

Introduction:

CP/M Box is an emulator of the Amstrad PCW computer architecture, that is: 8256, 8512, 9512, 9256, 9512+ and PcW10. It is also capable of emulating additional hardware, different options and configurations, etc.

It does not emulate the PcW16, since it belongs to a different architecture; it has its own emulator. The current version of the emulator is 1.11.1, to which this manual refers.

The objective is to achieve an emulation as faithful as possible to the original machine.

Inhaltsverzeichnis

Introduction:.....	1
Features:.....	2
Requirements:.....	2
Legal stuff:.....	3
Installation and use:.....	3
Installation:.....	3
Snapshots:.....	3
Actions:.....	3
Disks:.....	3
Videos:.....	4
Keyboard:.....	4
Joystick:.....	4
Mouse:.....	4
Light pen:.....	4
Mass storage:.....	4
Interface:.....	5
Title:.....	5
Menus:.....	5
Bar:.....	6
Screen:.....	7
Hardware:.....	7
Emulation:.....	8
Volume control:.....	9
Advanced use:.....	10
Debugger:.....	10
Registers:.....	11
Disassembly:.....	11
Breakpoints:.....	11
Others:.....	12
Hardware registers:.....	13
Memory:.....	13
PCW Format:.....	13
Macros:.....	14
Notes:.....	15
F.A.Q.:.....	15
More information:.....	16

Features:

- Everything is included inside one small executable file.
- Emulation of official models: 8256, 8512, 9512, 9256, 9512+, PcW10.
- Emulation of different memory sizes (from 128KB to 2MB), floppy drives (CF2, CF2DD, MF2DD) including geometric issues, video standards (PAL / NTSC), printer controllers, ...
- Emulation of additional hardware:
 - Integrated centronics port (PcW10).
 - Sprinter (8 Mhz).
 - Joysticks: Cascade (JoyceStick), Kempton, Spectravideo, Dk'tronics, Electric Studio.
 - Mice: AMX, Kempston, KeyMouse, Electric Studio.
 - Light pens: Trojan, Electric Studio.
 - CPS8256.
 - Storage: Cirtech GEM, Cirtech FlashDisk, uIDE.
 - Audio: Dk'tronics sound interface, EMR MIDI Interface.
 - Others: ColorIn.
- Understands the following formats:
 - Snapshot: PCW.
 - Floppy: DSK, IMG.
 - Actions: ACC.
 - Screen: PNG.
 - Video: WMV.
 - Storage: HDF, RAW / BIN / IMG.
- The interface allows drag and drop.
- A fast mode to manually accelerate emulation when needed.
- Audio and video options, including filters. Supports full screen.
- Automatic fast mode on disc load.
- Can use PC joysticks / pads.
- Redefinible keyboard.
- A debugger with breakpoints at fetch, read, write, input and output, allowing ranges and patters.
- Possibility of mapping a PC folder over unit M under CP/M.
- Exact emulation up to the cycle, including contention and video.
- Macros / commands support.
- Audio system with separate mixer for each voice.

Requirements:

- Windows 7 64b.
- An audio device.

Legal stuff:

1. I do not assume any responsibility.
2. You can do whatever you want with this program.
3. All product names, logos, brands, trademarks and registered trademarks are property of their respective owners.

Installation and use:

Installation:

This emulator does not need any kind of installation, simply unzip the contents of the ZIP file in a folder and proceed to execute it.

Additionally, this program supports the automation of commands through the command line, using the same syntax as the macros, which are explained in its corresponding section. It can be used to associate file extensions, using it with external launchers, etc.

Snapshots:

A snapshot file is a complete state of the machine (including memory, hardware status, disk content, etc.) along with metadata about the emulated machine.

When dragging a file on the window it would be loaded without further ado. If during the release on the window the shift key was pressed, the emulation would be activated if it were deactivated.

Actions:

An actions file is a snapshot together with additional data that describe the events that happened after that moment. It is useful to save tutorials or games in a more compact way than a video.

When dragging a file on the window it would be loaded without further ado. If during the release on the window the shift key was pressed, the emulation would be activated if it were deactivated.

Disks:

A disk file contains the user data of a physical diskette, along with some of its geometric information.

The emulator is capable of loading both CPCEmu DSK files (normal and extended, with or without protection) and raw images with IMG extension.

The emulator works with disk images in memory, with the option to dump them back when extracted. You also have the option to insert them write protected.

When dragging a file on the window it would be inserted on the unit A. If during the release on the window the control key was pressed, it would be inserted in the unit B; if the shift key was pressed, the emulator would be reset and the emulation would be activated if it were disabled (interesting option for boot disks).

Additionally, the emulator allows to accelerate emulation during disk loads.

Videos:

This emulator allows video recording, recording exactly what is shown on the emulated screen together with the sound.

This option requires Windows 7 or latter, and a computer capable of video realtime compression.

Keyboard:

The emulator maps each one of the PCW keys to the one closest to the keyboard of the PC. They can be changed by the user in the emulation options, as well as consult their assignment.

Also, the cursors of the PC along with the numeric * and / can emulate a joystick in case of not having one connected to the PC.

Joystick:

The emulator recognizes any joystick / pad that Windows recognizes, by directly mapping the X, Y axis and the first two buttons to the emulated joystick (Cascade, Kempston, Spectravideo, DK'tronics, Electric Studio), and converting the signal to digital in case of necessity.

If the keypad option is enabled, the joystick will additionally emulate the arrows on the PCW numeric keypad with the axes, and the space and tab with the buttons.

Mouse:

The emulator is capable of emulating a PCW mouse (AMX, Kempston, KeyMouse, Electric Studio). To be able to handle the pointer in the emulated PCW the pointer must be captured, which is done by clicking with the right button on the emulated screen. To release it, just press the escape key.

Light pen:

The emulator is capable of emulating a PCW stylus (Trojan, Electric Studio) using the mouse. The instructions are the same as in the previous case.

Additionally there is an option to show a special cursor on the screen to see its current position.

Mass storage:

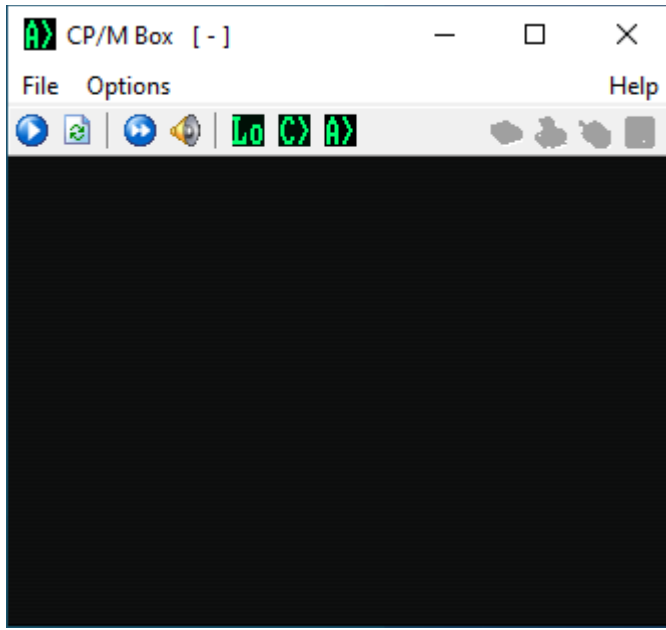
The emulator is capable of emulating several mass storage devices (Gem, Flash Disk, uIde). You can use both raw images and HDF files that include geometric information of the disk.

Given the size of these files, its information is not stored in the snapshots, but a link to the file that contains it.

Additionally, the emulator allows you to map a directory over the virtual M drive under CP/M.

Interface:

While windowed, the following elements are available:



Title:

At the top of the window, like in any other application, there is the title bar.

It has the title of the application and also the file name of the floppy inserted at the A drive.

Finally, it also has the typical icons for minimize, maximize and close the window, with the usual behaviour with one exception: maximize icon activates the full screen mode, which can be canceled pressing Escape.

Menus:

- File
 - Snapshot
 - Open - Opens a snapshot.
 - Save - Saves a snapshot.
 - Actions
 - Play - Starts playing actions.
 - Record - Starts recording actions.
 - Stop - Stops the recording or playback of the actions.
 - Capture - Saves a screenshot.
- Video
 - Record - Starts video recording.

- Stop - Stops video recording.
- Drive A / B
 - Insert - Inserts a disk in the drive.
 - Extract - Removes disk from the drive.
 - New - Creates a new unformatted disk and insert it.
 - RO - Reads only disk.
 - Dump - Dumps back the disk file if it was modified.
- Exit - Closes the emulator.
- Options
 - Hardware - Open the hardware options. It has its own section in the manual.
 - Emulation - Open the options on the emulator. It has its own section in the manual.
 - Languages - It has a submenu to choose one of the supported languages, and an auto-detection option.
 - Debugger - Open the debugger. It has its own section in the manual.
 - Persistence - Screen phosphor persistence emulation level. It has 4 levels, from less to more effect.
 - Speed - Emulation speed, relative to real machine.
 - Scale - Predefined window sizes.
 - Start / Stop - Toggles the emulation on / off.
 - Fast mode - Toggles between normal and maximum speed.
 - Button - Some interfaces have a button, appears when that hardware is selected. Its effect depends on the interface.
 - Reset - Restart the emulated machine, like pressing a reset button.
 - Hard reset - Restart the emulated machine from zero, like power it off then on.
- Help
 - Manual - Open this manual in the web browser.
 - About - Displays information about the emulator.

Bar:

Under the menus we have the bar, which is divided into three sections:

- Shortcuts - Shortcuts of different actions present in the menus. These are, from left to right:
 - Start / Stop.
 - Reset.
 - Fast mode.
 - Audio volume.
- Macros - Located after the shortcuts, this section has all the macros defined for the emulator. By default, there is only one (represented by "A>"), which inserts a file called cpm.dsk, resets and starts the emulation. They have their own section in the manual.
- Status - Aligned to the right, this section shows us visually the status of different hardware elements. From left to right we have:
 - Mass storage.
 - Printer.

- Mouse (captured).
- Disk.

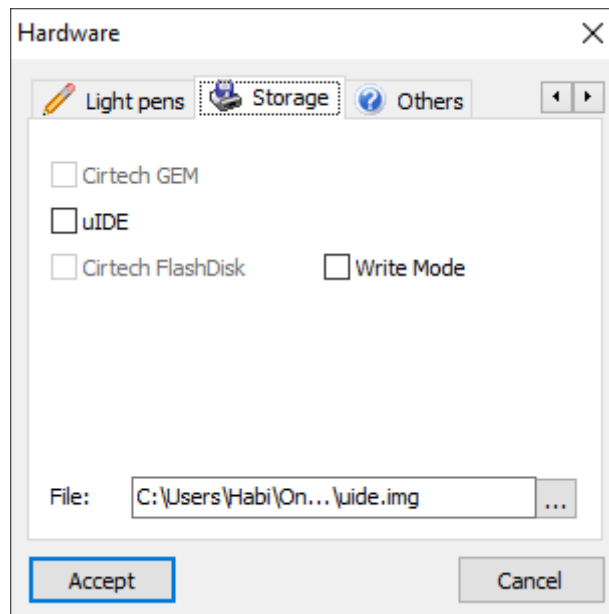
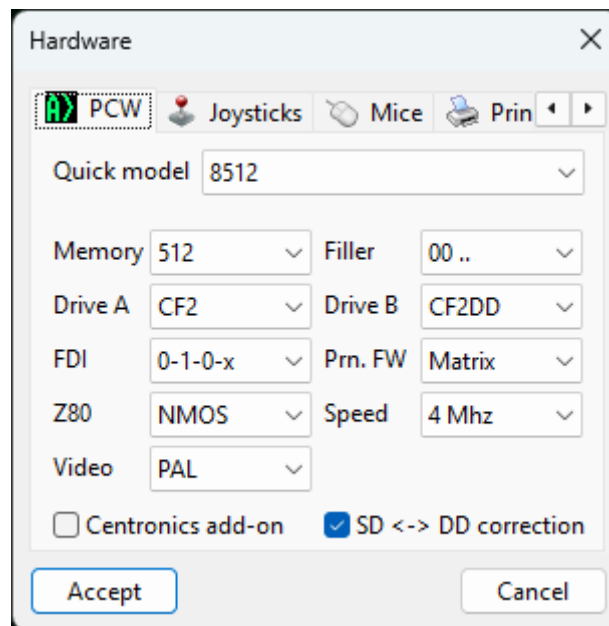
Screen:

This section represents the emulated computer screen, with its selected visual options.

It allows drag and drop of snapshot, actions and discs files over it.

Hardware:

In this window you can choose the hardware options to emulate. The acceptance of them will cause a reset of the emulation. The options are divided into several tabs:

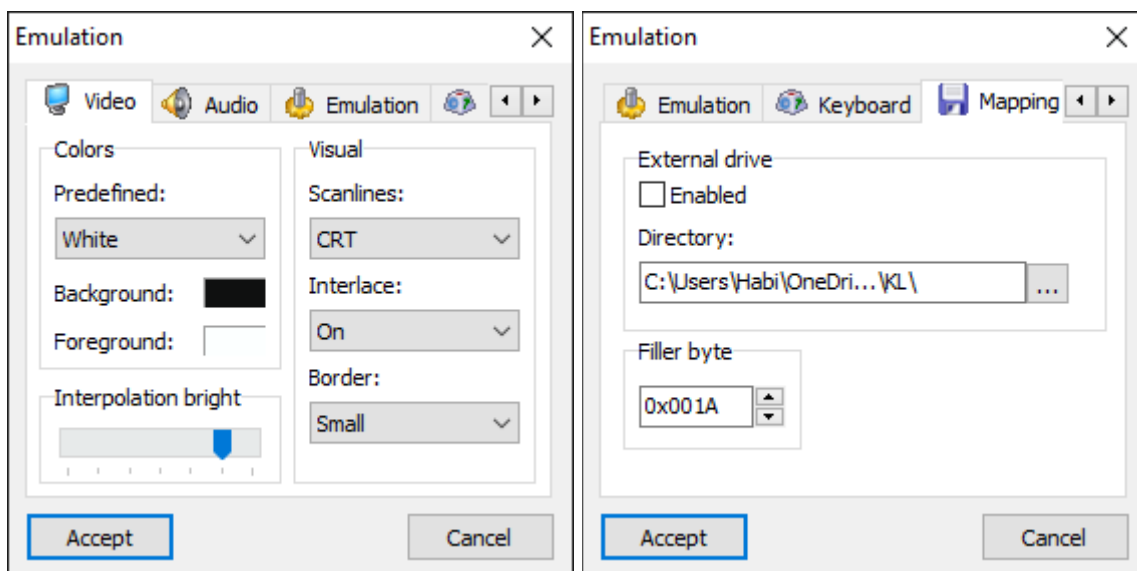


- PCW - Define the base model to emulate
 - Quick model - Sets the rest of the options to correspond to the chosen model.
 - Memory - Amount of memory, 128KB - 2MB.

- Filler - Initial memory values, depending on the integrated circuits used.
- Drive A / B - Type of disk drive in this unit.
- FDI - Interconnection of units / ready.
- Prn. FW - Type of printer controller (matrix or daisy wheel).
- Z80 - NMOS and CMOS variants.
- Speed - Normal (4Mhz) or Sprinter (8Mhz). The Sprinter emulation is preliminary.
- Video - Video standard (PAL or NTSC).
- Centronics add-on - PcW10 centronics interface.
- SD / DD correction - Correct disk geometry when the type of disk does not match that of the unit.
- Joysticks - Cascade (JoyceStick), Kempston, Spectravideo (RAM), DK'tronics, Electric Studio. If you select DK'tronics, the sound interface will also be emulated.
- Mice - AMX, Kempston, KeyMouse, Electric Studio.
- Printers - Standard (matrix or daisy wheel), CPS8256 serial, CPS8256 parallel.
- Light pens - Trojan, Electric Studio.
- Storage - GEM, uIDE, FlashDisk. The FlashDisk one includes a write mode option.
- Others - CPS8256, PCW 8256 Test, PCW 9512 Test, ColorIn (interface for coloring the PCW video) and EMR MIDI Interface. Diagnostic PCBs include an option to exchange ROMs.

Emulation:

In this window you can choose the options of the emulator. The acceptance of them will not cause a reset. The options are divided into several tabs:

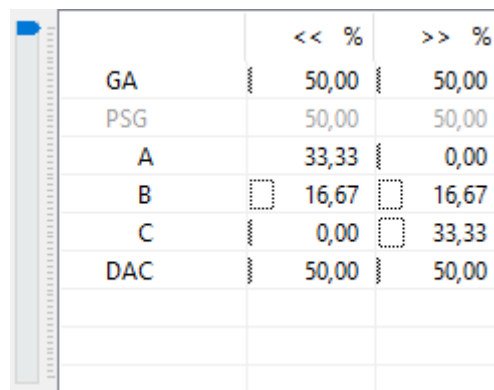


- Video
 - Colors - By clicking on the background and foreground colors you can change the visual aspect of the emulator. There are also predefined colors corresponding to green and white phosphor screens.
 - CRT effect - Displays defined or horizontally interpolated pixels in a similar way to a monochrome monitor.

- Interlaced - Displays defined or interpolated pixels vertically. Its intensity depends on the Interpolation bright bar.
- Border - Overscan area to be displayed.
- Window - Fixed aspect ratio option. Also, the application theme.
- Quality - Alternative screen dump, slower. Only useful for reductions.
- Others
 - Drives - Play disk sounds when accessed.
 - Low-pass filter - Applies a low-pass filter, which eliminates high frequencies.
 - Mouse - Sensibility of mouse movement, and also cursor autohidding.
 - Light pen - Option to show or not a special cursor when it is emulated.
 - Joypad - Allows the choice of the joystick / gamepad used. There is an option to emulate keypad (keystrokes).
 - Drive - Allows you to choose whether the disk indicator lights up at motor signal or command execution. There is also the option of fast loading, which accelerates the emulator while motor signal is active.
- Keyboard - Option to redefining the keyboard. By selecting a row we can choose the corresponding PC key from the drop-down list.
- Mapping - Mapping options for the virtual unit M of CP/M, and MIDI I/O. The fill byte is also specified (CP/M works on blocks of 128 bytes).

Volume control:

The volume control allows you to adjust the overall volume of the emulator, as well as the individual/stereo mix of each audio source.



	<< %	>> %
GA	50,00	50,00
PSG	50,00	50,00
A	33,33	0,00
B	16,67	16,67
C	0,00	33,33
DAC	50,00	50,00

To change a value, simply click on it. By default, each device adds up to 50% to each channel. The cells also display a small VU meter.

It's okay to exceed 100% in total: in addition to DC blocking for each source, there is dynamic range compression.

The only exception is MIDI which, being external, depends on each individual case. In the case of Microsoft's software synthesizer, only a global volume can be adjusted using the left channel.

Advanced use:

Debugger:

The debugger is a floating, resizeable window, and always on top of the emulator. As long as it is open, the emulator works in debugging mode.

When executing code, it acquires some transparency to indicate unavailability and to be able to see what it is underneath. Execution can always be stopped with the Escape key.

When not executing code, the current video position will be marked on the PCW screen with two red lines.

The window consists of several sections:

The debugger interface is divided into several sections:

- Registers:** Shows the state of various registers. AF is 0000, BC is 0000, DE is 0000, HL is 0000, IX is 0000, PC is 0000, and SP is FFFF. Other registers like IR, Iff 1, and Iff 2 are also visible.
- Disassembly:** A table showing the current instruction being executed. The instruction at address 0000 is 'DI'. Other instructions include 'XOR A', 'LD DE, FFFF', 'JP 11CB', 'LD HL, (5...', 'LD (5C5F)...', 'JR 0053', 'JP 15F2', 'RST 38', and 'LD HL, (5...'.
- Breakpoints:** A table for setting breakpoints with columns for address, expression, and code.
- Hardware registers:** A table showing the values of hardware registers. The PC register is 0000, Frame is 00000000, T-St. is 00000000, and Turbo is x1.
- Memory:** A table showing the memory dump. The current instruction at address 0000 is 'DI' with opcode F3. The memory dump shows the instruction bytes and their corresponding assembly instructions.

Registers:

This section shows the value of the different Z80 registers, in red if they were modified since the last operation. Their value can be modified directly. A double click follows the adress on code view, with shift on data.

Furthermore you can see the status of the different flags. Double clicking on them changes their value.

Finally, if you are using decimal mode, the values of the registers show their high and low parts separated if the mouse is hovered over it.

Disassembly:

This section represents a memory disassembly, by default from the PC register.

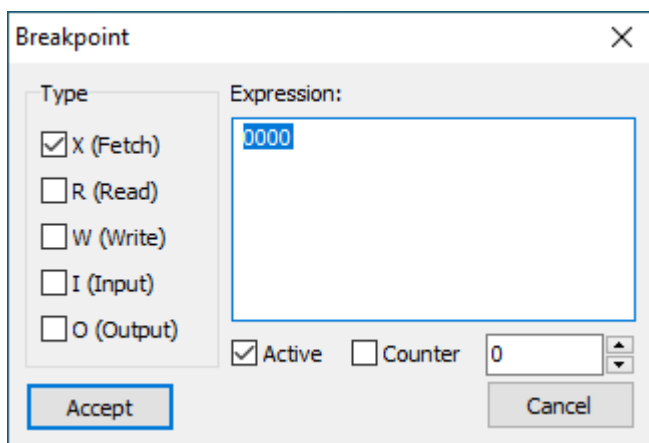
Rows with font in red indicate they have a breakpoint; the one for current PC value has also a bisque background. Double-clicking on a row fills in the data to set a breakpoint in that address.

Below are the following buttons:

- Go To - Changes the start address of the disassembly.
- Step into - Execute only the current instruction.
- Step over - Sets temporary breakpoint at the instruction following the current one, then run.
- Step out - Sets temporary breakpoint at the address indicated at the top of the stack, then run.
- Run - Continues execution.
- Run to - Sets temporary breakpoint at the selected address, then run.
- Tron - Starts / stops execution instruction by instruction mode.
- Up, Down - Browse disassembly. It can also be done using the mouse wheel over the disassembly.

Breakpoints:

This section allows you to define breakpoints. Each one has the following properties:



- Type - The types of accesses for which it occurs. They can be combined between Fetch (X), Read (R), Write (W), Input (I) and Output (O).

- Expression - An expression determines which directions it applies to. It is described below.
- Active - If checked, execution stops at the breakpoint.
- Counter - If checked, Count increases at the breakpoint.
- Count - Value that can be increased with the Counter option.

The buttons down below are for add, remove and edit a breakpoint. A double click over an existing one also edits it, and they can also be dragged to reorder the list. Also, there is a mode that combines BPs: all active for a break.

The expression determines which addresses should cause breakpoint. Its syntax is:

- [Expression]:=[Terms] | [Terms] "=" [Value]
- [Terms]:= [Term] [Terms2]
- [Terms2]:="'" | [Terms]
- [Term]:=[Base] [Pattern] | [Base] [Range]
- [Range]:=[Pattern] "-" [Pattern]
- [Value]:=[Base] [Pattern]
- [Base]:= "" | "%" [Pattern]
- [Pattern]:=[Digit] [Digits]
- [Digits]:="'" | [Pattern]
- [Digit]:="0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "A" | "B" | "C" | "D" | "E" | "F" | "X"

Notes on semantics:

- Default base is 16.
- There is not case sensitive.
- Any whitespace, carriage return, etc. is ignored, except between two [Digit]s.
- The base pattern must be a valid natural number between 2 and 16, in base 10.
- In any other context, a pattern must be a valid natural number in its corresponding base, some of its digits can be "X" indicating that any one is valid.
- A pattern that does not explicitly cover all the required bits (16 for addresses, 8 for values) will set the remaining bits to 0. Any extra bits will be ignored.
- [Base] changes the numeric base of the expression from that point onwards.
- If there is a value, it must match the content of the corresponding memory / IO (this version only match XR); if there is not, always matches.

Examples:

- ---IO "%2 xxxxxxxx0xxxxxxx" - breakpoint at any FDC access.
- X---- "%8 X0" - breakpoint at all RSTs.

Others:

- Stack - Shows the last values in the stack. As with registers, double click follows adress on code, with shift on data.
- Memory map - Displays the memory map of the PCW. With a double click on a row the mapping can be changed temporarily.

Hardware registers:

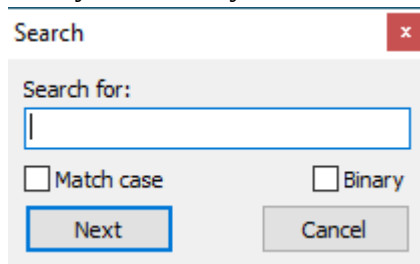
Here the value of some hardware registers is shown; again, in red those that have changed.

Memory:

This window represents a memory dump. The value of each position can be changed by writing a new value on its cell.

Below are the following buttons and options:

- Go to - Changes the start address of the dump.
- Load - Loads a memory dump, starting from the specified address.
- Save - Saves a memory dump, starting at the specified address and length.
- Search - Opens the floating search window. It allows searching for text strings (case) or binary on memory.



- Read / Write - Those are used to choose which memory to work on, to read or to write (they do not have to be the same).
- Grid - Draws a character grid over the screen.
- Exact video - This option causes the screen to be painted in sync and with previous values as future ones. Otherwise, it is painted at whole from the current state.
- Hexadecimal - This option switches all the views and data inputs between decimal and hexadecimal.

Furthermore, while in debugging mode, certain special Z80 instructions are supported:

- ED FE (DEBUG_BP)- Software breakpoint.
- ED FF (DEBUG)- Debugging API; function in A, parameter in BC:
 - 0 - The same as ED FE; BC is ignored.
 - 1 - Prints the 0-ended ANSI string pointed by BC, up to 256 characters.
 - 2 - Shows the value of BC and allows it's modification before continue.
 - 3 - Turns the fast mode off / on, depending on BC being 0 or other value.
 - 4 - Resets cycle counter.
 - 5 - Transfers cycle counter to BC and HL.

PCW Format:

The PCW snapshot format was born from the need to represent everything necessary for emulation (including tapes and discs) in a single file and at the same time to be easily manipulated by the advanced user.

Thus each PCW file is nothing more than a ZIP file, which contains other files inside. In this way it can be manipulated with any compressor application.

There are two types of files inside of it: text (INI, readable and editable with any text editor) and data dumps (other extensions, usually BIN).

There are always a "Config.ini" file (hardware configuration of the emulator), a "Hardware.ini" one (state of said hardware) and "RAM.bin" one (memory dump) present.

Each line in the text files is like "A=B", meaning property A has value B. Inside "Hardware.ini", the Z80 registers are the "Z80_xxx" properties.

Action files are the same as a snapshot, with an extra file with the port and keyboard data values that must be read.

Macros:

The emulator has a small macro interpreter, which consists in the execution of one or more actions of the emulator.

These can be launched from the command line or stored in the macros folder of the emulator.

In the case of being stored in the folder, these consist of two files with the same name and TXT and PNG extensions respectively. When starting the emulator these are listed and added in the form of buttons to the emulator, into their section of the bar, and can be invoked by pressing said button.

The language of macros is composed of tokens (commands or parameters of commands) separated by blanks (spaces, tabs, carriage returns). In the case that a parameter contains spaces, it can be surrounded with quotes to avoid being fragmented in tokens.

The language is very simple: a succession of separate commands, each one followed by its parameters equally separated. Each command has a short form (one character, less writing) and a long form (one word, more intuitive), both of which can be used interchangeably.

The commands are not case-sensitive. In short form the commands can optionally be preceded by - or /, for compatibility.

The execution of these commands is asynchronous to the emulation if it is active, unless otherwise indicated.

The commands are:

Short	Long	Description
8	EXIT	Without parameters. Closes the emulator.
9	WAITK	Without parameters. Waits until the keyboard buffer is empty.
@	EXEC	Take five parameters. The first one is the value of the Z80 register PC, the other four the memory configuration (ports \$F0-\$F3 of GA).
R	RESET	Take an optional parameter. Reset the emulator, and if there is a parameter, it specifies a file with the hardware definition to apply, or makes a hard reset if its *.
M	MAP	Take a parameter. Maps the emulated M unit, the parameter specifies the folder to map or . to deactivate it.
A	DISKA	Take a parameter. If the value is . then remove the disk from drive A, otherwise it inserts the disk image file specified in the parameter.

		Obsolete.
B	DISKB	Same as the previous case, but for unit B.
D	DISK	Similar to the previous two cases, but it takes two parameters. The first is the unit (A or B), the second the specified file or if it is . extracts the current one.
E	SYNC	Without parameters. "Synchronizes" status (stops the emulator and sets it to normal speed). Obsolete.
T	TYPE	Take a parameter. Adds the parameter as string of characters to the keyboard buffer. Supports character # as escape for characters, in hexadecimal two-digit format.
W	WAIT	Take a parameter. Waits for the number of milliseconds determined by it. In case the emulation is stopped, it is started.
F	FULL	Take an optional parameter. This can be OFF or 0 for window mode, ON or 1 for full screen mode, none for toggle.
G	SPEED	Take an optional parameter. OFF or 0 for normal mode, ON or 1 for fast mode, none to toggle.
O	TURN	Take an optional parameter. OFF or 0 to stop emulation, ON or 1 to start it, none for toggle.
X	EXTERNAL	Take two parameters. Executes the external program indicated by the first with the parameters indicated in the second.
P	POKE	Take two parameters. Writes into the absolute memory address indicated by the first the value indicated by the second. Both are numbers in hexadecimal.
Q	DEBUG	Without parameters. Opens the debugger.
S	SNAPLOAD	Take a parameter. Loads the snapshot file indicated by it.
U	SNAPSAVE	Same as previous case, but for saving.
Y	BINLOAD	Take two parameters. Loads into the absolute memory address indicated by the first the binary file indicated by the second.
Z	BINSAVE	Same as previous case, but for saving.

Notes:

- Under construction.

F.A.Q.:

- Where can I get games and applications? => Not here. And for copyright reasons I can not tell you where or how to use them.
- What do I write when "A>" appears? I do not know anything about CP/M. => There are websites dedicated to this.
- Are you going to port it to Linux? Do you need help to program it? => Not at the moment.
- When will the next version come out? => I do not know.
- Are you going to open source it? => Not at the moment.
- Will you emulate the PCW16? => Yes, I have a published emulator for it.
- I cannot create videos => If you have a "N" Windows version, please search for "Media Feature Pack".

- Do you accept donations? => Sure. I accept any PCW material.

Common mistakes:

- The button "A>" is grayed. => It needs a file called CPM.dsk in the emulator folder.
- I have two disk drives and the second one does not work. => The second must be DD or B180.fib must be used.
- Programs like "Trogló" or "ToolKit" do not work. => They need a CP/M BIOS version 1.4 or higher.
- I cannot load drivers when booting CP/M. => They need a CP/M BIOS version 1.7H or higher.
- Isvar from "Fairlight 2" moves on his own. => It is incompatible with the AMX mouse.

More information:

Contact:

- Web: <http://www.habisoft.com/pcw>
- EMail: p_c_w [AT] habisoft [DOT] com

Thanks:

- Thank to you for using this emulator, to the Grupo de Usuarios de Amstrad (G.U.A.) and to the PCWWiki.
- Special thanks to: Frank van Empel for his donation of a disk drive, Kachorro for donate a 9512, two 8256 and a lot of material, G.U.A. (and specially to CPCManiaco and Alt) for donation of a lot of material, Jevicac for donate a Gotek with HxC.
- Finally, thanks to Amstrad for creating the PCW and for allowing the redistribution of copyrighted material.

© 2011-2026 HabiSoft.