordenadores yugoslava, "Računari u vašoj kući" (Ordenadores en Casa), en su primer número de Enero de 1984. De este primer número se vendieron 100.000 ejemplares, y posteriormente se supo que al menos 8.000 personas habían construido su propio ordenador Galaksija.

El core de ZX-Uno es una versión del [proyecto µGalaksija](#), mejorada por azesmog, añadiendo el modo Galaksija Plus, y la posibilidad de cargar programas desde una tarjeta SD.

Características del core de ZX-Uno:

- Sólo salida VGA
- Sonido opcional
- Carga de programas desde la tarjeta SD
- Modos Galaksija Clásico (minus) y Galaksija Plus

## Formato de Tarjeta SD

Se puede utilizar una tarjeta SD, con la primera partición en formato FAT16, para almacenar los programas. Estos se pueden conseguir, por ejemplo, en la web de [retrospec.sgn.net](#).

## Teclado

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **GALAKSIJ** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **G**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

### Teclas especiales y botones

Durante la ejecución del core:

- **F1** (**Caps Shift+Symbol Shift+1** en ZXUnGo+): Velocidad de ejecución normal
- **F2** (**Caps Shift+Symbol Shift+2** en ZXUnGo+): Velocidad de ejecución 2x
- **F3** (**Caps Shift+Symbol Shift+3** en ZXUnGo+): Velocidad de ejecución 4x
- **F9** (**Caps Shift+Symbol Shift+9** en ZXUnGo+): Habilitar o deshabilitar sonido
- **Esc** (**Caps Shift+Espacio** en ZXUnGo+): Congela la ejecución de un programa BASIC o Código Máquina. Volviendo a pulsar, la ejecución continua
- **Supr** (**Caps Shift+Symbol Shift+V** en ZXUnGo+): **CLR**.: Detiene la ejecución de un programa BASIC mientras se mantenga pulsada
- **Fin** (**Caps Shift+Symbol Shift+B** en ZXUnGo+): Si se pulsa durante la ejecución de un programa BASIC realiza un BREAK
- **Mayús.+M**: Habilita el modo plus más avanzado en aquellos programas que estén preparados para usarlo (por ejemplo **TETRIS.GTP**)
- **M**: Vuelve al modo normal desde el modo plus
- **Ctrl+Alt+F5**: NMI
- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+F** y luego **N** en ZXUnGo+): Soft Reset
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

## Guía básica



Desde BASIC, se puede habilitar la carga de programas desde la tarjeta SD, ejecutando la siguiente instrucción:

```
A=USR(&F000)
```

Ahora, con instrucciones como **CD <directorio>** y **DIR** se puede llegar hasta el directorio donde se encuentre el fichero a cargar. Para cargar un fichero se tiene que escribir sólo el nombre, sin la extensión (por ejemplo, para un fichero **JUEGO.GTP** se debe escribir **JUEGO**). Una vez cargado en memoria, el programa se ejecuta con la instrucción **RUN**.

El core se inicializa en modo Galaksija Clásico (minus), pero es posible pasar al modo Galaksija Plus con la instrucción:

```
A=USR(&E000)
```

Se puede consultar [en Wikipedia](#) los comandos de BASIC disponibles.

# HT-1080Z

El ordenador húngaro HT-1080Z School Computer, era básicamente un clon del ordenador Video Genie, el cual a su vez era un clon del TRS-80 Model 1 con **Level II BASIC**, al que se le añadió un chip de sonido AY-3-8910 y circuitería adicional de entrada / salida.

El core está basado en el [trabajo original de Jozsef Laszlo](#) para la MiST, el cual azesmbog portó a la placa de desarrollo AEON, y ahora a ZX-Uno.

Sus características principales son:

- Nueva versión de ROM de BASIC de 14K, en lugar de la ROM de 12K de Level II BASIC original, que permite usar letras minúsculas con el Radio Shack Lower Case Kit
- Añade el puerto FF, incorporando un beeper
- Añade el SDOS v8.4 de PVV, para cargar ficheros **CAS** desde la tarjeta SD
- Salida de vídeo VGA y Video Compuesto

## Formato de Tarjeta SD

Se puede utilizar una tarjeta SD, con la primera partición en formato FAT16, para almacenar ficheros en formato **.CAS**, pero con las siguientes limitaciones:

- Sólo se soportan **CAS** de programas en código máquina, no están soportados los ficheros de BASIC
- Si se quieren usar programas en formato **CMD** deben ser primero ser convertidos a **CAS** usando la alguma utilidad como **trld**, **trs80-tool** o **cmd2cas**
- SDOS sólo carga **CAS** en formato binario (el formato beysikovsky no está soportado)

Se pueden obtener diferentes utilidades de conversión de ficheros de cinta en estos sitios:

<https://www.trs-80.com/wordpress/conversion-tape-utilities/>

<http://48k.ca/trld.html>

<https://github.com/pullmoll/cass80>

## Teclado

### Teclas especiales y botones

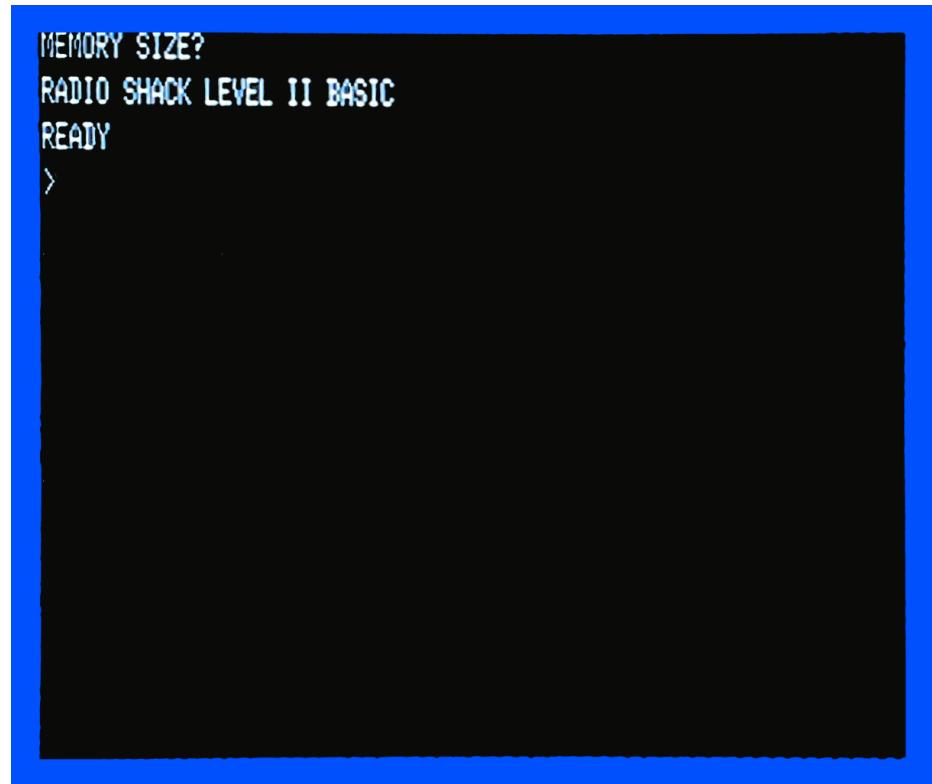
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **TRS80** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **F**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **F5** (**Caps Shift+Symbol Shift+5** en ZXUnGo+): Cambia color de la tinta
- **F6** (**Caps Shift+Symbol Shift+6** en ZXUnGo+): Cambia el color del papel
- **F7** (**Caps Shift+Symbol Shift+7** en ZXUnGo+): Cambia color del borde
- **F9** (**Caps Shift+Symbol Shift+9** en ZXUnGo+): Cambia entre modo 64/32 columnas (solo salida VGA)
- **F10** (**Caps Shift+Symbol Shift+0** en ZXUnGo+): Tecla **PAGE**
- **Inicio** (**Caps Shift+Symbol Shift+H** en ZXUnGo+): Tecla **CLEAR**
- **Mayús+0** (**Caps Shift+Symbol Shift+A** y luego **0** en ZXUnGo+): Si se ha cargado el fichero **ULCBAS.CAS** (Upper/Lower Case Driver) permite cambiar el estado de solo mayúsculas o mayúsculas+minúsculas.
- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+F** y luego **N** en ZXUnGo+): Reset
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

## Guía básica

Después de arrancar, se pregunta por el límite superior de la memoria accesible, pulsar **Enter** para continuar.



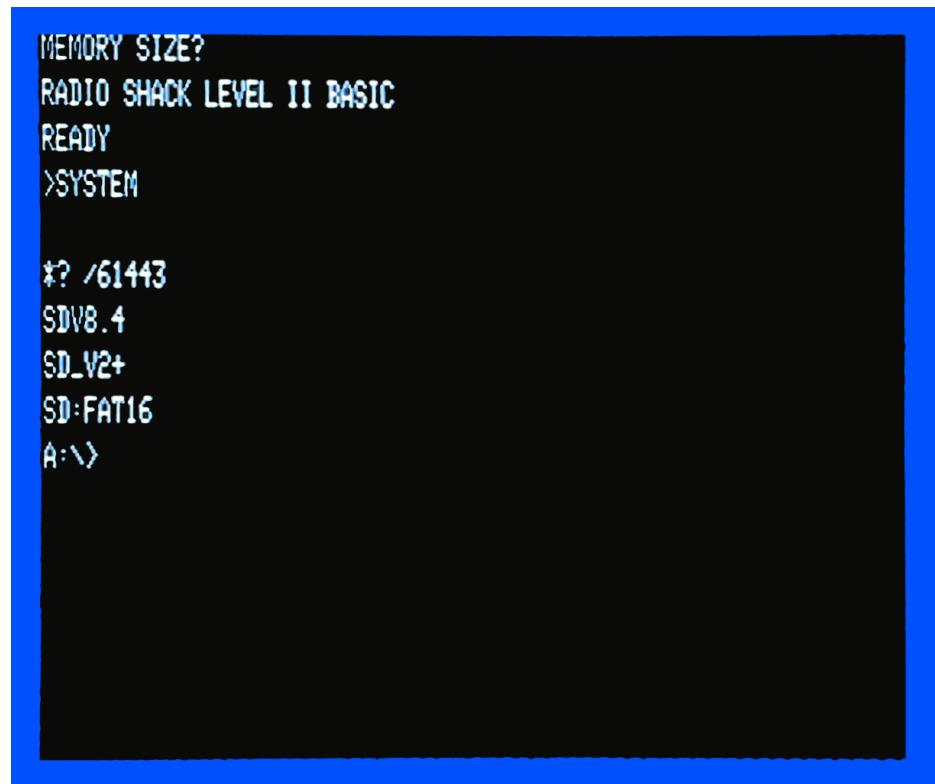
Una vez que ha cargado el intérprete de BASIC, si, por ejemplo, se escribe **PRINT MEM**, se verá la cantidad de memoria que disponible para BASIC.

Para cargar ficheros de cinta desde la tarjeta SD, se debe acceder a SDOS escribiendo

```
>SYSTEM  
*? /61440
```

o bien, si no se quiere que se limpie la pantalla

```
>SYSTEM  
*? /61443
```



```

MEMORY SIZE?
RADIO SHACK LEVEL II BASIC
READY
>SYSTEM

*? /61443
SDV8.4
SD_V2+
SD:FAT16
A:\>

```

Ahora, con instrucciones como **CD**, **DIR** se puede llegar hasta el directorio donde se encuentre el fichero **CAS** a cargar. Para cargar un fichero de cinta **.CAS** se tiene que escribir sólo el nombre, sin la extensión (por ejemplo, para un fichero **JUEGO.CAS** se debe escribir **JUEGO**).



Todos los comandos de SDOS se deben escribir en letras mayúsculas.



Pulsando **Espacio** se pausa la presentación de ficheros y directorios, y pulsando de nuevo cualquier tecla, se reanuda.



La ROM incorpora Radio Shack Lower Case Kit, el cual el core ya implementa. No obstante, para poder usar las letras minúsculas desde BASIC será necesario cargar antes desde la SD el Upper/Lower Case Driver (fichero **ULCBAS.CAS**).

# Jupiter ACE

El [Jupiter ACE](#) fue un microordenador británico lanzado al mercado en 1982 por la empresa Jupiter Cantab.

El core para ZX-Uno ha sido desarrollado por McLeod y mejorado por Azesmbog, y tiene, entre otras, las siguientes características:

- Soporte para carga desde dispositivos de audio externo
- Soporte de acceso a la tarjeta SD (implementando [SDOS](#))

## Formato de Tarjeta SD

Se puede utilizar una tarjeta SD, con la primera partición en formato FAT16, para almacenar ficheros en formato [.TAP](#) de Jupiter ACE. Estos se pueden conseguir, por ejemplo, en la web de [Jupiter ACE Archive](#).

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [ACE](#) ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [C](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [F1](#) ([Caps Shift+Symbol Shift+1](#) en ZXUnGo+) a [F4](#) ([Caps Shift+Symbol Shift+4](#) en ZXUnGo+): Modos Turbo
- [Ctrl+Alt+F5](#) ([Caps Shift+Symbol Shift+Y](#) en ZXUnGo+): NMI
- [Ctrl+Alt+Supr](#) ([Caps Shift+Symbol Shift+F](#) y luego [N](#) en ZXUnGo+): Reset
- [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)

## Guía básica

Desde Forth, se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando **LOAD <nombre>**.



Los ficheros **TAP** de Jupiter ACE son diferentes de los de ZX Spectrum, y, para poder utilizarse con algunos reproductores, han de ser convertidos previamente a **TZX** con la utilidad **acetap2zx** disponible en el [repositorio de ZX-Uno](#) (Usuario **guest**, contraseña **zxuno**).

Para cargar ficheros de cinta desde la tarjeta SD, se debe acceder a SDOS escribiendo

```
61440 call
```

Ahora, con instrucciones como **CD**, **DIR** se puede llegar hasta el directorio donde se encuentre el fichero **TAP** a cargar. Para cargar un fichero de cinta **.TAP** se tiene que escribir sólo el nombre, sin la extensión (por ejemplo, para un fichero **JUEGO.TAP** se debe escribir **JUEGO**).



Todos los comandos de SDOS se deben escribir en letras mayúsculas.

Una vez la cinta ha cargado, escribir el comando necesario para iniciar el programa.



En lugar de BASIC como otros sistemas, el Jupiter ACE utiliza **Forth**. Para ver los comandos disponibles (incluyendo los que inician un programa cargado desde cinta), se tiene que utilizar **vlist**.

```
vlist
FORTH UFLOAT INT FNEGATE F/ F* F
+ F- LOAD SVERIFY VERIFY BLOAD B
SAVE SAVE LIST EDIT FORGET REDEF
INE EXIT ." ( [ +LOOP LOOP DO UN
TIL REPEAT BEGIN THEN ELSE WHILE
IF ] LEAVE J I ) DEFINITIONS V
OCABULARY IMMEDIATE RUNS DOES
COMPILER CALL DEFINER ASCII LITE
RBL CONSTANT VARIABLE ALLOT C'
CREATE : DECIMAL MIN MAX XOR AN
D OR 2- 1- 2+ 1+ D+ - + DNEGATE
NEGATE U/MOD */ * MOD / */MOD /M
OD U* D< UK < > = @> @< @= ABS O
UT IN INKEY BEEP PLOT AT F. EMIT
CR SPACES SPACE HOLD CLS ##$ U
. SIGN #> <# TYPE ROLL PICK OU
ER ROT ?DUP R> >R ! @ O! O@ SWAP
DROP DUP SLOW FAST INVIS VIS CO
NUERT NUMBER EXECUTE FIND VLIST
WORD RETYPE QUERY LINE ; PAD BAR
E CURRENT CONTEXT HERE ABORT QUI
T OK
■
```

# MSX

MSX1FPGA es un proyecto para clonar MSX1 en una FPGA. 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

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD 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.

Se ha de 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](#)

Una vez descargado, copiar el contenido del [directorio SD](#) en la raíz de la primera partición de la tarjeta SD.

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 ZUNO+.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado MSX ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [3](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [F1 \(Caps Shift+Symbol Shift+1 en ZXUnGo+\)](#) a [F5 \(Caps Shift+Symbol Shift+5 en ZXUnGo+\)](#): [F1](#) a [F5](#)
- [Impr Pant \(Caps Shift+Symbol Shift+T en ZXUnGo+\)](#): Cambia el modo entre VGA y RGB
- [Bloq Desp \(Caps Shift+Symbol Shift+Y en ZXUnGo+\)](#): Cambia el modo de línea de exploración (Scanlines)
- [Pausa \(Caps Shift+Symbol Shift+U en ZXUnGo+\)](#): Cambia entre 50Hz y 60Hz
- [F11 \(Caps Shift+Symbol Shift+Q en ZXUnGo+\)](#): Activa o desactiva el modo turbo
- [Ctrl+Alt+Supr \(Caps Shift+Symbol Shift+F y luego N en ZXUnGo+\)](#): Soft Reset
- [Ctrl+Alt+Backspace \(Caps Shift+Symbol Shift+F y luego B en ZXUnGo+\)](#): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)
- [ALT Izquierdo \(Caps Shift+Symbol Shift+C en ZXUnGo+\)](#): MSX GRAPH
- [ALT Derecho \(Caps Shift+Symbol Shift+M en ZXUnGo+\)](#): MSX CODE
- [Re Pág \(Caps Shift+Symbol Shift+J en ZXUnGo+\)](#): MSX SELECT
- [Inicio \(Caps Shift+Symbol Shift+H en ZXUnGo+\)](#): MSX HOME ([Mayús+HOME](#) o [Caps Shift+Symbol Shift+A](#) y luego [Caps Shift+Symbol Shift+H en ZXUnGo+\)](#): CLS)
- [Fin \(Caps Shift+Symbol Shift+B en ZXUnGo+\)](#): MSX STOP
- [Ñ o Windows \(Caps Shift+Symbol Shift+X en ZXUnGo+\)](#): MSX DEAD



En BASIC, se puede usar [CTRL + STOP \(Ctrl+Fin o Caps Shift+Symbol Shift+Z](#) y luego [Caps Shift+Symbol Shift+B en ZXUnGo+\)](#) 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
```

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.

# NES

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

La versión para ZX-Uno ha sido creada para ZX-Uno por [DistWave y Quest](#), basado en el [core de NES de Ludde/Strigeus para la placa Nexys4](#).

Algunas de las características del core son:

- Sólo soporta salida VGA y utiliza timings poco rigurosos, por lo que es posible que de problemas en algunos monitores
- 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
- El core básico utiliza los 512 KB de SRAM del ZX-Uno. Se ha dividido la memoria en 2 bancos de 256 KB, uno para PRG\_ROM y otro para CHR\_ROM. Por tanto, las ROMs que utilizan más de 256 KB para CHR o PRG no funcionan. Evidentemente cualquier ROM mayor de 512 KB tampoco.
- Existen otras versiones del core adaptadas para 2 MB de SRAM y que permiten la carga de juegos que usen hasta 1MB de CHR RAM y 1MB de PRG RAM (2MB en total).
- Compatible con 1 o 2 joysticks

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, 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-Uno.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo **ZX** normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Esc** (**Caps Shift+Espacio** en ZXUnGo+) o botón 2 del joystick: Mostrar u ocultar el menú de configuración
- Teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+), y **Enter** para usar el menú
- **W, A, S, D** (personalizables) o el joystick 1: Controles de dirección del jugador 1
- **F** y **R** (personalizables) o botones de disparo del joystick 1: Botones A y B (Jugador 1)
- **5** y **1**: **Select** y **Start** (Jugador 1)
- **I, J, K, L** (personalizables) o el joystick 2: Controles de dirección del jugador 2
- **H, Y** (personalizables) o botones de disparo del joystick 1: Botones A y B (Jugador 2)
- **6** y **2**: **Select** y **Start** (Jugador 1)
- **F4** (**Caps Shift+Symbol Shift+4** en ZXUnGo+): Intercambiar la asignación de teclas de jugador 1 y jugador 2
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

### Personalización de los controles

Los controles por defecto del core con soporte de teclado se pueden modificar. Para ello, se ha de crear en la raíz de la tarjeta SD un fichero llamado **KEYSP1`** para el primer jugador y **KEYSP2** para el segundo (se puede descargar uno de ejemplo desde [el foro de ZX-Uno](#)).

Estos ficheros se pueden crear con un editor hexadecimal (como [HxD para Windows](#)) y están formados por una secuencia de bytes indicando en este orden: **Arriba**, **Abajo**, **Izquierda**, **Derecha**, **Botón A**, **Botón B**. Los códigos a utilizar en la secuencia se pueden consultar en la tabla [Scan Codes](#) al final del manual (Columna **MAKE**), pero, teniendo en cuenta que, si el código está emparejado con **0xE0**, se ha de usar el otro valor, sumáñole **0x80** (por ejemplo, para **E0 14**, sería **0x94**).



Por ejemplo, la secuencia **15 1C 44 4D 29 22** se correspondería con **Q A O P Espacio X**, y la secuencia **1D 1B 23 2B 1A 22** se correspondería con **W S D F Z X**

## Guía básica

Pulsando **Esc** (**Caps Shift+Espacio** en ZXUnGo+) 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+6** y **Caps Shift+7** en ZXUnGo+) 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 SD (Load ROM)
- Salir del menú (Exit)



Se puede avanzar y retroceder más rápido por las opciones de menú usando **Re Pág** (**Caps Shift+Symbol Shift+J** en ZXUnGo+) y **Av Pág** (**Caps Shift+Symbol Shift+N** en ZXUnGo+).

## Oric Atmos

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

La versión para ZX-Uno tiene las siguientes características:

- Salida de vídeo VGA o RGB
- NMI
- DOS-8D (basado en el [Pravetz 8d](#), clon búlgaro del Oric Atmos)

Para más información consultar [el foro oficial de ZX-Uno](#).

## Formato de Tarjeta SD

La tarjeta SD ha de ser en un formato exclusivo, y por tanto no puede ser utilizada con otros cores. Está basada en imágenes de disco [NIB](#) concatenadas. Para convertir imágenes de disco en otro formato ([DSK](#)), se puede utilizar la utilidad [dsk2nib](#) (notar que tienen que ser imágenes de 232960 bytes de tamaño).



Este proceso borra el contenido que hubiera antes en la tarjeta SD, y no se puede deshacer.

### Windows

Concatenar las imágenes usando [COPY](#):

```
COPY /B imagen1.nib + imagen2.nib + (...) + imagen20.nib oric_discos.img
```

Volcar la imagen en la SD, por ejemplo, usando [HDD Raw Copy Tool](#).

### MacOS y Linux

Concatenar las imágenes usando [cat](#):

```
cat imagen1.nib imagen2.nib (...) imagen20.nib > oric_discos.img
```

Volcar la imagen en la SD, usando [dd](#):

```
sudo umount /dev/...
sudo dd if=oric_discos.img of=/dev/...
```

## Teclado

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



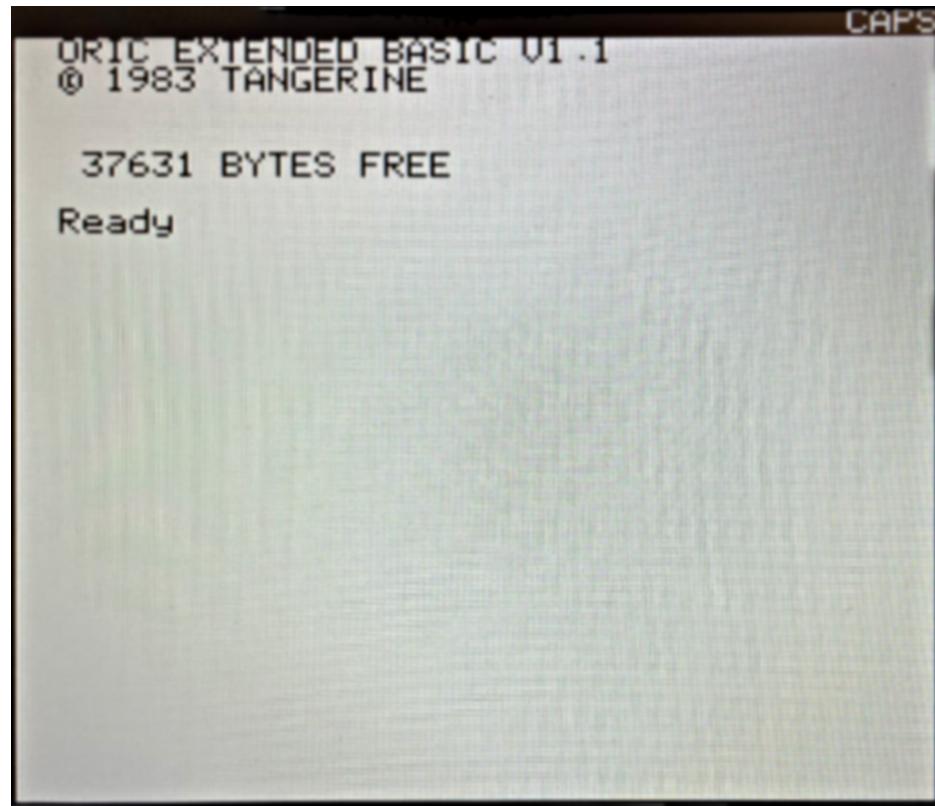
## Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **ORIC** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **A**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+): para cambiar entre modo de vídeo RGB y VGA
- **Inicio** (**Caps Shift+Symbol Shift+H** en ZXUnGo+): Mostrar datos de depuración (Pista de disco actual -dos dígitos-, número de imagen de disco -dos dígitos-, y registro de contador de programa -cuatro dígitos-)
- **Fin** (**Caps Shift+Symbol Shift+B** en ZXUnGo+): NMI
- **Inicio+Fin**: Reset
- **Re Pág** (**Caps Shift+Symbol Shift+J** en ZXUnGo+): Seleccionar el siguiente disco
- **Av Pág** (**Caps Shift+Symbol Shift+N** en ZXUnGo+): Seleccionar el disco anterior

## Guía básica



Se puede cargar DOS-8D con el comando:

```
CALL800
```

Ahora, se puede ver el contenido del disco actual con el comando **DIR**.

Para ejecutar un programa, se ha de escribir **-** delante del nombre. Así, por ejemplo, para un programa llamado **DEFENDER.EXE**:

```
-DEFENDER.EXE
```

## Oric Atmos (Kyp)

Otra versión distinta de core de [Oric Atmos](#), basada en la versión de Rampa para MiST, Mistica y SiDi.

Esta versión ha sido [desarrollada por Kyp069](#) 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 1.22, ROM alternativa que corrige errores
- Las direcciones del joystick están mapeadas a las teclas del cursor ([Caps Shift+5](#), [Caps Shift+6](#), [Caps Shift+7](#) y [Caps Shift+8](#) en ZXUnGo+), 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 SD

Este core no utiliza la tarjeta SD.

## 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**).



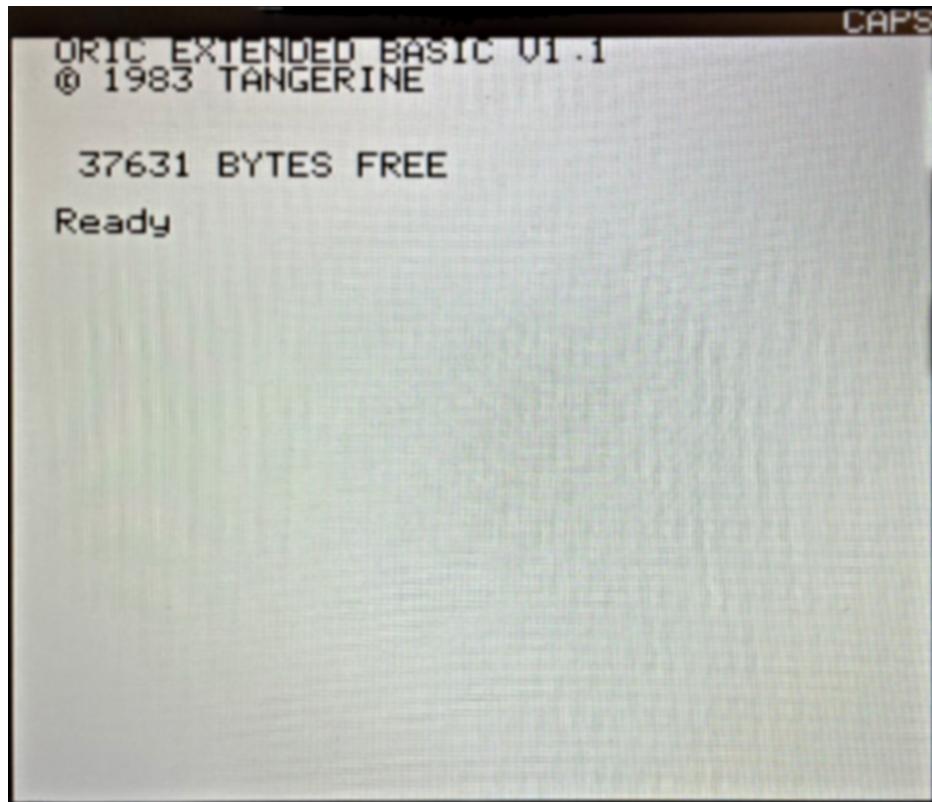
### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado ORIC (Caps Shift+Symbol Shift+F, luego U y luego A). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **F1 (Caps Shift+Symbol Shift+1 en ZXUnGo+):** Seleccionar la ROM 1.1
- **F2 (Caps Shift+Symbol Shift+2 en ZXUnGo+):** Seleccionar la ROM 1.22
- **F5 (Caps Shift+Symbol Shift+5 en ZXUnGo+):** NMI
- **F10 (Caps Shift+Symbol Shift+0 en ZXUnGo+):** Habilita la salida de audio al ejecutar **CSAVE**
- **Ctrl+Alt+Backspace (Caps Shift+Symbol Shift+F y luego B en ZXUnGo+) o F11 (Caps Shift+Symbol Shift+Q en ZXUnGo+):** Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**
- **Ctrl+Alt+Supr (Caps Shift+Symbol Shift+F y luego N en ZXUnGo+) o F12 (Caps Shift+Symbol Shift+W en ZXUnGo+):** Soft reset
- **Bloq. Despl. (Caps Shift+Symbol Shift+Y en ZXUnGo+):** para cambiar entre modo de vídeo compuesto y VGA
- **Alt Gr. (Caps Shift+Symbol Shift+M en ZXUnGo+):** Tecla **FUNC**
- **Esc (Caps Shift+Espacio en ZXUnGo+):** Tecla **ESC**
- **Supr (Caps Shift+Symbol Shift+V en ZXUnGo+):** Tecla **DEL**
- **Ctrl (Caps Shift+Symbol Shift+Z en ZXUnGo+):** Tecla **CTRL**
- **Mayus. (Caps Shift en ZXUnGo+):** 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, se puede utilizar 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 ZXUnGo+).



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

# Pong

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

La última versión para ZX-Uno ha sido realizada por avlixia, y algunas de sus características son:

- Dos modos de vídeo seleccionables: Vídeo compuesto/RGB y VGA
- 7 tipos de juego
- Soporte 2 o 4 jugadores
- Compatible con joystick(s), teclado, ratón y codificadores (encoders) rotatorios
- Varios modos de color

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

### Teclas especiales y botones

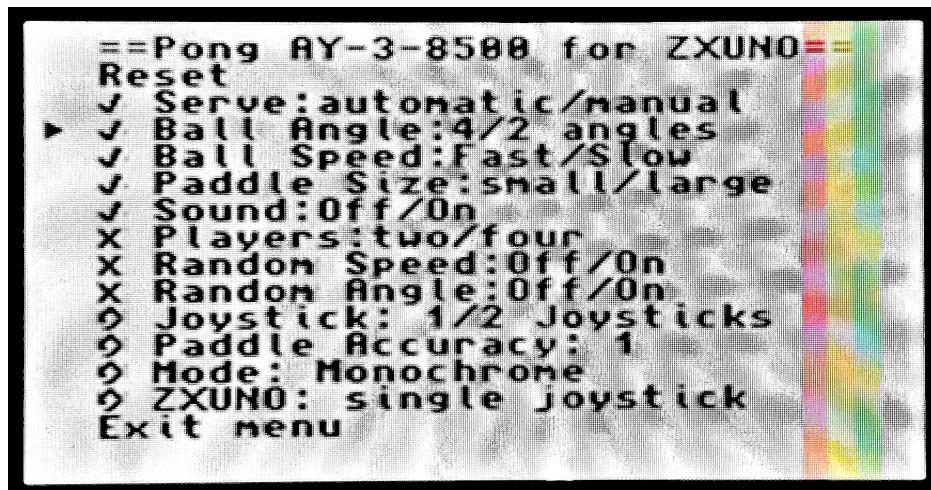
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo **ZX** normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- **Esc** (**Caps Shift+Espacio** en ZXUnGo+) o botón 2 del joystick: Mostrar u ocultar el menú de configuración
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+) o **F11** (**Caps Shift+Symbol Shift+Q** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**
- **F3** (**Caps Shift+Symbol Shift+3** en ZXUnGo+) o **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+): 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** (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+) o **O** 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+6** y **Caps Shift+7** en ZXUnGo+) y **Enter** para navegar por el menú

## Guía básica

Pulsando **Esc** (**Caps Shift+Espacio** en ZXUnGo+) o el botón 2 del joystick se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+) y **Enter** para elegir y seleccionar opciones del menú.



En él se pueden activar o desactivar las siguientes opciones:

- Reset del core
- 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)
- Número de jugadores (dos -two- o cuatro -four-)
- Velocidad Aleatoria (Random Speed)
- Ángulo Aleatorio (Random Angle)
- Control por Joystick, teclado o ratón (Joystick)
- Precisión del movimiento (Paddle Accuracy)
- Modo de color (Mode)
- Modo de joystick para ZX-Uno (único, con splitter o con VGA2M)
- Salir del menú (Exit menu)



En el caso de utilizar un ratón como método de control, es muy recomendable conectarlo al ZX-Uno antes de arrancar el core

## PC XT

El core [Next186lite](#) es una implementación adaptada y recortada del core [Next186](#) para FleaFPGA y Papilio Pro.

Características del sistema implementado:

- Sólo funciona en VGA
- Núcleo Next186 a 25 MHz y bus del sistema a 50 MHz. El procesador es equivalente a un 286 en modo real. No hay modo protegido
- 504 KB de memoria RAM convencional
- 60 KB de memoria VRAM
- Soporte para teclado y ratón PS/2
- Modo de texto 80x25
- Modos gráficos EGA 320x200x16 y MCGA 320x200x256, parcialmente funcionales (en el modo MCGA no se dibujan las últimas líneas de la pantalla porque faltan 2,5 KB de memoria de vídeo). No se emula una tarjeta gráfica real
- Sonido Beeper y Tandy 3 voice
- DAC de puerto paralelo para reproducir sonido digital

### Formato de Tarjeta SD

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 SD como disco. En los foros de [ZX-Uno](#) y [ZXDOS+](#) se puede obtener más información y encontrar algunas imágenes de tarjeta SD.

Una vez particionada y formateada (con la reserva al final, ver más adelante), se debe grabar la imagen de BIOS [Next186\\_BIOS\\_zxdos\\_ddr3.COM](#) (disponible [en este enlace](#)) en los últimos 16 sectores de la tarjeta SD.



**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 SD
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 SD
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
```

## 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
```

Si no estuviera ya, [instalar el core de PC XT](#) en el ZX-Uno.

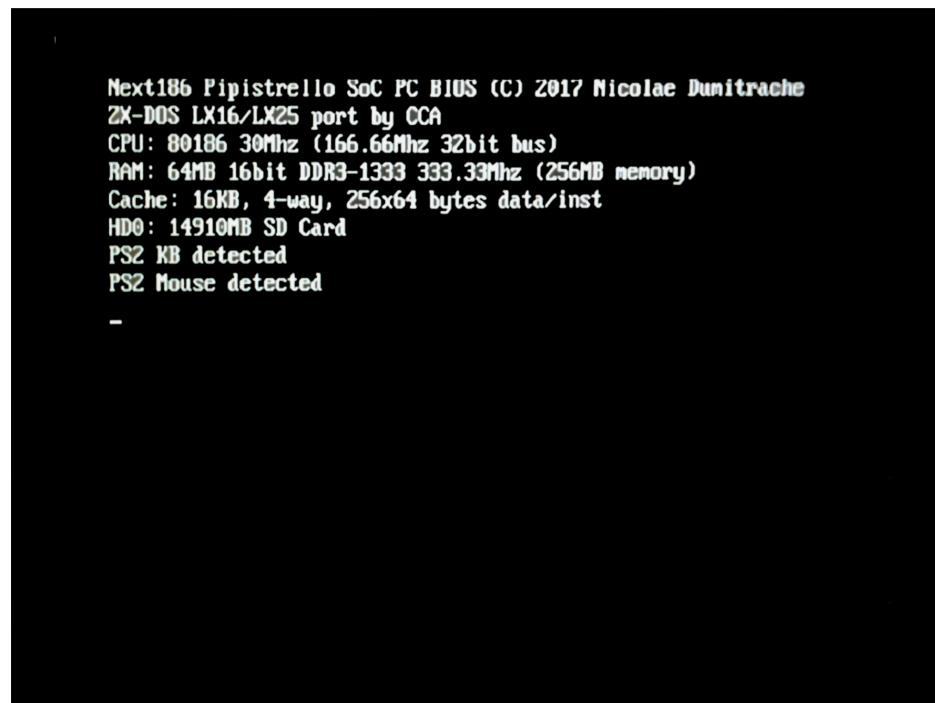
## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ se corresponden con el modo de teclado **PCXT** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **0**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

- **Ctrl+Alt+Supr** (**Caps Shift+Symbol Shift+F** y luego **N** en ZXUnGo+): Soft Reset
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**

## Guía Básica



1. Iniciar el ZX-Uno
2. En el caso de ZXUnGo+, cambiar el modo de teclado a **PC XT** (**Caps Shift+Symbol Shift+F**, luego **U** y luego **0**)
3. Reiniciar el ZXUnGo+ sin que se pierda la configuración del teclado pulsando **Caps Shift+Symbol Shift+F** y luego **B**
4. Rápidamente, pulsar **Caps Shift+2** o **Caps Shift+6** 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 español (**KEYB ES,**)



No se recomienda usar el modo de teclado **PC XT** en ZXUnGo+, a la vez que un teclado de PS/2 conectado, ya que se generan conflictos y no funciona correctamente.

# SAM Coupé

El [SAM Coupé](#) fue un ordenador doméstico desarrollado y comercializado a partir de 1989 por Miles Gordon Technology, basado en el Sinclair ZX Spectrum.

El core de ZX-Uno tiene las siguientes características:

- Soporte VGA (con scanlines) y RGB / Video compuesto
- Soporte de teclado completo
- Sonido SAA1099
- Carga desde dispositivos de audio externos

## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

### Teclas especiales y botones

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [SAM](#) ([Caps Shift+Symbol Shift+F](#), luego [U](#) y luego [B](#)). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- [Ctrl+Alt+F5](#): NMI.
- [Ctrl+Alt+Supr](#)([Caps Shift+Symbol Shift+F](#) y luego [N](#) en ZXUnGo+): Soft Reset
- [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard Reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)

## Guía básica



Sólo es posible cargar programas desde una cinta (u otro dispositivo externo de audio) con el comando **LOAD ""** (o pulsando **7** del teclado numérico, equivalente a pulsar **F7** en el teclado del SAM Coupé).



En el repositorio SVN de ZX-Uno ([aquí](#) - Usuario **guest**, contraseña **zxuno**) se pueden encontrar varios programas adaptados a fichero de cinta TZX.

# Sega Master System

La [Master System](#) fue una consola de videojuegos de 8 bits basada en cartuchos y tarjetas, desarrollada por Sega.

El core de ZX-Uno está basado en el [trabajo original de Ben para Papilio Plus](#).

Algunas de sus características son:

- Soporte para VGA y Video compuesto o RGB, con sincro PAL (50Hz)
- Soporte de joysticks
- Emulación de joystick vía teclado
- Lectura de archivos desde la tarjeta SD

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, con la primera partición en formato FAT16, para almacenar los ficheros con las imágenes (extensión [.SMS](#) o [.BIN](#)) de los juegos que se desee cargar. Los ficheros pueden estar en subdirectorios.

Si no estuviera ya, [instalar el core de Sega Master System](#) en el ZX-Uno.

## Teclado

### Teclas especiales y botones

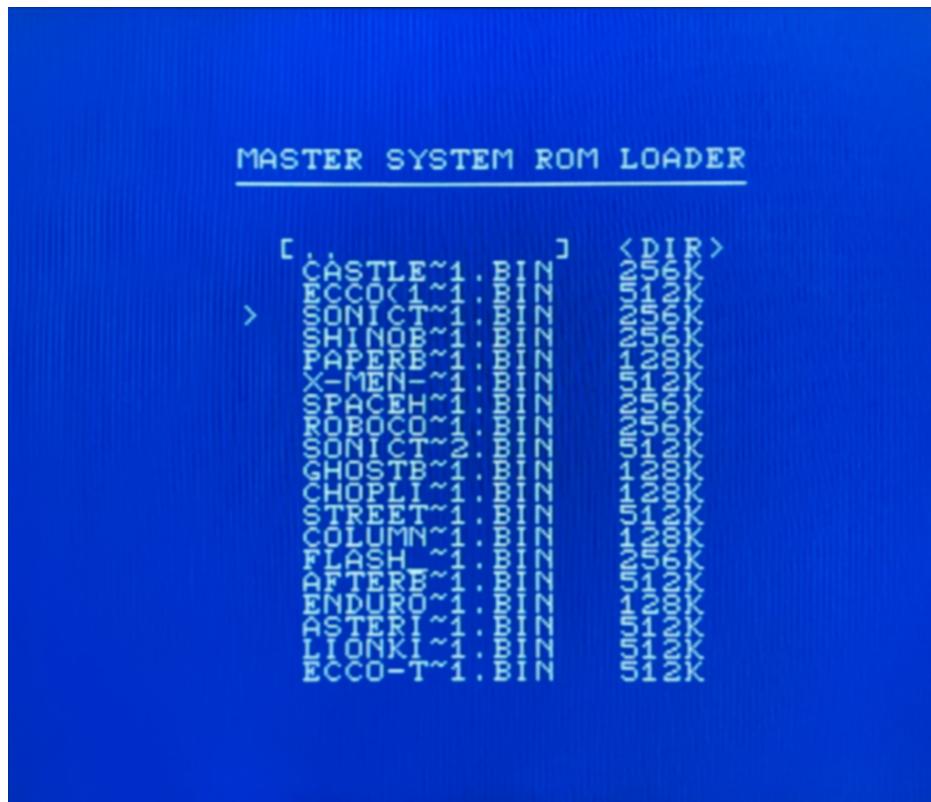
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo [ZX](#) normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- Cursor ([Caps Shift+5](#), [Caps Shift+6](#), [Caps Shift+7](#) y [Caps Shift+8](#) en ZXUnGo+) o [Q](#), [A](#), [O](#), [P](#) o el joystick 1: Controles de dirección del jugador 1
- [Z](#), [Windows](#) o el botón de joystick 1: Botón de disparo 1 del jugador 1
- [X](#), [Espacio](#) o el botón secundario de joystick 1: Botón de disparo 2 del jugador 1
- [Pausa](#) ([Caps Shift+Symbol Shift+U](#) en ZXUnGo+): Tecla de botón de pausa de Master System
- [Bloq Desp](#) ([Caps Shift+Symbol Shift+Y](#) en ZXUnGo+): para cambiar entre modo de vídeo RGB/Vídeo Compuesto y VGA
- [Ctrl+Alt+Backspace](#) ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)
- [F12](#) ([Caps Shift+Symbol Shift+W](#) en ZXUnGo+): Reset y mostrar el selector de ficheros

## Guía básica

Pulsando **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+) se muestra el selector de archivos **SMS** o **BIN**. Para desplazarse por el menú y elegir el archivo a cargar, se utilizan las teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+) y **Enter**.



## SmartROM

SmartROM es un nuevo firmware para el core de Spectrum del ZX-Uno pensado para funcionar en placas FPGA que no tengan flash, o en las que no se pueda usar. Está basado en una versión del core de McLeod hay había hecho un core "sin flash", que contiene la ROM y esxdos dentro del propio core. El proyecto SmartROM incluye una versión ligeramente modificada de ese core, junto con un fichero que en la SD que permite cargar una ROM de Spectrum distinta y, además, cambiar algunos ajustes.

### Formato de Tarjeta SD

La tarjeta se debe formatear [igual que para el uso con el core principal de Spectrum](#), es decir, con, al menos, una partición (la primera en el caso de haber varias) en formato FAT16 o FAT32.

Además, la tarjeta debe tener [instalado esxdos 0.8.8](#) y debe tener creado, en la raíz, un directorio llamado **ZXUNO** con, al menos, [el fichero SMARTROM.ZX1](#) y un fichero RomPack llamado **ROMS.ZX1** en su interior.

Además, si se está utilizando un teclado PS/2 que no esté en español, se puede copiar el [fichero de mapa de teclado](#) correspondiente, renombrado como **KEYMAP.ZX1**.

Si no estuviera ya, [instalar el core smartROM](#) en el ZX-Uno.

# Vectrex

La [Vectrex](#) era una consola con pantalla integrada y con los gráficos basados en vectores, en vez de píxeles.

El core para ZX-Uno está basado en el [trabajo original de DarFPGA](#).

Algunas de sus características son:

- Soporte VGA
- Menú OSD
- Controles vía teclado y vía joystick
- Tamaño máximo de la ROM a cargar: 8K

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, con la primera partición en formato FAT16 o FAT32, para almacenar los ficheros con las imágenes (extensión **.BIN** o **.VEC**) de los juegos que se desee cargar. Los ficheros pueden estar en subdirectorios.

En [archive.org](#) se puede obtener software para Vectrex.

Si no estuviera ya, [instalar el core de Vectrex](#) en el ZX-Uno.

## Teclado

### Teclas especiales y botones

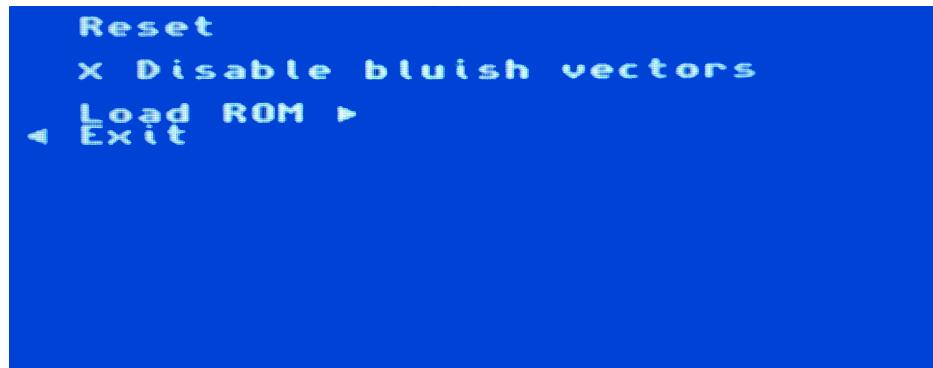
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo **ZX** normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

Durante la ejecución del core:

- Cursor (**Caps Shift+5**, **Caps Shift+6**, **Caps Shift+7** y **Caps Shift+8** en ZXUnGo+) o el joystick 1: Controles de dirección
- **Z**: Botón de disparo 1
- **X** o el botón principal de joystick: Botón de disparo 2
- **C** o el botón secundario de joystick: Botón de disparo 3
- **V**: Botón de disparo 4
- **Esc** (**Caps Shift+Espacio** en ZXUnGo+): para mostrar u ocultar el menú.
- **Ctrl+Alt+Backspace** (**Caps Shift+Symbol Shift+F** y luego **B** en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de **Enter**
- **F12** (**Caps Shift+Symbol Shift+W** en ZXUnGo+): Reset

## Guía básica

Por defecto la Vectrex arranca con el juego que trae en memoria de fábrica (Mine Storm).



Pulsando **Esc** se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+)y **Enter** para elegir y seleccionar en el menú.

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

- Reiniciar el core (Reset)
- Utilizar blanco neutro para los vectores (Disable bluish vectors)
- Cargar un fichero desde la SD (Load ROM)
- Salir del menú (Exit)

# Videopac

La [Philips Videopac](#), también conocida como Magnavox Odyssey<sup>2</sup>, Philips Videopac G7000 o Philips Odyssey<sup>2</sup> es una videoconsola lanzada en 1978.

La versión para ZX-Uno 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/Video Compuesto 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

## Formato de Tarjeta SD

Se debe de utilizar una tarjeta SD, 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-Uno.

## Teclado

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con cualquier modo de teclado excepto el modo ZX normal. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.. Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

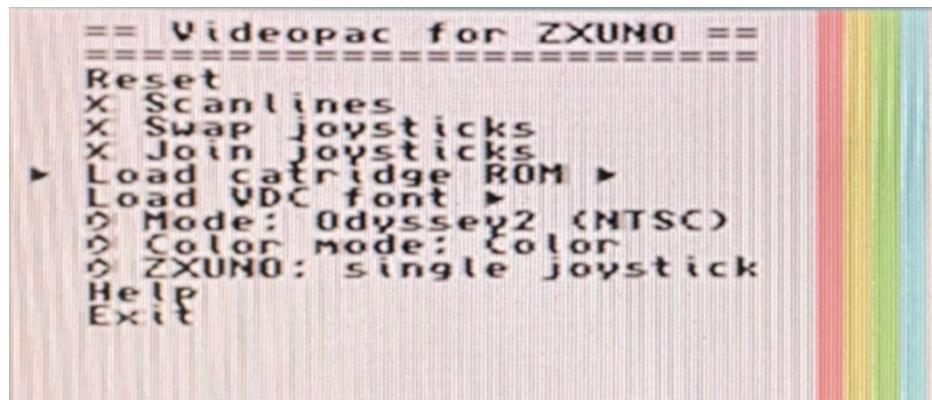
### Teclas especiales y botones

Durante la ejecución del core:

- **Bloq. Despl.** ([Caps Shift+Symbol Shift+Y](#) en ZXUnGo+): para cambiar entre modo de vídeo RGB/Vídeo Compuesto y VGA
- **F2** ([Caps Shift+Symbol Shift+2](#) en ZXUnGo+): Alternar entre RGB y vídeo compuesto cuando no se está usando el modo VGA
- **F3** ([Caps Shift+Symbol Shift+3](#) en ZXUnGo+): Reset
- **Ctrl+Alt+Backspace** ([Caps Shift+Symbol Shift+F](#) y luego [B](#) en ZXUnGo+): Hard reset. Backspace es la tecla de borrar hacia atrás, encima de [Enter](#)
- 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** ([Caps Shift+Espacio](#) en ZXUnGo+) o el botón 2 del joystick 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 ZXUnGo+) y luego [Enter](#) para elegir y seleccionar en el menú.

## Guía básica

Pulsando **Esc** (**Caps Shift+Espacio** en ZXUnGo+) o el botón 2 del joystick se muestra el menú de configuración. Se usan las teclas de cursor (**Caps Shift+6** y **Caps Shift+7** en ZXUnGo+) 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)
- 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 ZX-Uno ha sido [creada por Jepalza](#) y está basada en la [implementación anterior de un ZX97](#).

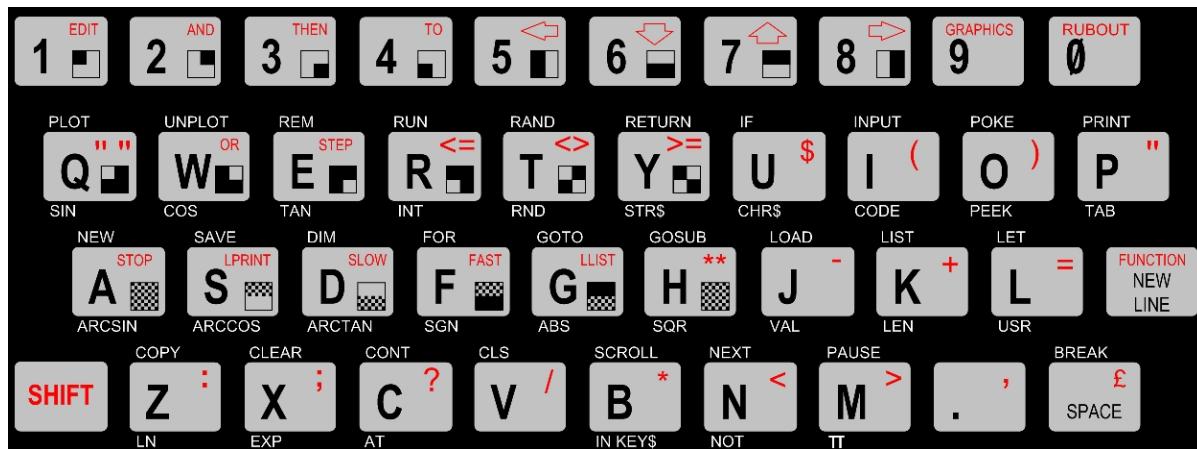
## Formato de Tarjeta SD

Este core no utiliza la tarjeta SD.

## Teclado

Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado [ZX81 \(Caps Shift+Symbol Shift+F, luego U y luego D\)](#). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

El teclado no está mapeado a la distribución de uno 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](#).



## Guía básica

Desde BASIC, se puede cargar desde una cinta (u otro dispositivo externo de audio) con el comando [LOAD"](#). Notar que, durante la carga, se desactiva la señal de vídeo y, al contrario que en la máquina original, se puede escuchar el audio de la cinta.



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.

## ZX81 y ZX80

Versión alternativa de core de ZX81 y ZX80 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 SD

Se debe de utilizar una tarjeta SD, 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 y ZX80](#) en el ZX-Uno.

## Teclado

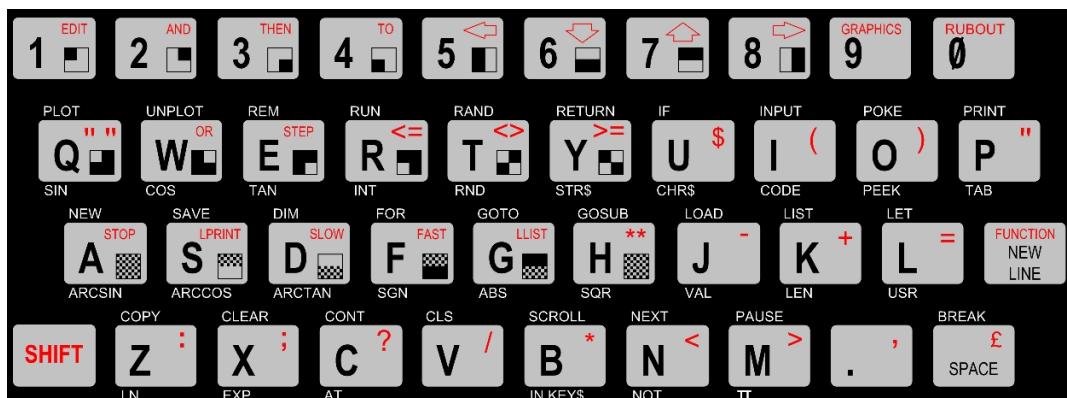
Las combinaciones específicas de ZXUnGo+ que se indican a continuación se corresponden con el modo de teclado **ZX81** (**Caps Shift+Symbol** **Shift+F**, luego **U** y luego **D**). Véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información.

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



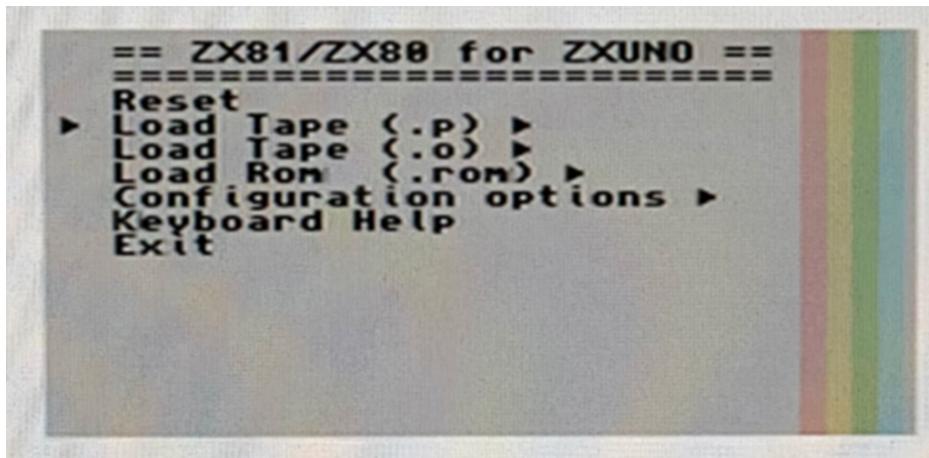
## Teclas especiales y botones

Durante la ejecución del core:

- **F1 (Caps Shift+Symbol Shift+1 en ZXUnGo+)**: Habilitar o deshabilitar los caracteres alternativos
- **F5 (Caps Shift+Symbol Shift+5 en ZXUnGo+)** o el botón 2 del joystick para mostrar u ocultar el menú.
- **F9 (Caps Shift+Symbol Shift+9 en ZXUnGo+)**: Deshabilita o habilita la salida MIC al conector de audio, porque algunos juegos generan un ruido molesto
- **F10 (Caps Shift+Symbol Shift+0 en ZXUnGo+)**: 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+Y en ZXUnGo+)**: para cambiar entre modo de vídeo RGB y VGA
- **Ctrl+Alt+Supr (Caps Shift+Symbol Shift+F y luego B en ZXUnGo+)**: Reset
- **Ctrl+Alt+Backspace (Caps Shift+Symbol Shift+F y luego B en ZXUnGo+)**: Hard reset.

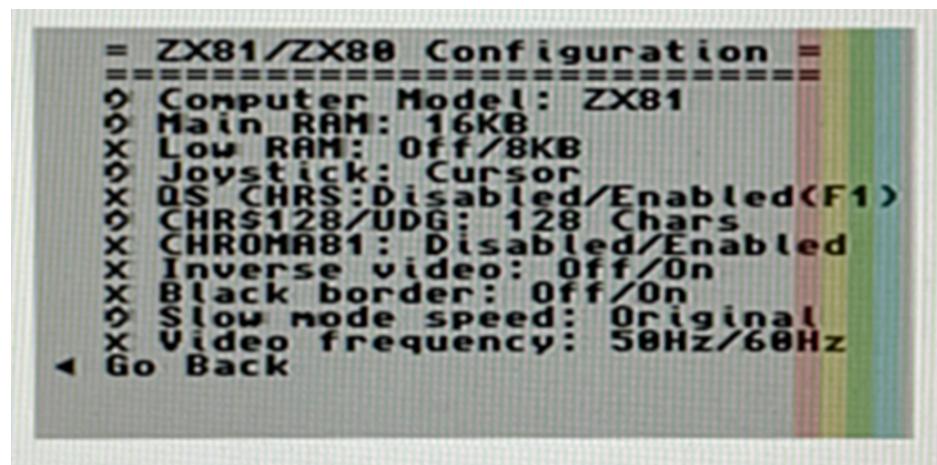
## Guía básica

Pulsando **F5 (Caps Shift+Symbol Shift+5 en ZXUnGo+)** 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 ZXUnGo+) 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

# Otro Hardware

## Joy2PS2

Es un dispositivo al que se le pueden conectar uno o dos mandos de diversas plataformas antiguas, y que convierte las órdenes de esos mandos en pulsaciones de teclado, que se transmite por un puerto PS/2. Así, se pueden usar los mandos con todo tipo de software que funcione con el teclado.



La web oficial es <http://spark2k06.strangled.net/joy2ps2/> y el repositorio de código donde se encuentra el firmware es <https://github.com/spark2k06/zxuno/tree/master/joy2ps2>

En el inicio, el mando principal siempre se encuentra asignado a las teclas de cursor, el primer botón a **Enter** y el segundo a **Escape**.

Pulsando **Start+Botón 1** o **Select+Start** o pulsando y soltando la tecla **Keymapper** (para mandos sin el botón **Start**), se entra el modo SHIFT, desde el cual se pueden activar distintos tipos de acciones, según la siguiente tabla:

	SHIFT MODE										
ZX-UNO	× 1	× 2	× 3	× 4	× 5	× 6	× 7	× 8	× 9	× 10	× 11
UP	KEY 'R'	KEY '0'	KEY '1'	KEY '2'	KEY '3'	KEY '4'	KEY '5'	KEY '6'	KEY '7'	KEY '8'	KEY '9'
DOWN	RESET	MRESET	ROMS	CORES	BIOS						
LEFT	KEYB MODE	KEY 'ESC'	KEYMAP SELECT	VIDEO SELECT							
RIGHT	NMI	LOAD "" (128K)	LOAD "" (48K)								
B1+B2	DB15/DB9 DB9 x 2										
START+UP	KB SPEED UP	KB SPEED UP	KB SPEED UP								
START+DOWN	KB SPEED DOWN	KB SPEED DOWN	KB SPEED DOWN								
B1+LEFT	DISABLE HOST->KB										
B1+RIGHT	ENABLE HOST->KB										

Así, por ejemplo, para hacer un reset del ZX-Uno, se ha de pulsar (en modo SHIFT) 1 vez abajo y luego botón 1.

## Actualización de firmware

Para poder aprovechar nuevas versiones se ha de actualizar la memoria flash de Joy2PS2 con la versión de firmware correspondiente.

Para poder instalar el firmware, se ha de hacer desde un ordenador (Windows, Mac, Linux), usando el software de actualización de binarios para AVR [XLoader para Windows](#) o bien con [AVRDUDE](#).

Se puede descargar la última versión del firmware [desde el repositorio oficial](#).

# 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-Uno](#), 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.

## 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, **"** 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-Uno (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, **"** 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 bloque de carga de la cinta, mientras el sonido se produce y el core del ZX-Uno 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 deseé 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.

[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 el orden de unas pocas decenas de ohmios. Esto, a veces, puede ser insuficiente para la entrada del ZX-Uno.



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-Uno (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, **"** 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.

[pzx2wav](#) 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](#).

[Audiotap](#) es un programa con interfaz gráfica para Windows (pero que se puede usar también con Wine en Linux o Mac) que permite convertir ficheros [TAP](#) de Commodore como [WAV](#) de audio.

[Comodoro 2.0](#) es un script que permite convertir a formato [TAP](#) y [TZX](#) para poder realizar cargas por audio en Commodore Pet, VIC-20, 16 y 64.

[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.

[cgc2wav](#) permite convertir ficheros [CAS](#) de EACA Colour Genie a fichero `WAV` de audio.

[castool](#) es una herramienta disponible con el software [MAME](#) y que permite convertir muchos tipos de fichero de cinta a fichero [WAV](#) de audio.

[mzf2wav](#) convierte ficheros de cinta [MZF](#) en [WAV](#).

# 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 SD, 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.ZX1 -w -o FLASHempty.ZX1
```

Donde **FLASH.ZX1** es la ruta al fichero obtenido desde el core de Spectrum, y **FLASHempty.ZX1** es la ruta al nuevo fichero "limpio".

## Mostrar contenido de una imagen

Para ver el contenido de una imagen llamada **FLASH.ZX1** (cores instalados y algunos datos de configuración), se puede usar el comando

```
... zx123_tool.py -i FLASH.ZX1 -l
```

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

```
... zx123_tool.py -i FLASH.ZX1 -l -r
```

## Modificar la BIOS de una imagen

Para modificar la BIOS de un fichero llamado **FLASH.ZX1**, usando la BIOS en otro fichero llamado **FIRMWARE.ZX1**

```
...zx123_tool.py -i FLASH.ZX1 -a BIOS,FIRMWARE.ZX1
```

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.ZX1**, usando la BIOS en otro fichero llamado **FIRMWARE.ZX1**, y además configurar el modo de vídeo en VGA:

```
...zx123_tool.py -i FLASH.ZX1 -a BIOS,FIRMWARE.ZX1 -m 2
```

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.ZX1 -a ROM,5,xdlh17,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>	<b>Habilitar teclado issue 3 (en vez de issue 2)</b>
<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 **TBBLUE.ZX1**, llamándolo **TBBlue**, usar un comando como este:

```
...zx123_tool.py -i FLASH.ZX1 -a 'CORE,3,TBBlue,TBBLUE.ZX1'
```

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

```
...zx123_tool.py -i FLASH.ZX1 -a 'CORE,3,TBBlue,TBBLUE.ZX1' -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.ZX1 -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.ZX1**, creando uno nuevo llamado **FLASHnew.ZX1**, instalar la BIOS desde el fichero **FIRMWARE.ZX1**, configurar el modo de vídeo en VGA, 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 **TBBLUE.ZX1**, llamándolo **TBBlue**, configurado como el core por defecto, usar un comando como este:

```
... zx123_tool.py -i FLASH.ZX1 -w -o FLASHnew.ZX1 -a BIOS,FIRMWARE.ZX1 -m 2 -a
ROM,5,xdnlh17,Spec48,48.rom -a 'CORE,3,TBBlue,TBBLUE.ZX1' -c 3
```

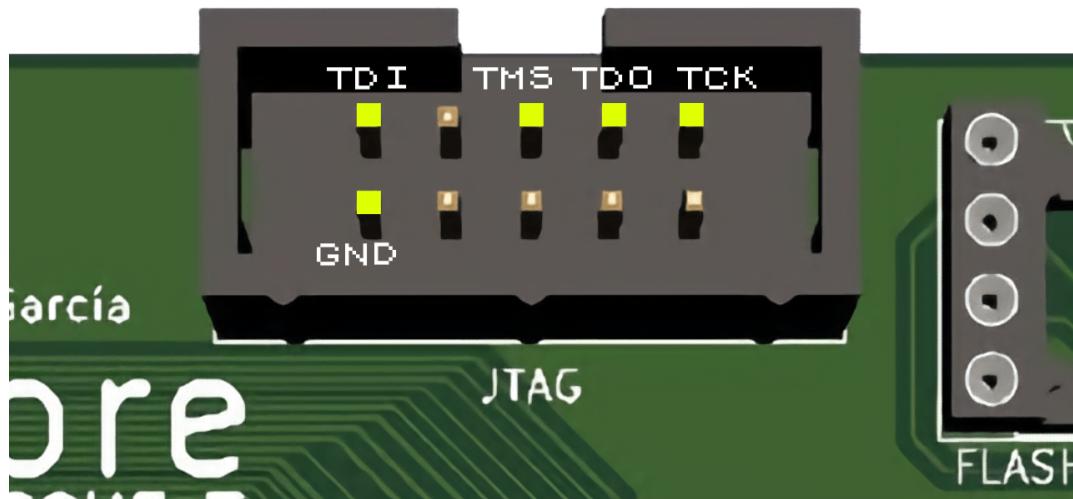
# Recuperación del firmware

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-Uno 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 ZXUnCore. Para ello, usar la siguiente imagen como referencia.



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



En el caso de utilizar USB-Blaster, ZXUnCore tiene la distribución adecuada para usar directamente el conector 2x5 incluido.

## 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) y, opcionalmente, un adaptador USB-Blaster
- Tarjeta SD para el ZX-Uno con la primera partición en formato FAT16 o FAT32
- Teclado y monitor para conectar el ZX-Uno

### Software necesario:

- Imagen Flash y recovery para ZX-Uno, del [repositorio de Github](#) y el [foro de ZX-Uno](#)

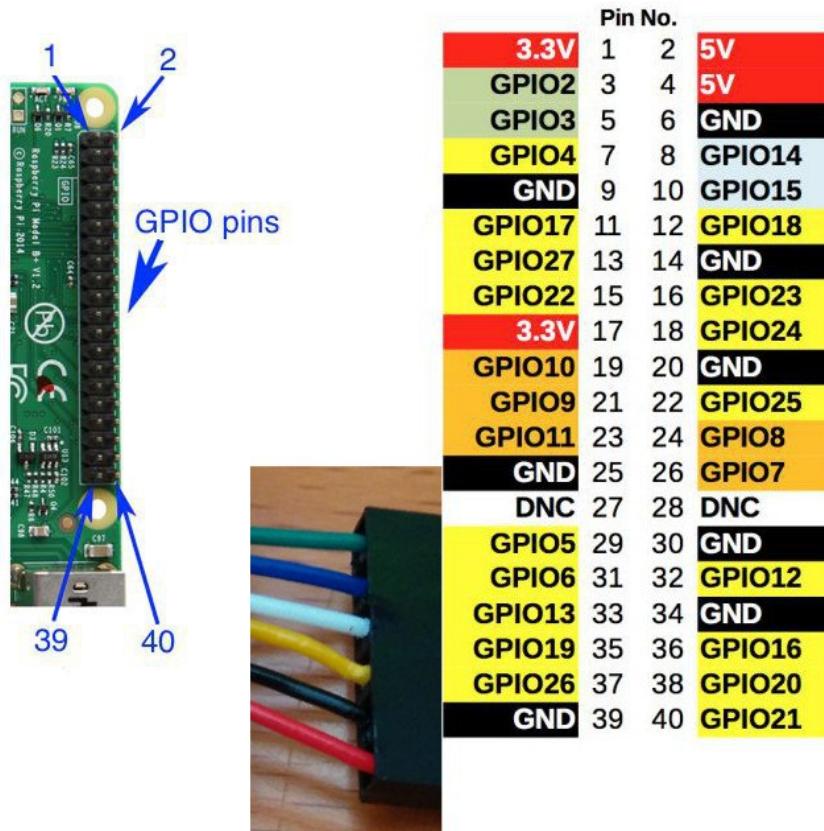
### Pasos a seguir:

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 [tal y como se explica anteriormente](#). Si se va a hacer la conexión usando GPIO, 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-Uno	GPIO	Pin Raspberry Pi
TMS	GPIO#6	<b>31</b>
TDI	GPIO#13	<b>33</b>
TDO	GPIO#19	<b>35</b>
TCK	GPIO#26	<b>37</b>
GND	GND	<b>39</b>

4. Copiar en la Raspberry Pi el fichero [recovery.bit](#) obtenido anteriormente del [foro de ZX-Uno](#). En nuestro ejemplo, se dejará en [/home/pi/zxuncore/unbrick/](#)

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

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

10. Asegurarnos de que estamos en el directorio donde se encuentra el archivo **recovery.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:

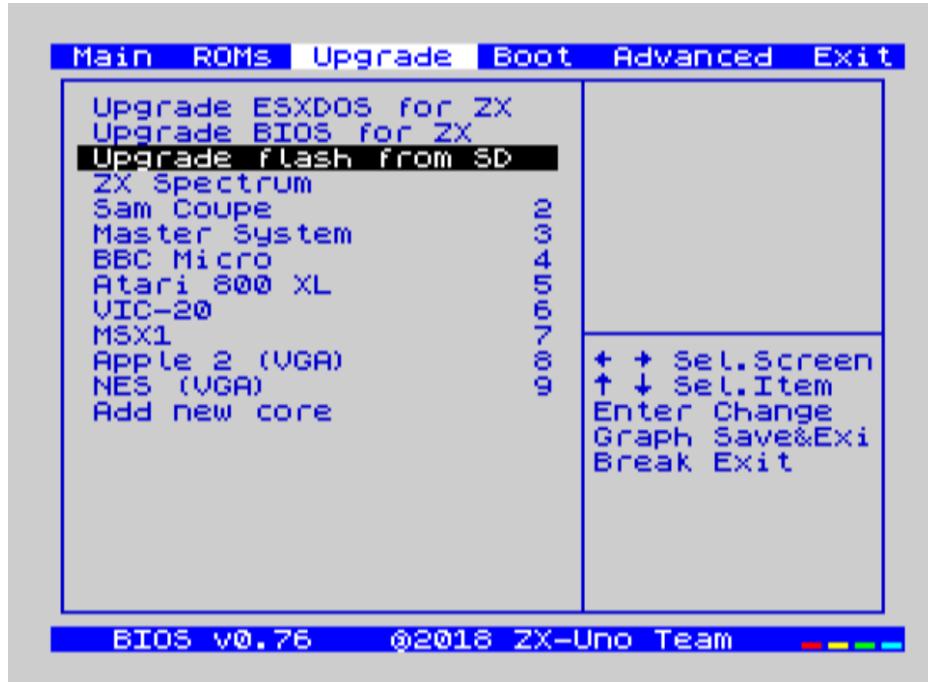
```
cd /home/pi/zxuncore/unbrick  
sudo openocd -f /home/pi/zxuncore/unbrick/raspberrypi2-native.cfg -f  
/usr/local/share/openocd/scripts/cpld/xilinx-xc6s.cfg -c "init; xc6s_program xc6s.tap;  
pld load 0 recovery.bit ; exit"
```

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.bit ; exit"
```

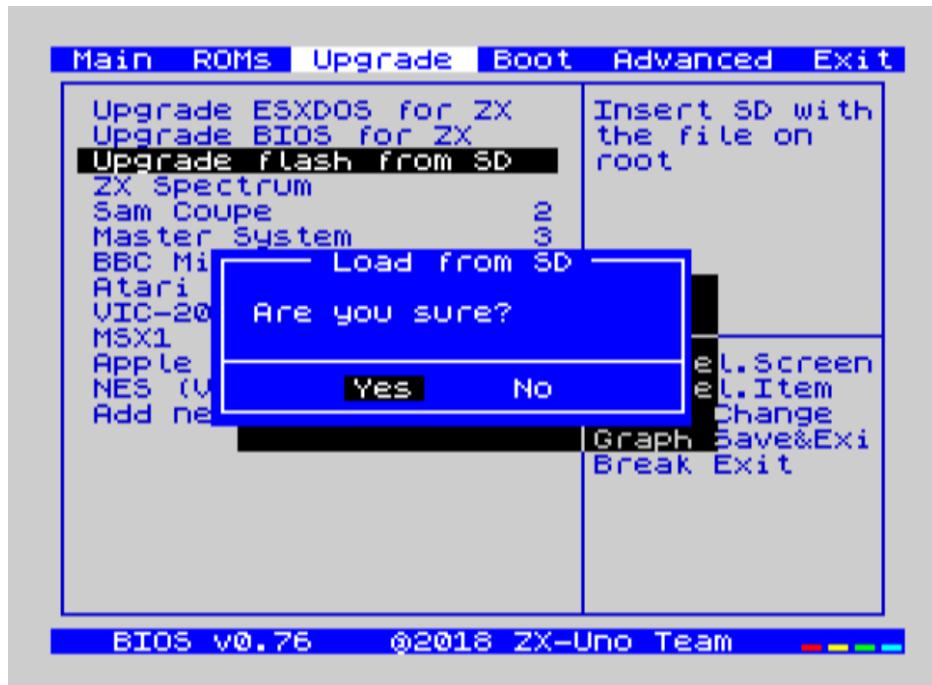
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+, cuando cualquier modo de teclado excepto el modo **ZX** normal, véase el [apartado dedicado a los modos de teclado](#) para más información) 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-Uno la tarjeta SD con la primera partición en formato FAT16 o FAT32, y en la que habremos copiado el fichero **FLASH.ZX1** [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, así que, en el caso de ZXUnGo+, puede que no funcionen las combinaciones como **Caps Shift+5**, etc. En este caso, se ha de cambiar el modo de teclado a cualquiera excepto el modo **ZX** normal (véase el apartado dedicado a los modos de teclado de ZXUnGo+ para más información), para que funcionen de nuevo temporalmente.

15. Tras unos minutos, el proceso finalizará, y podremos comprobar como, al apagar y encender, el ZX-Uno vuelve a arrancar correctamente.



Si no se obtiene imagen, pulsar de nuevo **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+) 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 ZXUnGo+, 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+Y**)
2. Cambiar al cualquier modo de teclado excepto el modo **ZX** normal (véase el [apartado dedicado a los modos de teclado](#) de ZXUnGo+ para más información)
3. Reiniciar el ZXUnGo+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+F** y luego **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+Y**)
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 ZXUnGo+ y volver a encenderlo

## Recuperación usando MacOS y Cable USB-Blaster

### Material necesario:

- Cable USB-Blaster [preparado con las conexiones adecuadas para ZXUnCore](#)
- Imagen Flash y recovery para ZX-Uno, del [repositorio de Github](#) y el [foro de ZX-Uno](#)

### 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.ZX1** en la raiz de la tarjeta SD para el ZX-Uno.

5. Conectar el cable USB-Blaster al ZXUnCore y al Mac

6. Encender el ZX-Uno

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

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

```
cable usbblaster
detect
pld load recovery.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

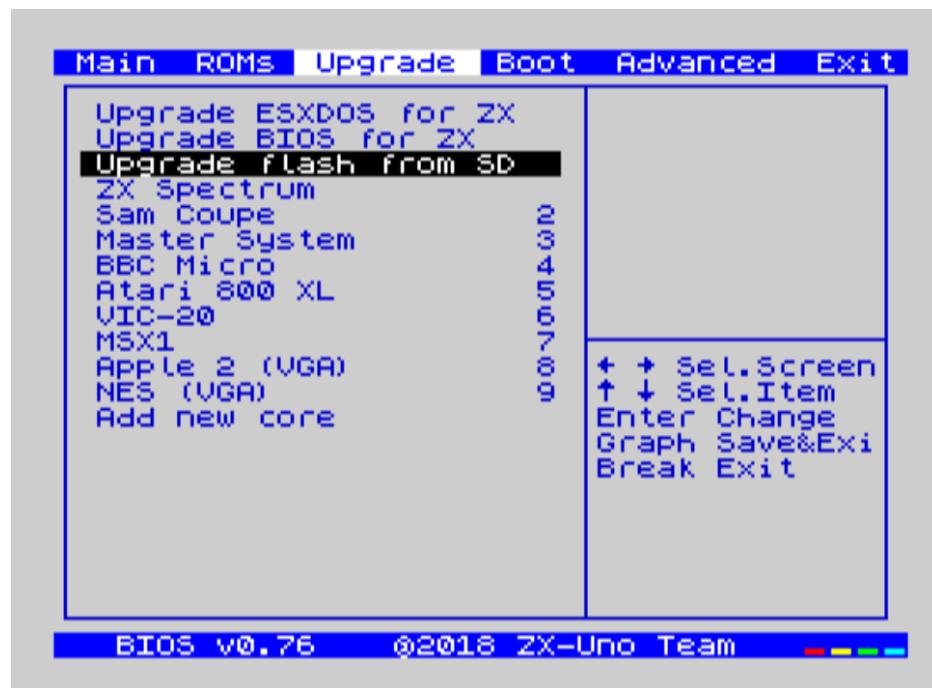
UrJTAG is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for UrJTAG.

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> █
```

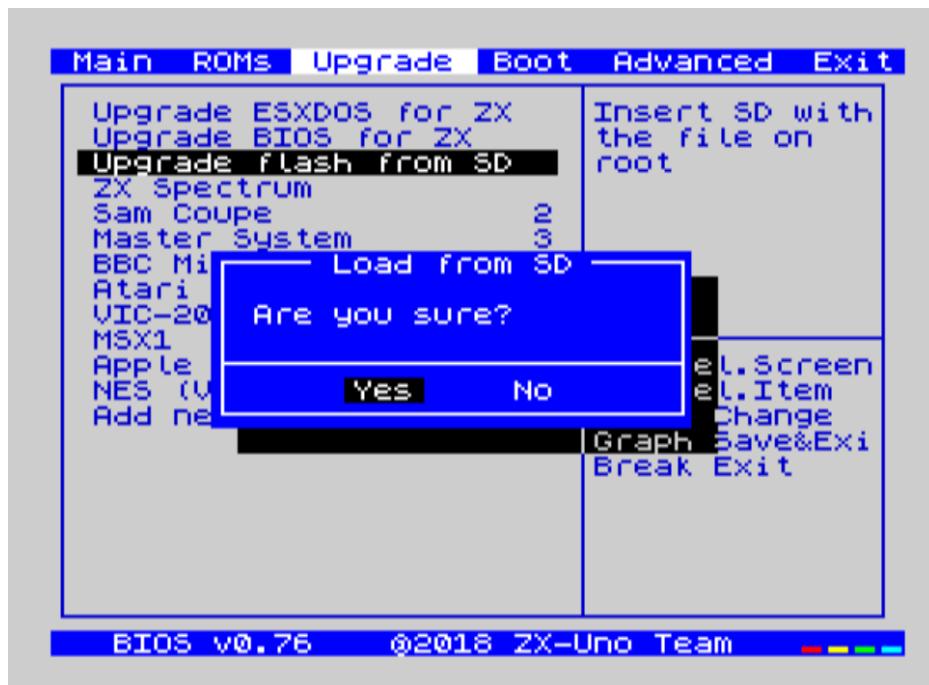
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+) 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-Uno la tarjeta SD con la primera partición en formato FAT16 o FAT32, y en la que habremos copiado el fichero **FLASH.ZX1** 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, así que, en el caso de ZXUnGo+, puede que no funcionen las combinaciones como **Caps Shift+5**, etc. En este caso, se ha de cambiar el modo de teclado a cualquiera excepto el modo **ZX** normal (véase el apartado dedicado a los modos de teclado de ZXUnGo+ para más información), para que funcionen de nuevo temporalmente.

13. Tras unos minutos, el proceso finalizará, y podremos comprobar como, al apagar y encender, el ZX-Uno vuelve a arrancar correctamente.



Si no se obtiene imagen, pulsar de nuevo **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+) 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 ZXUnGo+, 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+Y**)
2. Cambiar al cualquier modo de teclado excepto el modo **ZX** normal (véase el **apartado dedicado a los modos de teclado** de ZXUnGo+ para más información)
3. Reiniciar el ZXUnGo+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+F** y luego **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+Y**)
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 ZXUnGo+ 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. <<#\_preparación\_del\_cableado>>, preparado con las conexiones adecuadas para ZXUnCore>
- Imagen Flash y recovery para ZX-Uno. 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](#)

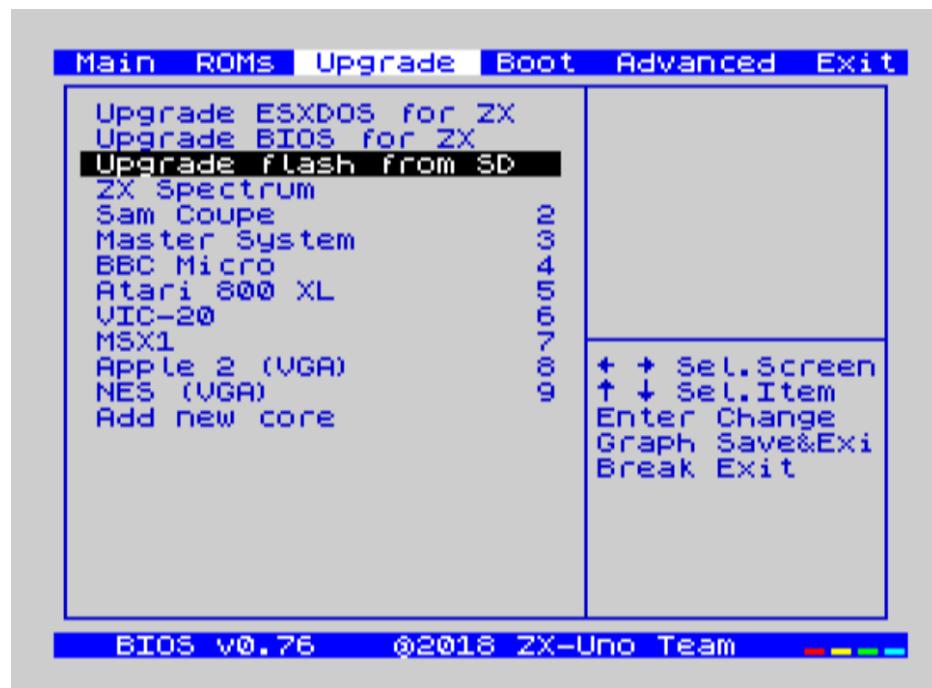
### Pasos a seguir:

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.ZX1** en la raiz de la tarjeta SD para el ZX-Uno.
5. Conectar el cable USB-Blaster al ZXUnCore y al PC con Windows
6. Encender el ZX-Uno
7. Asegurarnos de que estamos en el directorio donde se encuentra el archivo **recovery.bit**, y lanzar **jtag.exe**.
8. Aparecerá una consola donde se han de escribir los comandos:

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

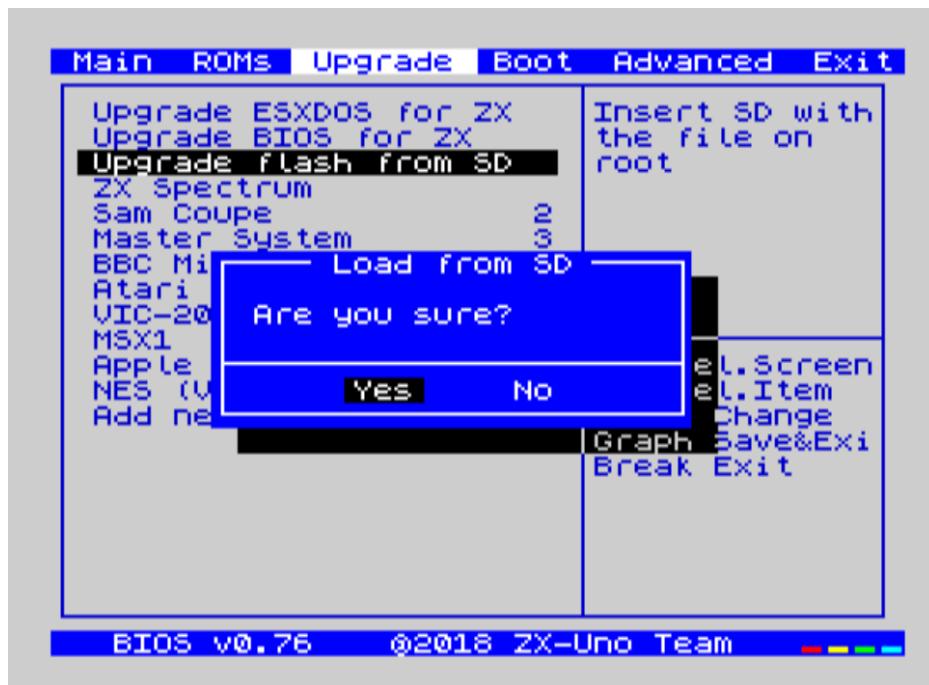
- Si todo va bien, veremos cómo cambia el estado de los LED de la FPGA y veremos la imagen de la BIOS en el monitor.

En el caso de que no se vea imagen, pulsar **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+) 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-Uno la tarjeta SD con la primera partición en formato FAT16 o FAT32, y en la que habremos copiado el fichero **FLASH.ZX1** 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, así que, en el caso de ZXUnGo+, puede que no funcionen las combinaciones como **Caps Shift+5**, etc. En este caso, se ha de cambiar el modo de teclado a cualquiera excepto el modo **ZX** normal (véase el apartado dedicado a los modos de teclado de ZXUnGo+ para más información), para que funcionen de nuevo temporalmente.

13. Tras unos minutos, el proceso finalizará, y podremos comprobar como, al apagar y encender, el ZX-Uno vuelve a arrancar correctamente.



Si no se obtiene imagen, pulsar de nuevo **Bloq. Despl.** (**Caps Shift+Symbol Shift+Y** en ZXUnGo+) 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 ZXUnGo+, 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+Y**)
2. Cambiar al cualquier modo de teclado excepto el modo **ZX** normal (véase el **apartado dedicado a los modos de teclado** de ZXUnGo+ para más información)
3. Reiniciar el ZXUnGo+ sin que se pierda la configuración del teclado (**Caps Shift+Symbol Shift+F** y luego **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+Y**)
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 ZXUnGo+ y volver a encenderlo

# Referencias

## 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 SHFT	12	F0,12
L CTRL	14	F0,14
L GUI	E0,1F	E0,F0,1F
L ALT	11	F0,11
R SHFT	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

# Enlaces

[ZX-Uno](#)

[foroFPGA](#)

[canal oficial de Telegram de ZX-Uno](#)

[ZX-Uno FAQ](#)

[Repositorio de ZXUno4All](#)

[ZXUnCore en foros va de Retro](#)

[ZXUno4All en foros va de Retro](#)

[Repositorio de ZXUnoPS2](#)

[Repositorio de joy2ps2](#)

[Guía rápida del ZX-Uno](#)

[Wiki de ZX-Uno.](#)

[Core ZX Spectrum](#)

[The ZX Spectrum +3e Homepage](#)

[Sharing a +3e disk with PC \(FAT\) partitions](#)

[Configuración de teclado de ZX-Uno.](#)

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

[WiFi \(RetroWiki\)](#)

[Cargando Leches 2.0](#)

[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-Uno \(Kyp\)](#)

[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 on GitHub](#)

[TOSEC: Magnavox Odyssey 2 \(2012-04-23\)](#)

[Videopac G7000 / Odyssey2 for FPGA](#)

[zx123\\_tool](#)

[esxdos](#)

[ZX-Modules.](#)

[Long Filename Browser de Bob Fossil](#)

[New NMI Handler de Dr. Slump](#)

[Fuzix](#)

[SmartROM](#)

[ZX Tape Player](#)

[PlayZX](#)

[Tapir](#)

[Fuse Utilities](#)

[MakeTSX](#)

[Lynx2Wav](#)

[2lynx2wav](#)

[cgc2wav](#)

[mzf2wav](#)

Manual de ZXUno4ALL : Enlaces

[RetroConverter](#)

[Lince Script](#)

[Programming a Spartan 6 with a Raspberry Pi](#)

[Tutorial para desbriquiar el ZX-Uno con una Raspberry](#)