

ZXDOS+ and gomaDOS+ Manual

kounch

Version 1.0.0

Index

Introduction	1
Initial Setup	1
microSD card formatting	2
Windows	2
MacOS	2
Linux	3
esxdos	4
BIOS	6
Main	6
ROMs	7
Upgrade	7
Boot	8
Advanced	9
Exit	10
ZX Spectrum	11
Keyboard	12
Spanish	12
English	12
Spectrum	12
Special keys and buttons	13
ROMs	14
esxdos	15
Basic Guide	15
ZXDOS+ Commands	17
Upgrade	18
BIOS	18
ROMs	18
Cores	18
esxdos	19
References	20

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 formatted 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 Management 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`, `cdisk`, `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.

A screenshot of a ZX Spectrum screen showing the esxdos boot process. The screen has a black background with white text. At the top left is a stylized logo that looks like 'ZXDOS'. To its right, the text 'v0.8.6-DivMMC', '© 2005-2018', and 'Papaya Design' is displayed. The boot sequence follows: 'Detecting Devices...', 'sda: card', 'Mounting drives...', 'hd0: ZX , FAT16, 128M', and then four lines of 'Loading' followed by file names and '[OK]' status: 'Loading ESXDOS.SYS... [OK]', 'Loading RTC.SYS... [OK]', 'Loading NMI.SYS... [OK]', and 'Loading BETADISK.SYS... [OK]'.

```
ZXDOS v0.8.6-DivMMC
© 2005-2018
Papaya Design

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]
```

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
```

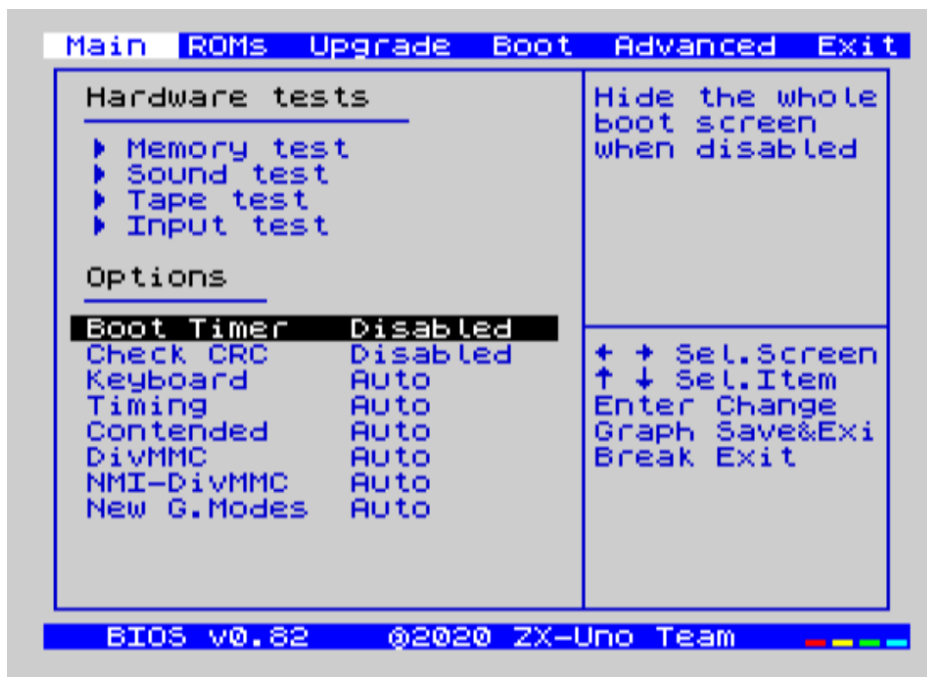
<<#_zxdos+_commands,It is explained later> what each of them does.

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

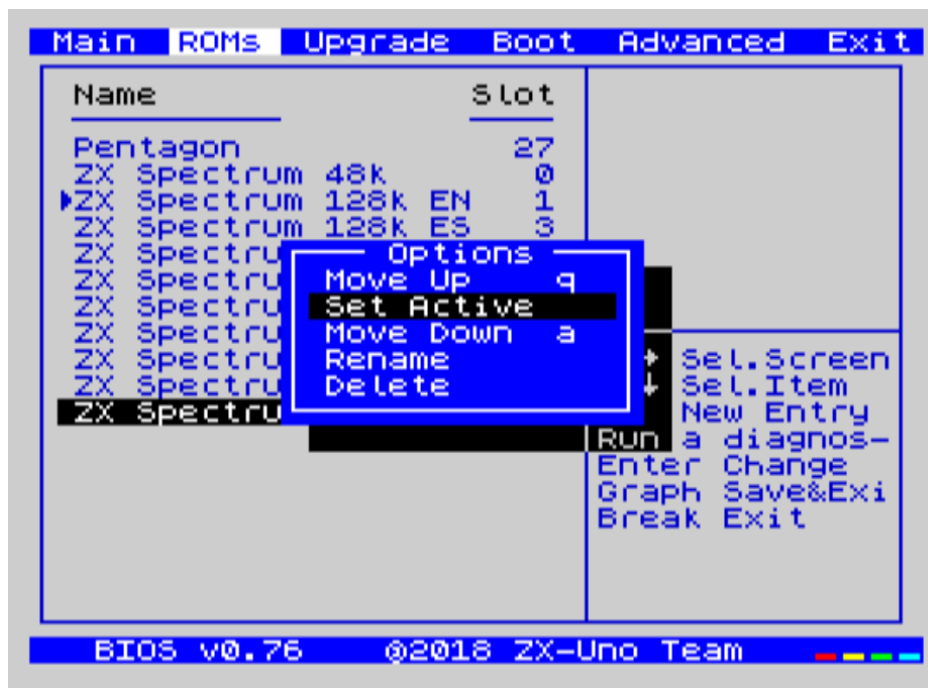


In the first configuration screen, in addition to being able to run several tests, you can define the default behavior for the following:

- 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)

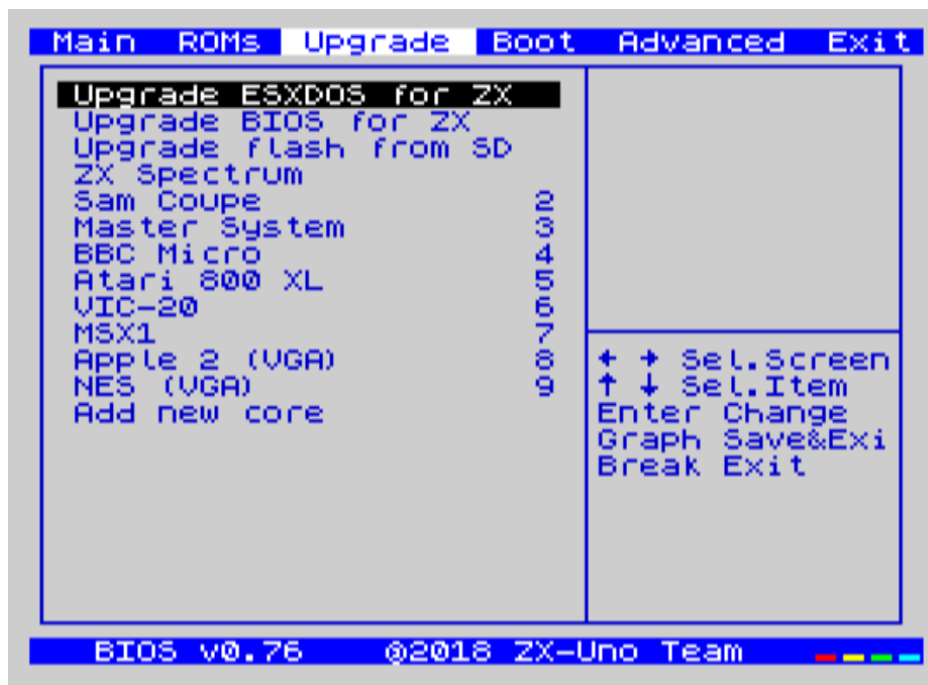
More technical information can be found on [de ZX-Uno Wiki](#).

ROMs



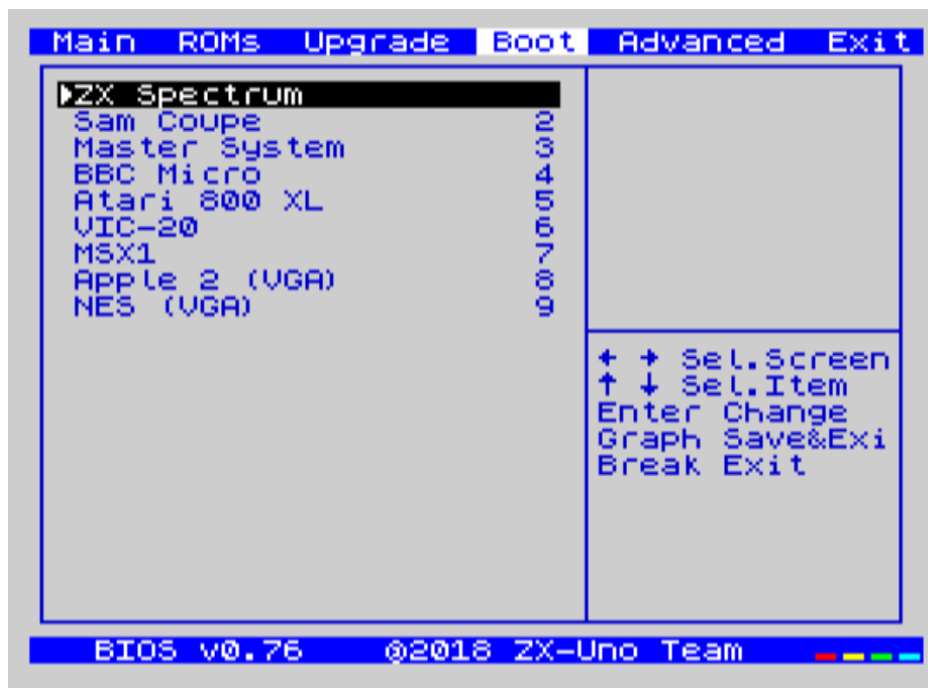
The second screen shows the installed ZX Spectrum ROMs. You can reorder (Move Up, Move Down), rename or delete each of them, as well as choose the one that will be loaded by default at startup (Set Active).

Upgrade



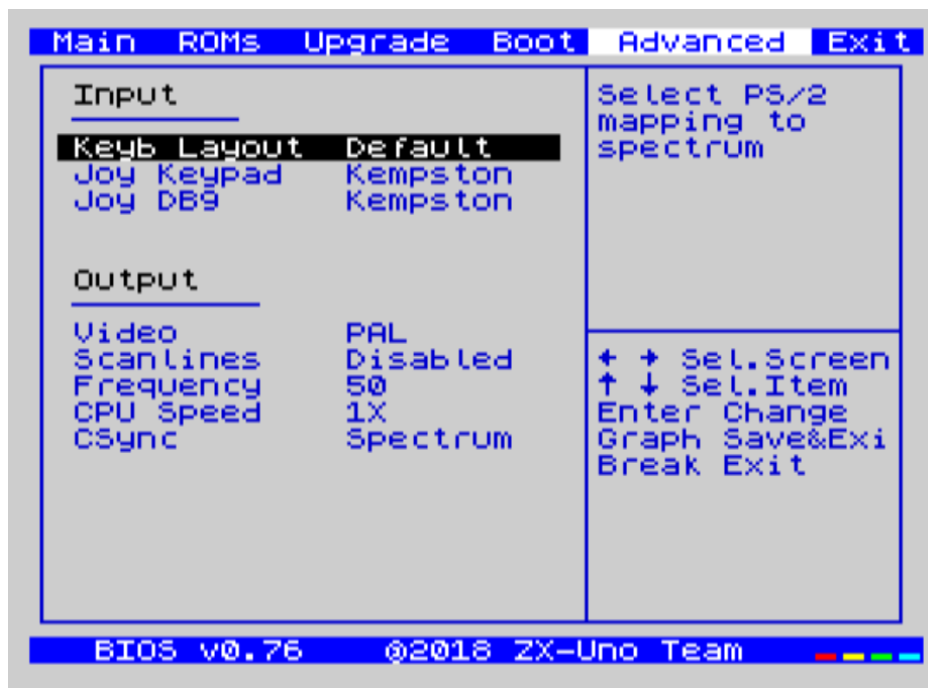
Upgrade screen is used to perform the different updates of the Flash memory content: esxdos, BIOS, Cores, etc. (see [the section corresponding to updates](#) for more information).

Boot



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

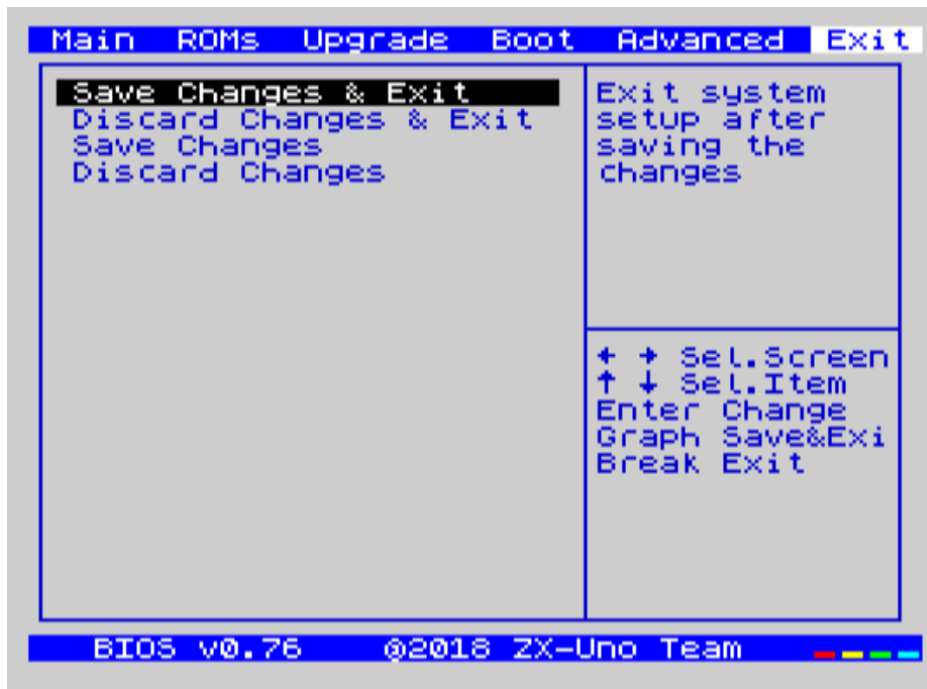
Advanced



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`.

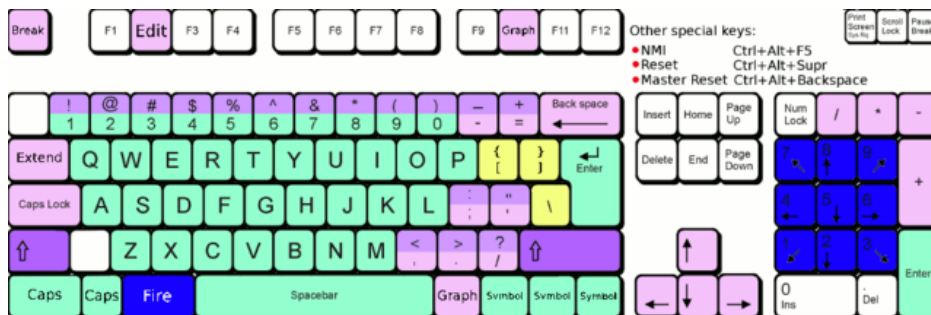
For the map to be preserved after a master reset, it has to be selected as **Default** in the BIOS.

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** (**NMI** en gomaDOS+): 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 (beginning 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.
- **Bloq. Despl.** (**Caps Shift+Symbol Shift+G** on gomaDOS+): Switches between composite and VGA video modes.

During startup:

- **F2** (**Caps Shift+1** on gomaDOS+, Full Mode) Enter BIOS setup
- **Bloq. Mayús** 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 of 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
- **snapshot**: 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 load a file from a drive (BASIC Program, SCREEN, CODE, etc.)
- **SAVE** to save data in a file
- **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 update 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)
- **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 content 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

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/\[ZX1RomPack\]](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 where 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 as the first).

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.

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 \(WiFi\)](#)

[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](#)