# ZXDOS+ and gomaDOS+ Manual

kounch

Version 1.0.3

# Index

# Introduction

ZXDOS+ y gomaDOS+ are the continuation of ZX-Uno a hardware and software project based on an FPGA board programmed to work like a ZX Spectrum computer, and created by the ZX-Uno team: Superfo, AVillena, McLeod, Quest and Hark0.

Over time, the project has been growing, and now it is possible to install different software configurations (cores) in the flash memory of the FPGA, which work like different systems than the ZX Spectrum, being able to choose to start the ZXDOS+ with the desired configuration among all those installed.

ZXDOS+ and gomaDOS+ official web page is http://zxdos.forofpga.es.

Most of the functions and features of ZXDOS+ and gomaDOS+ are the same, so this document will generally talk about ZXDOS+, indicating the differences with gomaDOS+ where necessary.

# Initial Setup

In order to be able to set up and use a ZXDOS+ or gomaDOS+ you need, at least, the following:

- USB charger, a TV or other device that offers USB power
- VGA cable and monitor
- PS/2 keyboard (in the case of ZXDOS +)

In order to take advantage of its full potential, you should also have:

- A microSD card, not necessarily very large
- PC speakers to connect to the audio output, or a stereo jack cable to two red/white RCA connectors to connect to the TV
- A standard Atari joystick, such as a Megadrive DB9 gamepad
- A PS/2 mouse

## microSD card formatting

In order to use a microSD card, it has to be formmatted with, at least, one FAT16 or FAT32 format (depending on the case, one or the other format is recommended for compatibility with different third-party cores). It must be the first partition if there are more than one.

FAT16 partitions have a maximum size of 4GB

### Windows

For simple configurations, and cards of the correct size (less than 2GB for FAT16 or less than 32GB for FAT32), you can use the official formatting tool of the SD Association .

For other, more complex, configurations, and depending on operating system version, you may use the command line tool `diskpart` or Windows Disk Managemente GUI.

### MacOS

For simple configurations, and cards of the correct size (less than 2GB for FAT16 or less than 32GB for FAT32), you can use the official formatting tool of the SD Association or Disk Utility, which is included with the operating system.

In other case, you should use the command line.

For example, to format a card, shown as `disk6`, with only one FAT16 partition (if the card size is less than 2GB):

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

To split it into two FAT16 partitions of the same size (if the card size is 4GB or less):

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

To create two FAT 16 partitions (e.g. to use MSX core) and have the rest of space as another FAT32 partition (for cards more than 8GB in size):

```
diskutil unmountDisk /dev/disk6
diskutil partitionDisk /dev/disk6 MBR %DOS_FAT_16% ZXDOSPLUS 4G %DOS_FAT_16% EXTRA 4G
"MS-DOS FAT32" DATA R
sudo newfs_msdos -F 16 -v ZXDOSPLUS -b 4096 -c 128 /dev/rdisk6s1
sudo newfs_msdos -F 16 -v EXTRA -b 4096 -c 128 /dev/rdisk6s2
```

`diskutil` cannot create FAT16 partitions which are bigger than 2G and also format them. That's why, in this example, after only creating the partitions, we have to format them.

To create one FAT32 4GB partition (e.g. to use with Amstrad CPC core), and then have the rest of space available as a second FAT32 partition (for cards of more than 4GB):

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

#### Linux

There are a lot of tools for Linux that can format and/or partition an SD card (`fdisk`, `parted`, `cfdisk`, `sfdisk` or `GParted` to name a few). It should only be taken into account that the partition scheme must always be MBR, and the first partition (the one that will be used for esxdos) must be primary partition.

## esxdos

esxdos is a firmware for the DivIDE/DivMMC hardware interfaces (which ZXDOS+ implements). This allows access to storage devices such as a microSD card. It includes commands similar to those of UNIX, although to use them you must precede them with a period, for example `.ls`,` .cd`, `.mv`, etc.

For it to work, it is necessary to include the corresponding files in the first partition of the microSD card.

At the time of writing this document, the version included with ZXDOS+ is 0.8.6, and it can be downloaded from the official website at this link.

Once downloaded and extracted, you have to copy the directories `BIN`, `SYS` and `TMP`, and all of their content, to the root of first partition of the microSD card.

If everything has been done correctly, when you turn on the ZXDOS+ Spectrum core, you will see how esxdos detects the card and loads the necessary components to work.

It is also recommended to add the specific esxdos commands for ZXDOS+. These can be obtained from the project source page (here and here), and are as follows:

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

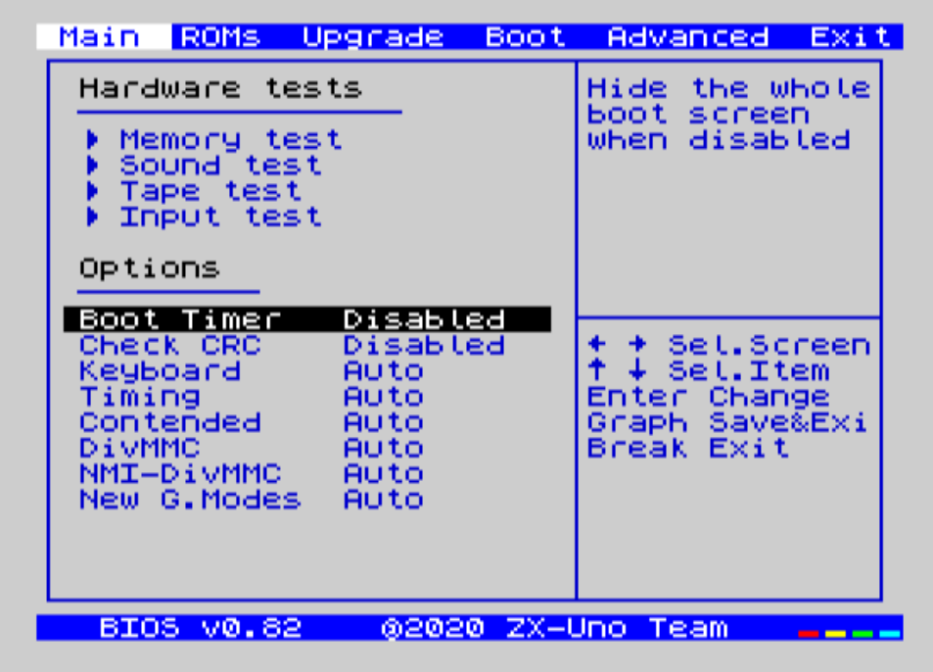
<<#_zxdos+_commands,It is explained later> what each of them does.

# BIOS

Pressing the F2 key during boot will access the BIOS setup. The BIOS firmware is the first program that runs when the ZXDOS+ is turned on. The main purpose of BIOS is to start and test the hardware and load one of the installed cores.

Using cursor keys (left and right), you can navigate through the BIOS setup screens. With up and down keys you can choose the different elements of each screen and, with the Enter key, it is possible to activate and choose the options for each of these. Esc key is used to close open option windows without applying any action.

## Main

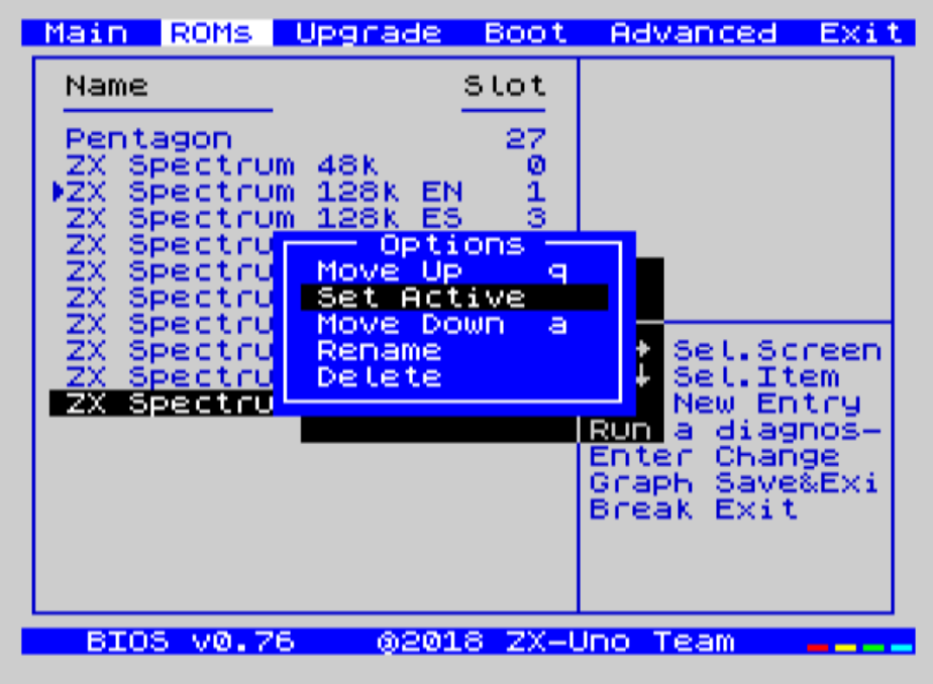



In the first configuration screen, in addition to being able to run several tests, you can define the default behavior for the following:

- Boot Timer: Sets how long the boot screen is available (or hiding it completely)
- Check CRC: Check ROM integrity when loading (more secure) or bypassing it (faster)
- Keyboard
- Timing: ULA Behaviour (48K, 128K, Pentagon Modes)
- Contended
- DivMMC
- DivMMC NMI Support
- New Graphic Modes Support (ULAPlus, Timex, Radastan)

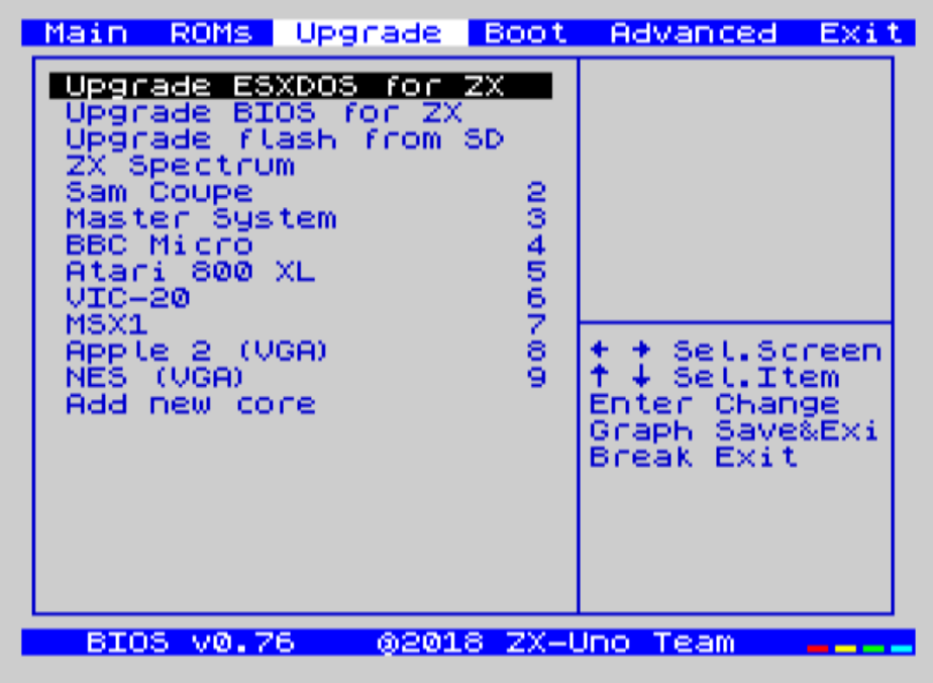
More technical information can be found on de ZX-Uno Wiki.

### ROMs



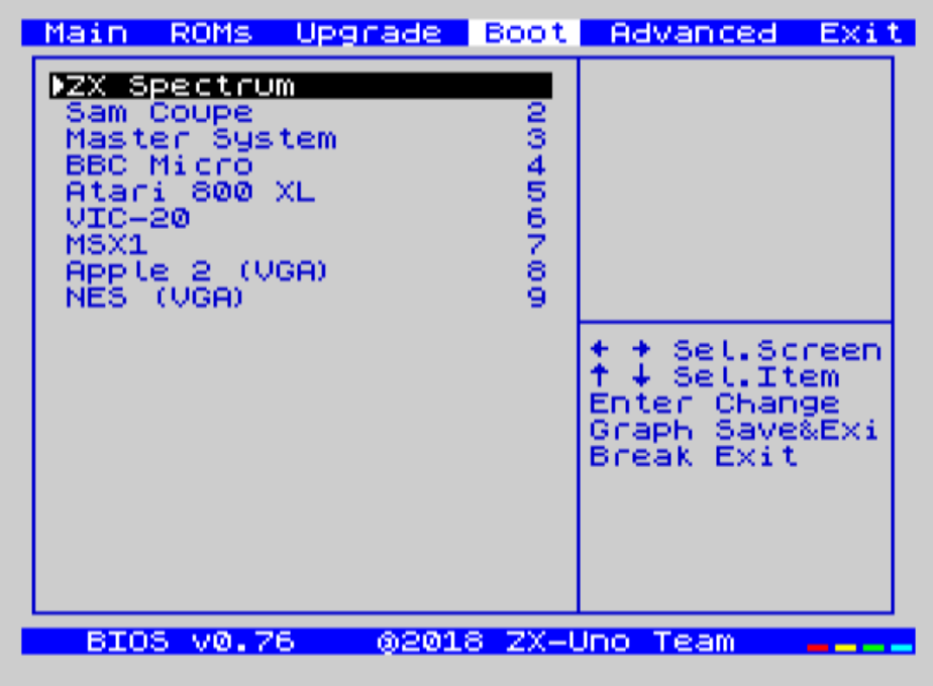

The second screen shows the installed ZX Spectrum ROMs. You can reorder (Move Up, Move Down), rename or delete each of them, as well as choose the one that will be loaded by default at startup (Set Active ).

### Upgrade



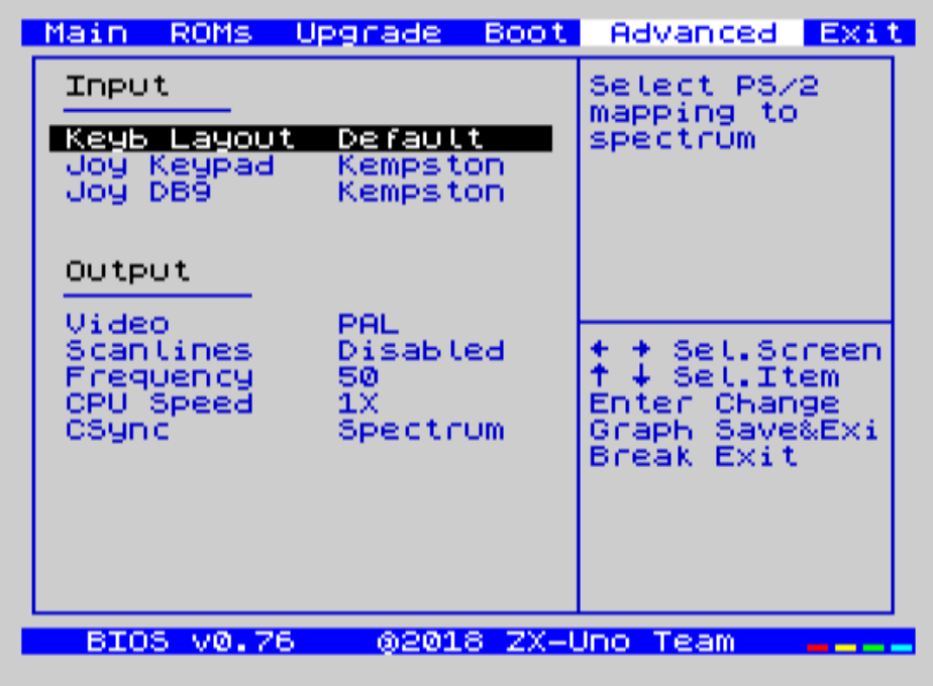

*Upgrade* screen is used to perform the different updates of the Flash memory content: esxdos, BIOS, Cores, etc. (see the section corresponding to updates for more information).

## Boot




In the *Boot* screen you can choose which one of the installed cores is loaded by default at startup.

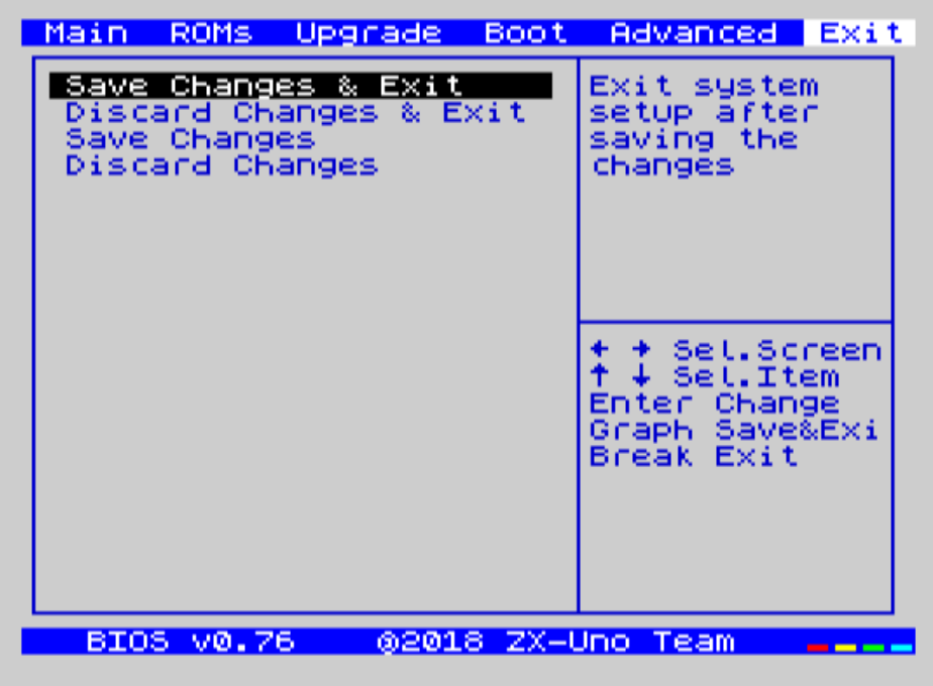
## Advanced




The Advanced configuration screen is used to edit the following settings:

- Keyboard layout (Keyb Layout): See the corresponding section for more information)
- Joystick behavior when emulated with the numeric keypad (Joy Keypad): Kempston, Sinclair Joystick 1, Sinclair Joystick 2, Protek or Fuller
- Behavior of a joystick connected to the port (Joy DB9): Kempston, Sinclair Joystick 1, Sinclair Joystick 2, Protek, Fuller or simulate the keys `Q`, ` A `, `O`, ` P `, `Space` and `M`
- Video output: PAL, NTSC or VGA
- Scanline simulation: Enabled Disabled
- VGA horizontal frequency: 50, 51, etc.
- CPU speed: Normal (1x) or accelerated (2X, 3X, etc.)
- Csync: Spectrum or PAL

## Exit




Finally, from the last screen you can:

- Exit BIOS configuration saving changes
- Discard changes and exit
- Save changes without exiting
- Discard Changes

# ZX Spectrum

The main core is the one implementing a ZX Spectrum computer. This core is special, and it cannot be substituted for another that is not a ZX Spectrum, since the ZXDOS+ uses it for its operation.

These are some of its main characteristics:

- ZX Spectrum 48K, 128K, Pentagon and Chloe 280SE implementation
- ULA with ULAplus, Timex and Radastan modes (including hardware scroll and selectable palette group)
- Ability to disable memory contention (for Pentagon 128 compatibility)
- Ability to choose the keyboard behavior (issue 2 or issue 3)
- Possibility to choose the timing of the ULA (48K, 128K or Pentagon)
- Control of screen framing, configurable for type of timing, and possibility to choose between original Spectrum synchronisms or progressive PAL standard.
- Timex horizontal MMU support with HOME, DOC and EXT banks in RAM.
- Programmable raster interruption in line number, for any TV line.
- Possibility of activating/deactivating memory bank management registers, for better compatibility with each implemented model
- Ability to activate / deactivate the devices incorporated into the core to improve compatibility with certain programs
- ZXMMC support for + 3e and DIVMMC support for esxdos and compatible firmwares
- Turbo Sound support
- SpecDrum support
- Each channel A, B, C of the two AY-3-8912, beeper and SpecDrum chips can be directed to the left, right, both or neither outputs, allowing the implementation of configurations such as ACB, ABC, etc.
- Real joystick and keyboard joystick support with Kempston, Sinclair 1 and 2, Cursor, Fuller and QAOPSpcM protocol.
- Turbo mode support at 7MHz, 14MHz, 28MHz
- Keyboard support (PS/2 protocol) and user-configurable mapping from within Spectrum itself.
- PS/2 mouse support emulating the Kempston Mouse protocol.
- Possibility of video output in composite video mode, RGB 15kHz, or VGA.
- User selectable vertical refresh rate to improve compatibility with VGA monitors.
- Multicore boot support: from the Spectrum you can select an address of the SPI Flash and the FPGA will load a core from there.

# Keyboard

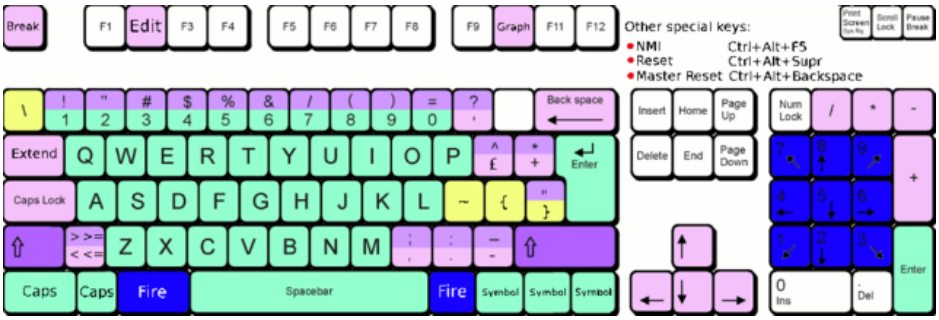
The keyboard map (physical keys of the keyboard assignment to the keystrokes that are presented to the different cores) is changed using the `Advanced` menu of the BIOS. There are three different maps to choose from: Spanish (default), English, and Spectrum (advanced).

You can also change it using the `keymap` utility. Inside `/ bin` you have to create a directory called `keymaps` and copy inside the keyboard map files that you want to use. For example, to switch to the US map you have to write `.keymap us` from esxdos.

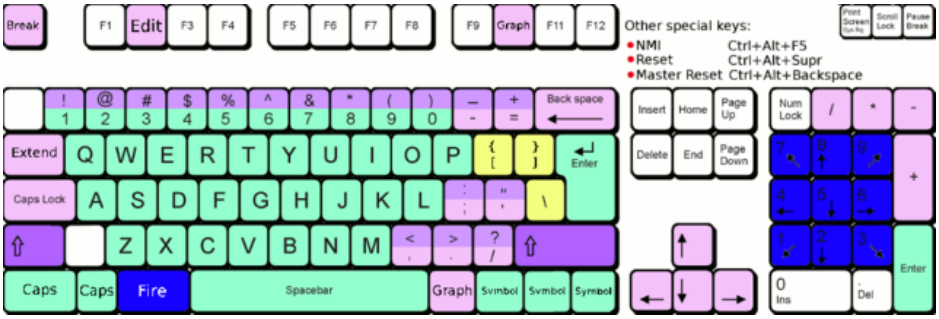
For the map to be preserved after a master reset, it has to be selected as `Default` in the BIOS.

For more information, see this message in the ZX-Uno forum.

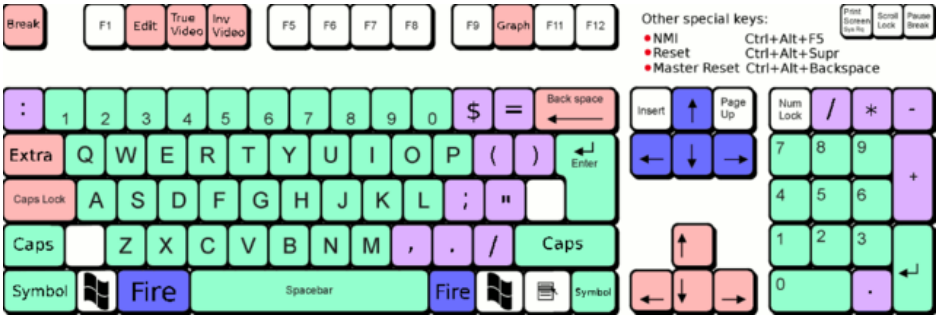
## Spanish




## English




## Spectrum



# Special keys and buttons

gomaDOS+ keyboard, being similar to the original ZX Spectrum keyboard, lacks some of the existing keys on a modern PC keyboard. To correct this, you can change the way the Spectrum keyboard is interpreted between two different modes: Conventional Mode (which is the default mode) and Full Mode.

To switch between the two keyboard modes, use the key combination `Caps Shift+Symbol Shift+F` and then press `D`.

Special keys that can be used while running the main core (ZX Spectrum):

- `Esc`: BREAK
- `F2` (`Caps Shift+Symbol Shift+2` on gomaDOS+, Full Mode): Edit
- `F5` (`Caps Shift+Symbol Shift+5` on gomaDOS+, Full Mode - or `Caps Shift+Symbol Shift+F` and then `Y` -): NMI
- `F7` (`Caps Shift+Symbol Shift+7` on gomaDOS+, Full Mode): Play or pause when playing .PZX files
- `F8` (`Caps Shift+Symbol Shift+8` on gomaDOS+, Full Mode): Rewind .PZX file to the previous mark
- `F10` (`Caps Shift+Symbol Shift+0` on gomaDOS+, Full Mode): Graph
- `F12` (`Caps Shift+Symbol Shift+W` on gomaDOS+, Full Mode): Turbo Boost. Speeds up CPU to 28MHz while pressed (beginnig with core EXP27).
- `Ctrl+Alt+Backspace` (`Caps Shift+Symbol Shift+B` on gomaDOS+): Hard reset. Backspace is the delete key, located in the top-right portion of the keyboard, above `Enter`.
- `Ctrl+Alt+Supr` (`Caps Shift+Symbol Shift+N` on gomaDOS+): Soft reset.
- `Scroll Lock` (`Caps Shift+Symbol Shift+G` on gomaDOS+): Switches between compositve and VGA video modes.

During startup:

- `F2` (`Caps Shift+1` on gomaDOS+, Full Mode) Enter BIOS setup
- `Caps Shift` or `Cursor down` (`Caps Shift+2 on gomaDOS+): Core selection menu
- `Esc` (`Caps Shift+Espacio` on gomaDOS+): ZX Spectrum core ROM selection menu
- `R`: Loads the Spectrum core ROM in "real" mode, disabling esxdos, new graphics modes, etc.
- `/` (numeric keyboard) (`Symbol Shift+V` on gomaDOS+): Load ZX Spectrum core ROM in "root" mode
- Number from `1` to `9`: Load the core in the flash location corresponding to that number

# ROMs

The ZX Spectrum core has can be initialized using different ROM versions (48K, 128K, Plus 2, etc.). These are stored in the flash memory of the ZXDOS+, and you can choose which one to load by pressing the `Esc` key during boot. You can also define the ROM that you want to load by default using the BIOS setup.

See the updates section for more information on how to expand or modify the ROMs stored in flash memory.

# esxdos

## Basic Guide

There are two different kind of esxdos commands, the so-called "DOT" commands, which, as the name suggests, begin with a period, and the commands that are extensions to the existing ones in BASIC.

The main "DOT" commands are the following:

- `128`: Para enter 128K mode from within 48K mode
- `cd`: Change current working directory
- `chmod`: Change file attributes
- `cp`: Copy a file
- `divideo`: Play a DivIDEo (.DVO) video file
- `drives`: Show currently available drives
- `dskprobe`: Utility which shows low level content of an storage device
- `dumpmem`: Can dump RAM memory content to a file
- `file`: Tries to recognize the type of data contained in a file (like the UNIX command)
- `gramon`: Monitor to search graphics, sprites, fonts, etc. in RAM memory
- `hexdump`: Shows the contents of a file using hexadecimal notation
- `hexview`: Allow to see and navigate through the contents os a file using hexadecimal notation
- `launcher`: Creates a shortcut (launcher) to open directly a TAP file
- `ls`: Show the content of a directory
- `lstap`: Show the content of a .TAP file
- `mkdir`: Create a directory
- `mktrd`: Create a .TRD disk file
- `more`: Show the content of a text file
- `mv`: Move a file
- `partinfo`: Show partition information of an storage device
- `playpt3`: Play .PT3 music file
- `playsqt`: Play .SQT music file
- `playstc`: Play .STC music file
- `playtfm`: Play .TFC music file
- `playwav`: Play .WAV audio file
- `rm`: Remove a file or a directory
- `snapload`: Load snapshot file

- `speakcz`: Reads text aloud using czech pronunciation
- `tapein`: Mounts a .TAP file so that it can be used then from BASIC using LOAD sentence
- `tapeout`: Mount a .TAP file so that it can be used then from BASIC using SAVE sentence
- `vdisk`: Mount a .TRD disk file to use with the TR-DOS environment (once all the drives have been mounted, you can enter TR-DOS emulation by typing: `RANDOMIZE USR 15616`)

Some BASIC extended commands are:

- `GO TO` to change the current drive and/or directory (e.g.: `GO TO hd1` or `GO TO hd0"games"`)
- `CAT` to show the content of a drive
- `LOAD` to lad a file from a drive (BASIC Program, SCREEN, CODE, etc. for example `LOAD *"Screen.scr" SCREEN$`)
- `SAVE` to save data in a file (e.g: `SAVE *"Program.bas"`)
- `ERASE` to delete a file

In addition, esxdos also has an NMI manager, an application that loads when NMI (F5) is pressed, and lets you browse the microSD card and load easily files (TAP, Z80, TRD, etc.). Pressing the "H" key invokes a help screen, which shows all the available keys.

## ZXDOS+ Commands

As explained in the installation part, there are a series of commands that are exclusive to ZXDOS+:

- `back16m`: Dumps to a `FLASH.ZX1` file, in the root directory of the SD card, the contents of a 16 Meg SPI Flash memory. After finishing, it is necessary to execute the command `.ls` so that the cache is written to the card
- `back32m`: Version of the backup command for 32 Meg SPI Flash memories. After finishing its execution, you must execute the command `.ls` to finish recording the cache on the microSD card. If not, the length of the file will be wrongly set to 0
- `corebios`: To upddate simultaneously ZX Spectrum core and BIOS
- `dmaplayw`: Plays .WAV file, which has to be 8 bits, unsigned y mand sampled at 15625 Hz
- `esprst`: Resets the WiFi ESP8266(ESP-12) module
- `iwconfig`: To configure the WiFi module
- `joyconf`: Configuration and tests for keyboard and DB joysticks
- `keymap`: Used to load a different keyboard map definition
- `loadpzx`: To load a .PZX tape file
- `playmid`: Plays .MID music files using the MIDI addon
- `playrmov`: Plays radastanian format video files `.RDM`). This command does not work on 48K mode.
- `romsback`: Dumps to a `ROMS.ZX1` file, in the root directory of the microSD card, all ZX Spectrum core ROMS which are stored in SPI flash memory
- `romsupgr`: Load from a `ROMS.ZX1` file, in the root directory of the microSD card, all ZX Spectrum core ROMS into SPI flash memory
- `upgr16m`: Load the conent of a `FLASH.ZX1` file, in the root directory of the microSD card, to a 16 Meg SPI Flash memory
- `upgr32m`: Version of the upgrade command for 32 Meg SPI Flash memories
- `zxuc`: Utility to configure al options of BIOS, which also can be stored in the microSD in configuration files that can be loaded later
- `zxunocfg`: Configuration utillity for certain features of ZX-Uno such as timings, contention, keyboard type, CPU speed, video type or vertical frequency

# Wi-Fi

Each gomaDOS+, and some models of ZXDOS+, include inside an ESP-12 module with an ESP8266 Wi-Fi chip, that can be easily used with a ZX Spectrum core (e.g., EXP27 160820 core) which has synthesized an UART device, that allows communication with the module.

There are two "DOT" commands for configuring software access to the module. Then can be downloaded from GitHub official repository:

- `esprst`, which restarts the module
- `iwconfig`, to register the Wi-Fi network name (SSID) and password, keeping them in the file `/sys/config/iw.cfg`.

For example:

```
.iwconfig mywifi mypassword
```

## Network tools for ZX-Uno pack

These are programs, developed by Nihirash and that are available to download from his web.

- `netman`: Utility to configure the ESP Wi-Fi chip for other programs from Nihirash. Does not work in 48K mode
- `uGophy`: Gopher client. Does not work in 48K mode
- `irc`: Internet Relay Chat client. Works better at 14 Mhz
- `wget`: Utility to download files with HTTP (does not work with HTTPS)
- `platoUNO`: PLATO client. Also works better at 14 Mhz. For more information about PLATO, check IRATA.ONLINE web

## FTP-Uno

FTP cliente developed by Yombo, available at GitHub.

Configuration steps:

1. Edit `FTP.CFG` file with all the required information (SSID and password, FTP server, etc.)
2. Copy `FTP.CFG` inside `/SYS/CONFIG/` in microSD card
3. Also copy `ftpUno.tap` to any place in the card
4. Start up ZXDOS+ y load the tape file `ftpUno.tap`

## UART Terminal

Program example included with ZXYLib C library, developed by yombo, that let's you send directly typed characters using the UART, and also see the result. Available to download at this link.

Once the file `UARTTERM.tap` is in the card and loaded, you can type several specific commands for ESP8266 chip. For example:

- `AT`. To check if ther is communication. `OK` would be the result if everything is fine
- `AT+RST`. To restart the chip. Exactly what `esprst` command does
- `AT+GMR`. To see some information, like firmware version, etc.
- `AT+CWMODE_CUR=1`. Put temporarily the chip into Wi-Fi client mode, until next restart
- `AT+CWMODE_DEF=1`. Put temporarily the chip into Wi-Fi client mode, and save it as default
- `AT+CWJAP_CUR="<WiFiNetwork>","<WiFiPassword>"`, where `<WiFiNetwork>` Wi-Fi ID of the network to connect to, and `<WiFiPassword>` the access password, connects temporarily to that network
- `AT+CWJAP_DEF="<WiFiNetwork>","<WiFiPassword>"`, connects to the network, and saves the settings as default in the chip flash memory
- `AT+CWAUTOCONN=1` sets the chip to connect automatically on boot to the default network (`AT+CWAUTOCONN=0` disables it)

You can see all the available commands reading the official documentation.

# Making RDM (RaDastan Movie) files

The PLAYRMOV "DOT" command plays radastanian format video files. To convert your own videos, you need makevideoradas, a utility that is available at SVN repository.

If using Windows, there is already an executable file (makevideoras.exe). For Linux or MacOS, you must have installed command line developer utilities in order to compile an executable

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

Apart from makevdideoradas, you need another two tools: ffmpeg and imagemagick. These can be installed with a package manager (apt, yum, pacmam, brew, etc.) or downloading the source code and compiling.

Now, the first step to convert our video (for example myvideo.mp4), is exporting the frames as 128x96 pixel BMP image files. We create a temporary file (img for this example), to store them.

```
mkdir img
(...)/ffmpeg -i myvideo.mp4 -vf "scale=128:96,fps=25" -start_number 0
img/output%05d.bmp
```

Now we transform the BMP files to 16 colours (v3) BMP files.

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

Finally, we assemble the .RDM file (in this example myvideo.rdm) and cleanup the temporary files and directory.

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

There is more information about all this process at this thread in Zona de Pruebas forums.

# Upgrade

## BIOS

To update the BIOS, a file named `FIRMWARE.ZX2` (for a ZXDOS+ with an FPGA LX16 board) or `FIRMWARE.ZXD` (for a ZXDOS+ with an FPGA LX25 board) must be obtained. The latest version of the firmware files can be downloaded from the official repository

Updating the firmware (BIOS) is delicate. It should not be done if it is not necessary. If doing so, ensure that the ZXDOS+ has uninterrupted power (such as a UPS or a laptop USB with battery).

Copy the file to the root of the MicroSD card, turn on and press `F2` to enter BIOS, select `Upgrade`, choose *"Upgrade BIOS for ZX"*, and then *"SDfile"*. The system will read the file `FIRMWARE···` and notify when finished.

## ROMs

To update the ROMs installed for ZX Spectrum, a file named `ROMS.ZX1` must be obtained, which must be copied to the MicroSD card. Boot the ZXDOS+ using a "rooted" ROM, and then just enter the command `.romsupgr`. This will burn all the ROMs, which will be available for use.

Remember that if the ZXDOS+ is started by pressing the `/` key (on the numeric keyboard) (`Symbol Shift+V` in gomaDOS+), then the default ROM of the ZX Spectrum core will be loaded in" root "mode.

To do the opposite process (save the ROMs in a `ROMS.ZX1` file), you can use the` .romsback` command.

`ROMS.ZX1` files can be easily edited with the http: // guest: zxuno@svn.zxuno.comsvn/zxuno/software/ZX1RomPack/[ZX1RomPack] utility. Although it is a Windows program, it works perfectly, for example using Wine or similar programs, either on MacOS or Linux.

## Cores

There are a number of available slots whre you can store cores (the number depends on the size of the SPI Flash of the ZXDOS+ model), the first slot being reserved for the main ZX Spectrum (this does not prevent having more ZX Spectrum cores in other slots as well of the first).

Official cores are available to download from GitHub repository.

To update or install a new core there are several possibilities.

The easiest way is to obtain the latest version of the file that defines the core, which will be a file that must be named `COREnn.ZX2` (for a ZXDOS + with an FPGA LX16 board) or `COREnn.ZXD` (for a ZXDOS + with an LX25 board), where `nn` is the slot number where to install (for example `CORE2.ZX2`

or CORE2.ZXD for slot 2).

Starting with BIOS version 0.80, files are named using the COREXXy.ZXn convention where XX *always* is a two-digit number. Thus, an old CORE4.ZXD file has to be renamed as CORE04.ZXD. The y part of the name is ignored, so longer and more descriptive names can be used (such as CORE04_example.ZXD).

Copy the file to the root of the microSD card, turn on and press F2 to enter BIOS. Choose Upgrade, select the row corresponding to the chosen core number (for example, 2 - just after Spectrum), press enter and then *" SD file "*. The system will read the file COREnn .. and warn when it is updated, although first it will ask for the name (to be shown in the list to choose from at startup and in the BIOS list).

The ZX Spectrum core update is exactly the same as other cores, but instead of the name CORE1.ZX2 or CORE1.ZXD, it has to be a file called SPECTRUM.ZX2 or SPECTRUM.ZXD.

# esxdos

To update esxdos to a new version, the distribution must be obtained from the official website.

Once downloaded and extracted, the contents of BIN and SYS directories have to be copied to the root of the card, merging the existing ones (to preserve the exclusive ZXDOS+ commands).

Copy ESXMMC.BIN in the root of the microSD card, renaming it as ESXDOS.ZX2 (for a ZXDOS+ with FPGA LX16 board) or ESXDOS.ZXD (for a ZXDOS+ with LX25 board).

Start ZXDOS + with the card inserted and press F2 to access BIOS setup. Select the Upgrade menu and choose *"Upgrade esxdos for ZX"*. In the dialog that appears choose *"SD file"* and, when it asks *"Load from SD"* answer *"Yes"* to the question *"Are you sure?"*. The content of the file ESXDOS··· will be read, written to the flash storage and you will be notified when it is updated.

Do a Hard-reset, or turn it off and on.

If everything has been done correctly, when you turn on the ZXDOS+ you will see how esxdos detects the card and loads the necessary components to work, showing the new version at the top.

# Other cores

## ZX Spectrum Next

ZX Spectrum Next is an FPGA based project, which wants to be the evolution of the Sinclair ZX Spectrum line of computers. It brings new features while keeping hardware and software compatibility with previous ZX Spectrum computers.

Specially thanks to avlixa, ther exists a ZX Spectrum Next core synthesized for ZXDOS+.

The core, for the moment does not havd any of these features:

- Raspberry Pi
- Internal beeper
- EDGE expansion Connector
- RTC module
- Membrane keyboard
- Flashing additional cores or upgrading the Next core from within the Next core
- MIC out
- HDMI Video
- UART communication using the joystick port

The user manual is available to download at the official web page.

### microSD card format

You have to use a microSD card with the first partition formatted as FAT16 or FAT32, and inside, the standard esxDOS distribution, matching ZXDOS+ BIOS version (see esxdos corresponding section for more info).

Download NextZXOS distribution from the official page.

Extract NextZXOS in the root of the microSD card, and then edit `config.ini` under `c:/machines/next` to include the line `ps2=0` if it doesn't exist or edit the existing line from 1 to 0. This effectively switches the dual PS/2 port to keyboard first as the Next Firmware (TBBLUE.FW) switches the primary input to mouse. Also edit the line `intbeep=0` to disable the internal beeper (this last step is not necesary on gomaDOS+).

If it wasn't already, install ZX Spectrum Next core into ZXDOS+.

## Keyboard

## Special keys and buttons

Take into account that `Ctrl+Alt+backspace` does not work with the ZX Spectrum Next core. You have to power cycle if you want to use another core. Also, there is no Reset or Drive button.

While the core is running:

- `F1`: Hard Reset
- `F2`: Scandoubler. Doubles the resolution. Should be of for SCART
- `F3`: Change vertical frequency between 50Hz and 60Hz
- `F5`: Soft Reset
- `F7`: Scanlines
- `F9`: NMI
- `F10`: divMMC NMI. Simulates Drive button. If used with Caps Shift it forces a rescan of drives and a reload of the boot screen under esxDOS

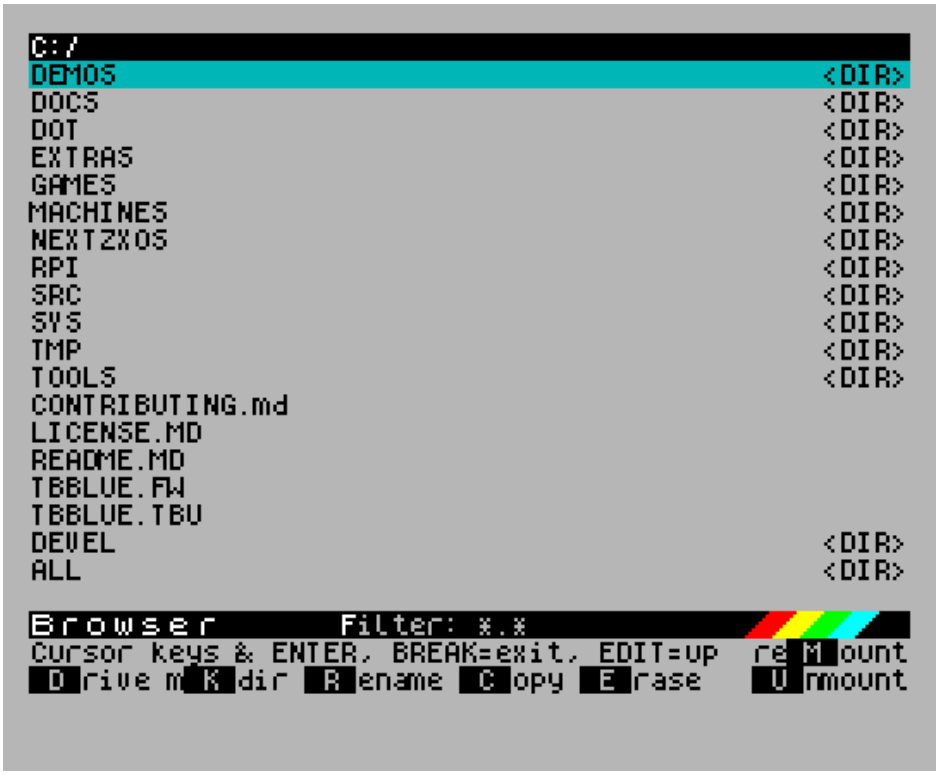
## Basic Guide

On first boot, some help screens will show up. After pressing Space key, NextZXOS Startup Menu appears.


You can navigate the menu with the cursor keys or a joystick (if configured as Kempston, MD or cursor). Enter or the joystick button chooses one element.

More⋯ shows a second menu with more options.

If you choose `Browser`, NextZXOS Browser will start, and then you can see the contents of the microSD card and load a file (TAP, NEX, DSK, SNA, SNX, Z80, Z8, etc.).




At the time of writing, the ZX Spectrum Next core for ZXDOS+ does not support the use of a Raspberry Pi-based accelerator, so it is not possible to load TZX files.

It is not possible to load TRD files from the Browser (NextZXOS must be configured to load a "personality" with esxdos).

For more information, see the official user manual.

# MSX

MSX1FPGA is a project to clone MSX1 in FPGA. The original development is by Fabio Belavenuto and is available at GitHub.

Some of its features are:

- MSX1 at 50Hz or 60Hz
- 128K Nextor (MSX-DOS2 evolution) ROM with SD driver
- Reconfigurable keyboard map
- Scanlines

## microSD format

You have to use a microSD card with the first partition in FAT16 format. You can also use a second FAT16 partition for MSX software, and leaving the firt one only for the system startup.

You need to get:

- Basic SD project files SD from GitHub
- Nextor driver (`NEXTOR.SYS`) and ROM (`NEXTOR.ROM`) also from GitHub
- MSX1 ROM (`MSX_INT.rom`, `MSX_JP.rom` or `MSX_USA.rom`) at the same repository

Copy the contents of the SD directory in the root of the first partition of the microSD.

Copy `NEXTOR.SYS` to the same place.

Copy `NEXTOR.ROM` inside the `MSX1FPGA` directory.

Copy one MSX1 ROM (`MSX_INT.rom`, `MSX_JP.rom` or `MSX_USA.rom`) inside the `MSX1FPGA` directory, but renaming it to `MSX1BIOS.ROM`.

The file `/MSX1FPGA/config.txt` keeps the core configuration, using this format:

```
11SP01
||||||
|||||+-Scanlines: 1=Enabled, 0=Disabled
||||+--Turbo: 1=Initialize with turbo enabled
|||+---Color System: N=NTSC, P=PAL
||+----Keymap: E=English, B=Brazilian, F=Francese, S=Spanish, J=Japanese
|+-----Scandoubler(VGA): 1=Enabled, 0=Disabled
+------Nextor: 1=Enabled, 0=Disabled
```

If it wasn't already, install MSX core into ZXDOS+.

## Keyboard

### Special keys and buttons

While running the core:

- `Print Scr`: Changes between VGA and RGB mode
- `Scroll Lock`: Enables or disables scanlines
- `Pause`: Changes between 50Hz and 60Hz
- `F11`: Enables and disables turbo mode
- `Ctrl+Alt+Supr`: Soft Reset
- `Ctrl+Alt+F12`: Hard Reset
- `Ctrl+Alt+Backspace`: Restarts the FPGA
- `Left ALT`: MSX GRAPH
- `Right ALT`: MSX CODE
- `Page Up`: MSX SELECT
- `Start` MSX HOME (`Shift+HOME`: CLS)
- `End`: MSX STOP
- `Ñ` or `Windows`: MSX DEAD

In BASIC use `CTRL+STOP` keys to stop the execution of a program.

To change the video mode between 50Hz and 60Hz (and thus play at correct speed PAL games), you can use `DISPLAY.COM`, which can be downloaded here.

## Basic Guide

To go to BASIC from MSX-DOS you must execute `BASIC` command.

To go to MSX-DOS from BASIC, execute `CALL SYSTEM`.

### MSXCTRL

An exclusive utility of MSX1FPGA core, which lets you control all the core options that were previously available only by editing the configuration file or with some key combination.

When running `MSXCTRL` all the use parameters are shown:

```
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` show help for a parameter. For example, `MSXCTRL -i` show the current configuration, `-t 1` sets turbo mode on, etc.

### Other

There are different ways to load games depending on the kind of file: .CAS, .DSK o ROM (see this ZX-Uno forums thread for more info).

The spanish keymap officially available can be replaced with a better one. See here for more information.

# Amstrad CPC

ZXDOS+ Amstrad CPC core is based on the FPGAmstrad project by Renaud Hélias.

Some of its features are:

- VGA: 640x480 VGA centered at 60Hz
- Disk selection: The first disk image detected is inserted on startup, and pressing a key makes a reset and loads the next one

## microSD card format

You have to use a microSD card with the first partition in FAT32 format, 4GB in size, and with 4096 bytes per cluster.

You also need the following ROM files (they are available at the original project Wiki) or from the GitHub repository: - `OS6128.ROM` - `BASIC1-1.ROM` - `AMSDOS.ROM` - `MAXAM.ROM`

It is also recommended to copy one or more disk image files (`DSK`) with the software that you want to run.

Copy all `ROM` and `DSK` files to the root directory of the FAT32 partition.

## Keyboard

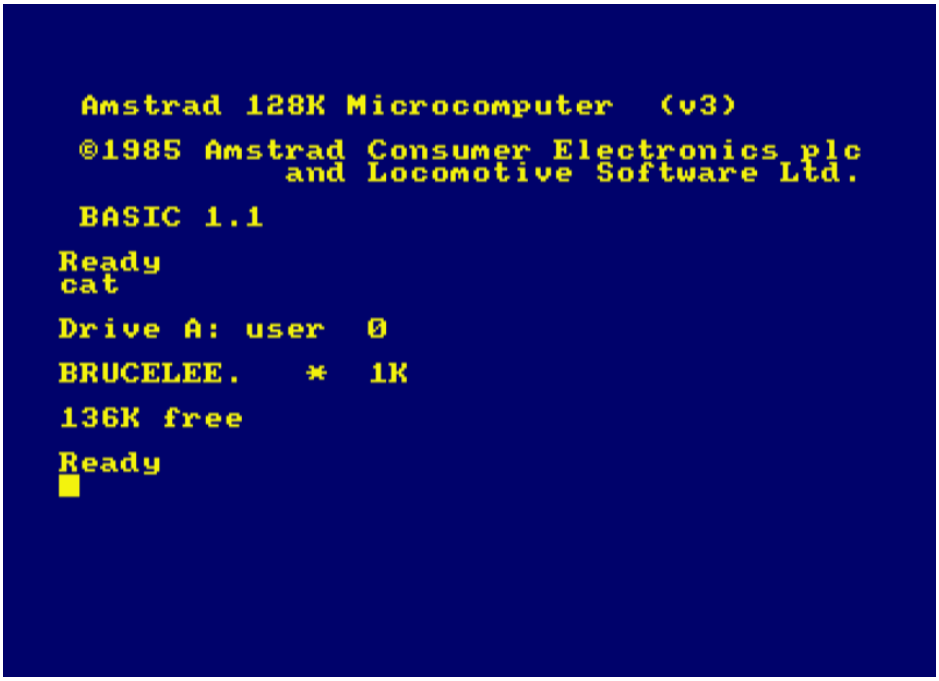
## Special keys and buttons

During core execution:

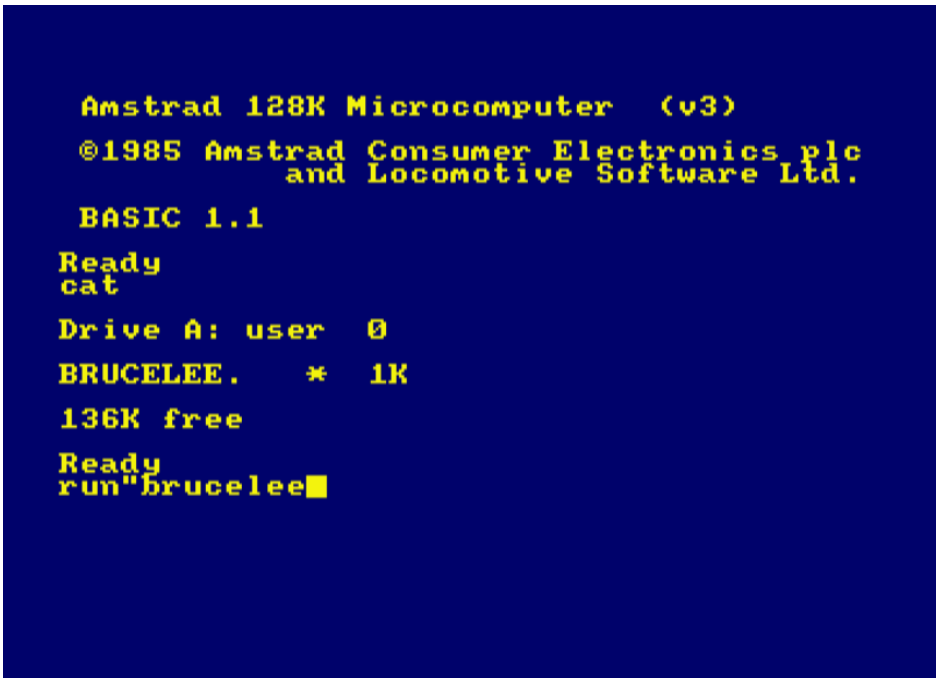
- `Page Up`: Reset the Amstrad computer and loads the next `DSK` file alphabetically
- Only the left shift key works properly

## Basic Guide

Use the `CAT` command to see the contents of the currently loaded DSK file.




Type the command `RUN"<name>` to load a program from disk




Press `Page Up` key to reset and load the next `DSK` file.

# Acorn Atom

Acorn Atom was a home computer made by Acorn Computers Ltd. The ZXDOS+ core (based on the ZX-Uno core made by Quest) is an adaptation of the AtomFPGA project. You can get more information at ZX-Uno Forums.

## microSD card format

You have to use a microSD card with the first partition in FAT16 format.

Download the latest version of Atom Software Archive from GitHub.

You can set up the files in the microSD in two different ways:

1. Extract all the contents of the archive to the root of the SD card. `SYS` directory contents are compatible with esxdos `SYS` directory, so you can merge both into one.
2. Have less files an directorios in the root directory. Create a directory named `ATOM` in the SD root, and copy inside all the uncompressed archive content, except for the directory `MANPAGES` which must also be in root. Then, extract and the files from `trick_ATOM_folder` archive (available at ZX-Uno Forum), replacing any file with the same name. You will get a file and directory structure like this:

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

## Keyboard

## Special keys and buttons

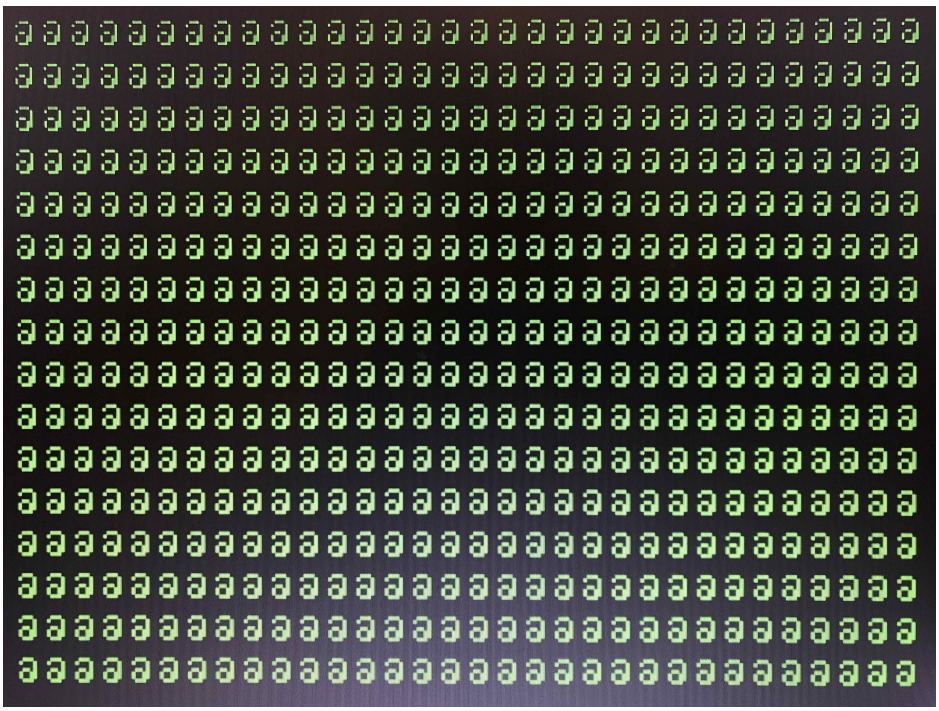
While the core is running:

- `Shift+F10`: Shows Atom Software Archive Menu
- `F10`: Soft Reset
- `F1`: Turbo mode 1Mhz
- `F2`: Turbo mode 2Mhz
- `F3`: Turbo mode 4Mhz
- `F4`: Turbo mode 8Mhz

The keyboard uses the following mapping:

## Basic Guide

Sometimes, after starting up the core, a screen full of `@` appears. Ejecting and inserting, or only inserting, the microSD card will fully start the system.



Once it's running, press `Shift+F10` to show a menu where you can choose and load Atom Software Archive programs from the card.

# Commodore 64

The Commodore 64 (C64, CBM 64/CBM64, C=64,C-64, VIC-641), was an 8-bit home computer manufactured by Commodore International.

The ZXDOS+ core is developed by Neuro.

## microSD card format

You can use a microSD card with the first partition formatted as FAT16 or FAT32. Disk image (`D64`) and tape (`TAP`) files can be loaded from the SD card.

See the corresponding section for instructions of how to install the Commodore 64 core in ZXDOS+.

## Keyboard

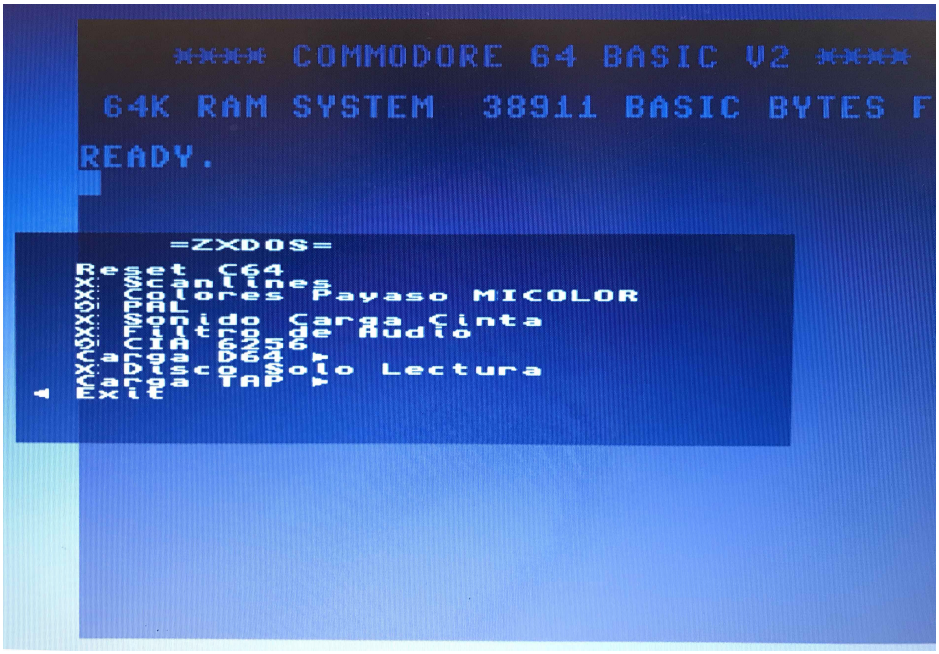
## Special keys and buttons

While the core is running:

- `F12` (`Caps Shift+Symbol Shift+W` on gomaDOS+): Shows options menu
- `Scroll Lock` (`Caps Shift+Symbol Shift+G` on gomaDOS+): switches between VGA and RGB modes
- `Esc`: RUN/STOP (`Shift+RUN/STOP`: Load from tape)

## Basic Guide

After pressing F12, the option menu is shown.




The menu offers the following options

- Core reset
- Enable o disable scanlines
- Change color palette (Colores Payaso MICOLOR)
- Enable or disable PAL mode
- Enable or disable tape loading sound (Sonido Carga Cinta)
- Enable or disable audio filter (Filtro de Audio)
- Load D64 file from SD (Carga D64)
- Load TAP file (Carga Tap)

After a disk is inserted, normally, you can use `LOAD "*",8,1` and press `Enter` to load the software inside. Once `READY` is shown on screen, type `RUN` and press `Enter` to execute it.

If there was more than one program in the disk, type `LOAD "$"` and press `Enter`. Then, type `LIST`, and press `Enter`, to see a list with the files in the disk. Now, to load one of them, type `LOAD "<name>",8` (where `<name>` is the name of the file to load) and press `Enter`. Once `READY` is shown on screen, type `RUN` and press `Enter` to execute it. If this didn't work try again with the command `LOAD "<name>",8,1`.

To load from tape, you can type `LOAD` and press `Enter`, or just press `Shift+Esc` (`Shift+RUN/STOP`).

# Phoenix

Space-Themed shooter video game released in arcades by Amstar Electronics.

Some of the features of the ZXDOS+ core are:

- Two different video modes: RGB/PAL60Hz and VGA 60Hz
- Scanlines on VGA mode
- Controls can be optionally rotated 90º

## microSD format

This core does not use the microSD card.

## Keyboard

## Special keys and buttons

While the core is running:

- `Q` and `A` or `Left Cursor` and `Right Cursor` (or a joystick): Movement control
- `Z` or `X` `Left Windows Key` and `Space` (or joystick buttons 1 and 2): Fire 1 and 2, also to insert coin and `Start`
- `F2`: Switches between VGA and RGB modes
- `-` (numeric keyboard): Enable or disable scanlines
- `Tab`: Enables or disables 90º rotation of the direction of controls

## Basic Guide

By default, the core starts with normal controls, configured for vertical displays. If you have an horizontal display, the image will be rotated. To ease the control, and make it more natural and according to what you see, when typing `Tab`, up-down directions are switched with left-right. This is both for joystick and keyboard controls.

# Pong

Pong was one of the earliest arcade video games manufactured by Atari.

Some features of this core are:

- Two different video modes: RGB/PAL60Hz and VGA 60Hz
- 7 game variants
- Support for 2 or 4 players
- Support for Joysticks

## microSD format

This core does not use the microSD card.

## Keyboard

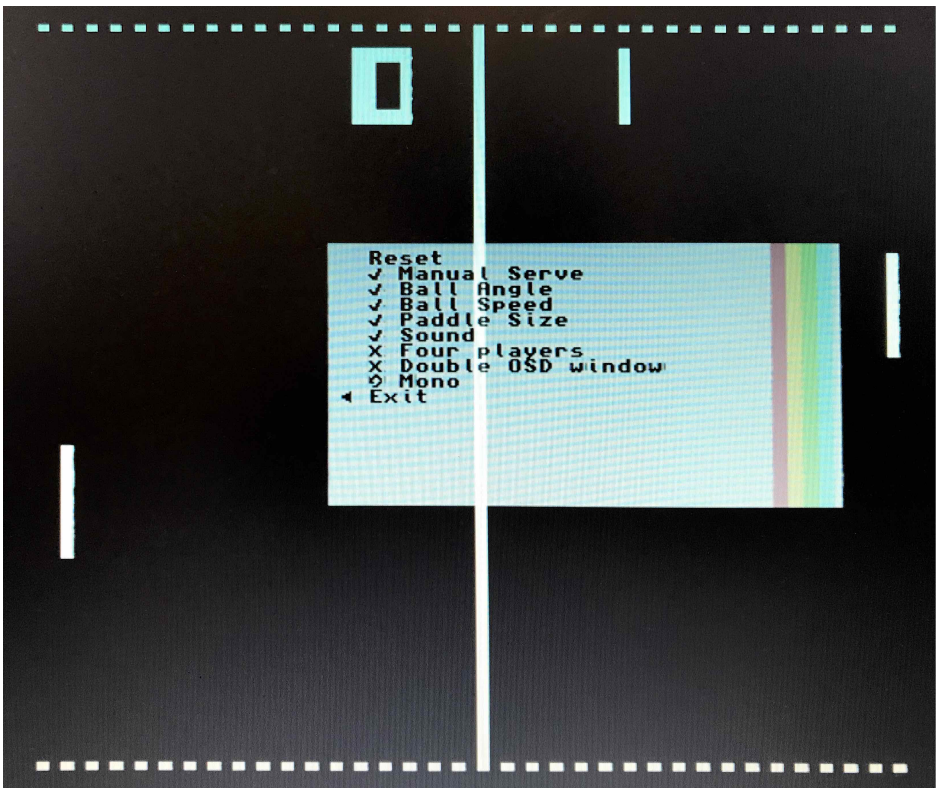
## Special keys and buttons

While the core is running:

- `Esc` or joystick button 2 (or `Caps Shift+Space` on gomaDOS+): Show or hide configuration menu
- `Ctrl+Alt+Backspace` (`Caps Shift+Symbol Shift+F` and `G`): Hard reset
- `Scroll Lock`: switch between VGA and RGB mode
- `F3` o `F12`: Restart game
- Number betwee `1` and `7`: Change the game variant
- `Cursor up` and `Cursor down`: Control left pad (Player 1 in 2 player mode and player 3 in 4 player mode)
- Joystick 1: Control left pad (Player 1)
- `Q` y `A`: Control right pad (Player 2 in 2 player mode and player 4 in 4 player mode)
- Joystick 2: Control right pad (Player 2).
- `Z`, `M` or joystick button 1: Manual serve

## Basic Guide

Pressing `Esc` or joystick button 2 (`Caps Shift+Space` on gomaDOS+) shows or hides the configuration menu.




The following options are available:

- Manual Serve
- Ball Angle
- Ball Speed
- Paddle Size
- Sound
- Four players
- Double OSD Window size
- Exit

# NES

Nintendo Entertainment System (also known as Nintendo NES or just NES) is the second home video game console produced by Nintendo.

The ZXDOS+ core has been made by Nihirash, based on the previous version for ZX-Uno by DistWave y Quest.

Some features of this core are:

- HQ2X filters that "removes pixels" from the image
- Scanlines simulation
- Made with NES NTSC clock timings, so only USA ROMs run fine. PAL ROMs run faster than they sould
- You can load ROMs from the SD
- You need, at least, one gamepad or joystick connected, and it must have several buttons
- Only VGA video mode is supported, with non-accurate timings, so it may not work with some displays

## microSD card format

You need a microSD card with the first partition in FAT16 format to store ROM image files of the games to load. ROM files can be inside subdirectories.

See the corresponding section for instructions of how to install the NES core in ZXDOS+.

## Keyboard

## Special keys and buttons

While the core is running:

- `Esc` or joystick button 2 (or `Caps Shift+Space` on gomaDOS+): Show or hide configuration menu
- `Ctrl+Alt+Backspace` (`Caps Shift+Symbol Shift+F` and `G`): Hard reset

## Basic Guide

Pressing `Esc` or joystick button 2 (`Caps Shift+Space` on gomaDOS+) shows or hides the configuration menu.


The following options are available:

- Reset NES
- Scanlines
- HQ2X Filter
- P1 Select
- P1 Start
- Load ROM
- Exit

# Troubleshooting

## Firmware recovery

Sometimes (e.g. when installing an experimental core or when upgrading the ZX Spectrum Next or the BIOS) it may happen that the ZXDOS+ stops booting. The board LEDs are on, but there is no display, and it doesn't do anything when trying the different key combinations to access BIOS setup, etc.

When this happens, there are several recovery methods that let you install again the firmware.

### Recovery using a Raspberry Pi

**Hardware required**:

- Raspberry Pi (with SD card, keyboard, display, power supply, etc.) and with internet connection
- 5 jump wires (if possible, female on both sides)
- One hex key with the right socket size for ZXDOS+ cover screws
- microSD for ZXDOS+ with the first partitiion formatted as FAT16 or FAT32
- Keyboard and display for ZXDOS+

**Software required**:

- Flash image and recovery file for ZXDOS+ (LX25), from the official repository

**Instruction Steps**:

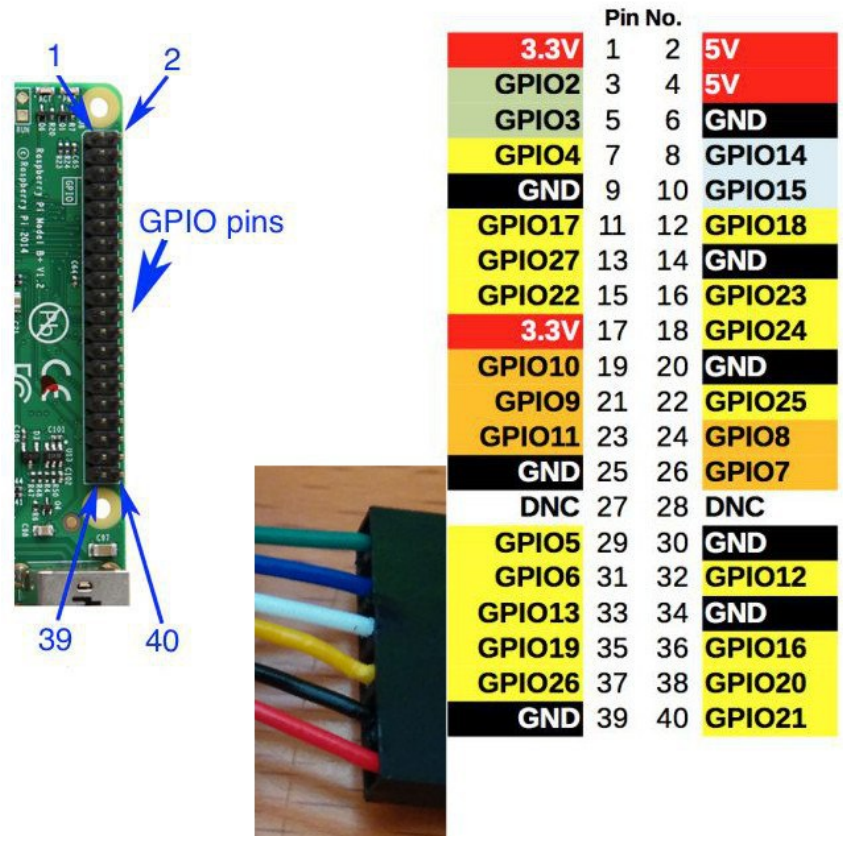
1. Install Raspberry Pi OS (formely known as Raspbian) to the Raspberry Pi SD card (using the official download, NOOBS, PINN, etc.)
2. Install Open OCD:

```
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
git clone git://git.code.sf.net/p/openocd/code openocd-code
cd openocd-code/
./bootstrap
./configure --enable-sysfsgpio --enable-bcm2835gpio
make
sudo make install
cd ..
rm -rf ./openocd-code
```

3. Open the ZXDOS+ case and connect the FPGA JTAG lines (TMS, TDI, TDO, TCK y GND), using the wires, to the Raspberry Pi GPIO pins.


**DO NOT** connect the 3V line

Take note of the chosen pins, making sure that GND is connected with GND.




In this example, the 31, 33, 35, 37 and 39 pins will be used (corresponding to GPIO #6, GPIO #13, GPIO #19, GPIO #26 and GND), like this:

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

4. Copy to the Raspberry Pi the file named recovery.zxd.bit previously downloaded from the official repository. For our example, it will be at /home/pi/zxdosplus/unbrick/
5. Make a copy of Open OCD configuration file, to the same directory where recovery.zxd.bit is.

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

6. Edit `raspberrypi2-native.cfg` copy, updating `bcm2835gpio_jtag_nums` (uncommenting, if necessary), with your JTAG and GPIO connection numbers, at the line `bcm2835gpio_jtag_nums`. For our example:

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

7. Comment, if it wasn't already, the line `bcm2835gpio_swd_nums`:

```
#bcm2835gpio_swd_nums 11 25
```

8. Add, to the end of the file, the line `adapter speed 250`:

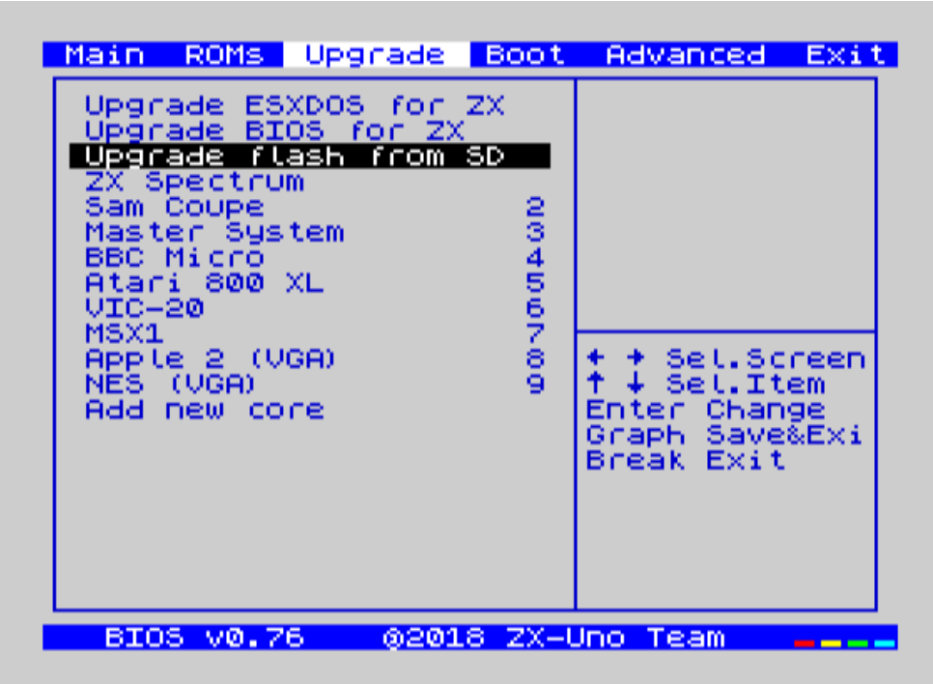
```
adapter speed 250
```

9. Start ZXDOS+
10. Make sure that, on the Raspberry Pi, we are in the directory where `recovery.zxd.bit` is, and execute the command that loads the BIOS on recovery mode, using the path to the previously edited `raspberrypi2-native.cfg`.

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

If all goes well, we will see that the FPGA LED change their state and the BIOS is shown on the display.

If there is no image on the display, press `Scroll Lock` (`Caps Shift+Symbol Shift+G` on gomaDOS+): to switch between RGB and VGA modes, just in case the recovery BIOS did start in the wrong mode for our setup.




11. Insert in the ZXDOS+ the microSD card formatted as FAT16 o FAT32, and with the `FLASH.ZXD` file downloaded previously.

12. Select the option `Upgrade Flash from SD`. Press Enter, choose `Yes`, and press Enter again to start the Flash writing process.

This will erase all the previously installed cores and ZX Spectrum ROMs.


After some minutes, the process will end, and, after turning the ZXDOS+ off and on, it should start fine.

If no image is shown, press again `Scroll Lock` (`Caps Shift+Symbol Shift+G` on gomaDOS+): to switch between RGB and VGA modes. In this case, you should have to enter the BIOS and change the right advanced setting that matches your display.

# References


ZX-Uno

ZX-Uno FAQ

Guía rápida del ZX-Uno

Core ZX Spectrum

Layouts de teclado

Nuevo firmware de teclado ZX-GO+

Almost (In-) Complete List of esxDOS DOT-Commands

WiFi (RetroWiki)

WiFi en ZX-Uno

Core de ZX-Uno Test UART (WiFi)

Network tools for ZX-Uno pack

ESP8266 AT Instruction Set

Core ZXNEXT en ZXDOS

ZX Spectrum Next en ZXDOS

Core MSX

MSX1FPGA

MSX Pack

Nextor para MSX

Nextor User Manual

MSX-DOS

Atom Software Archive en carpeta ATOM

Teclado Core Atom

Programming a Spartan 6 with a Raspberry Pi

Tutorial para desbriquear el ZX-Uno con una Raspberry