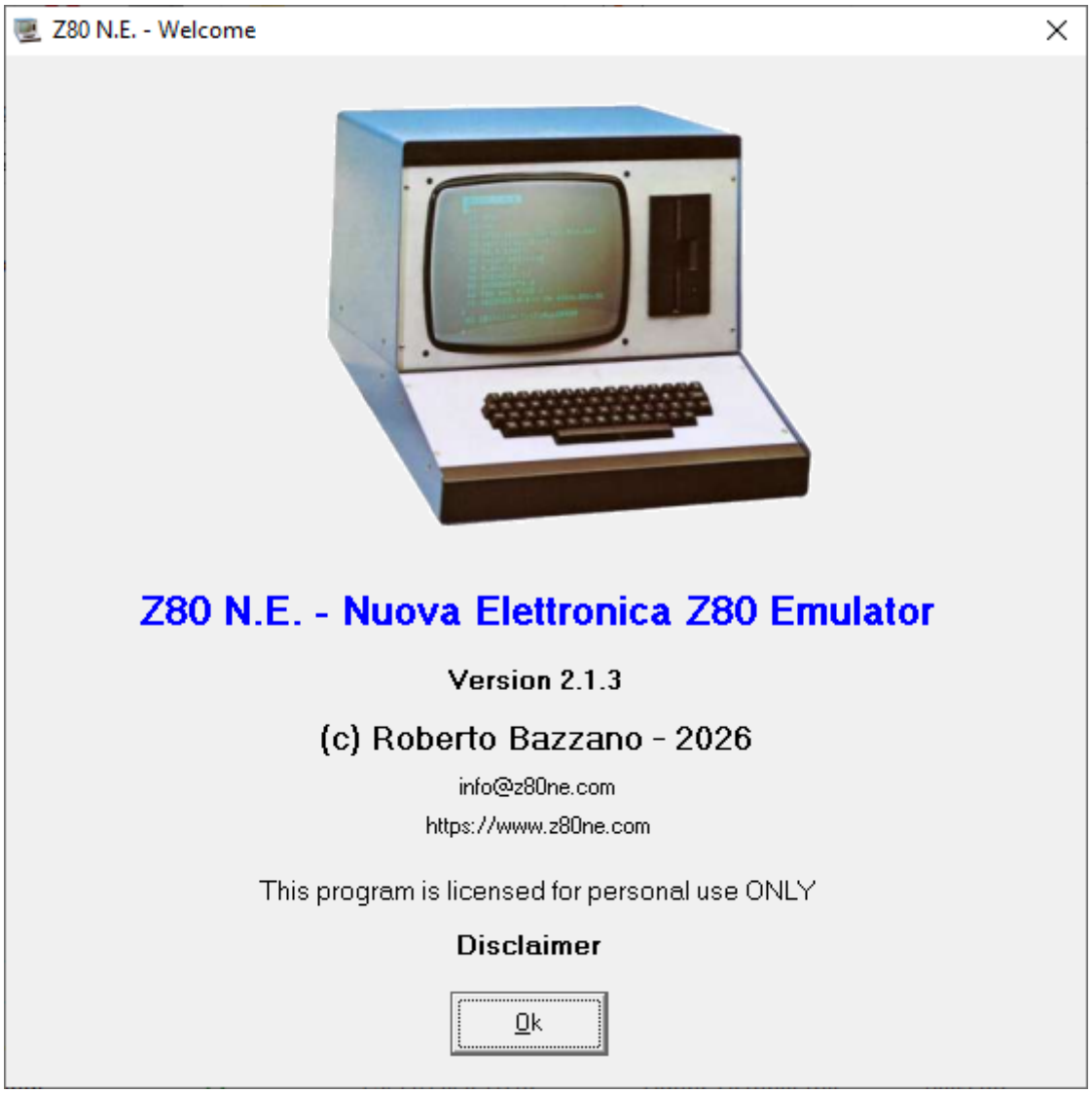
Z80 N.E.

Nuova Elettronica Z80 Emulator



Contents:

General description 4
Install 5
Uninstall 6
Splash screen 7
Main control 8
LX.383 – Hex keyboard and displays 17
LX.385 – Tape interface 18
LX.388 – Video card and monitor 20
LX.390 – Floppy disk interface 22
LX.529 – Hi-res video card and monitor 27
LX.530 – Beeper 29
LX.547 – Interrupt card 30
LX.683 – Hard-disk interface 31
LX.710 – Auto-repeat add-on card 33
Printer 34
Ram editor 36
Load ram 38
Save ram 39
Fill ram 40
Copy ram 41
I/O editor 42
Load input ports – Load output ports 44
Save input ports – Save output ports 45
Disassembler/Assembler 46
Select eprom firmware 48
Address space configuration 50
Load binary files 51
Breakpoints 53
Trace configuration 56
Edit disk or files 58
Disk info 61
Assign file names 62
Edit file 64

File editor 65
Load data (file mode) 67
Save data (file mode) 68
Disk editor 69
Load data (disk mode) 71
Save data (disk mode) 72
File references and general architecture 73

General description

Z80 N.E. Emulator is a software emulator of the **Nuova Elettronica Z80 computer system**. It recreates the original Z80-based machine together with its firmware and hardware expansion cards, allowing users to run it on a modern Windows PC while preserving, as closely as possible, the behavior and operating experience of the original platform.

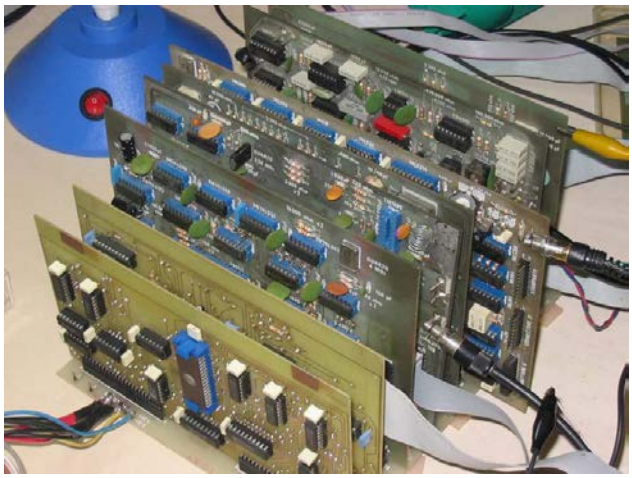
The emulator reproduces not only the CPU itself, but also the main peripherals and expansion boards of the original system, including the hexadecimal keyboard and displays, tape interface, video cards, floppy-disk interface, beeper, interrupt card and hard-disk interface. In addition, it provides several emulator-specific tools, such as a printer window, RAM editor, I/O editor, conditional breakpoints, trace files, disassembler/assembler, files editor and disk editor. As a result, it can be used not only to run the original software environment, but also to inspect, debug and study the machine in depth.

From a historical perspective, the Z80 N.E. belongs to the family of educational and hobbyist computer systems built around the **Zilog Z80** microprocessor and distributed by the Italian publisher **Nuova Elettronica** from October 1979 onward. The purpose of this emulator is therefore not simply to run old software, but to recreate the experience of using that modular system, including its original firmware choices, peripherals and operating environments such as BASIC, NE-DOS and the other boot options provided by the included EPROMs.

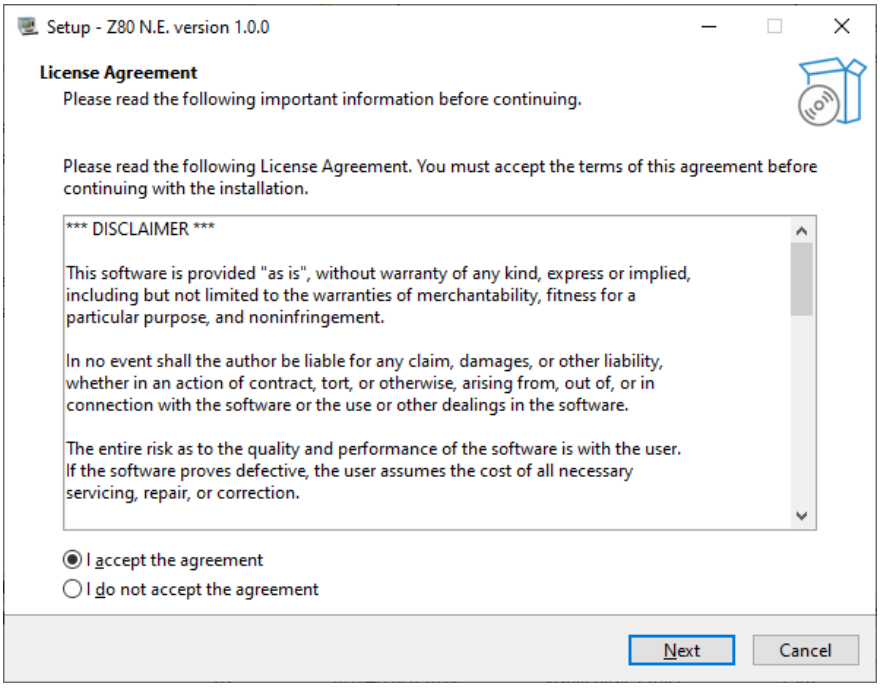
While remaining faithful to the original hardware, the emulator also offers a number of practical features that are useful on a modern PC, such as adjustable CPU speed, step-by-step execution, state save and restore, memory and I/O editing, disassembly, assembly, image-file-based storage and automatic update support.

Further information about the original machine, along with downloadable original software, can be found at https://www.z80ne.com.





Install



To install the application, download it from the official website:

https://www.z80ne.com/eng/z80emu.asp

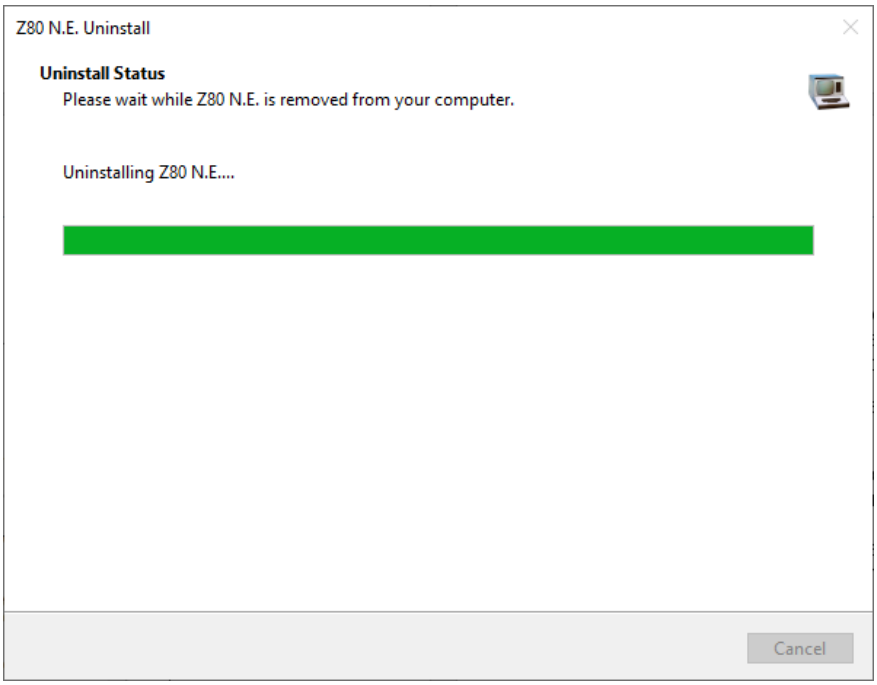
After downloading it, double-click the setup icon.

Then you need to:

- accept the license agreement
- optionally review the What's New page,
- choose an installation directory (or keep the default one)
- optionally choose to create a desktop icon
- click the “Install” button

After the installation is complete, you can optionally launch the Z80 N.E. emulator immediately.

Uninstall

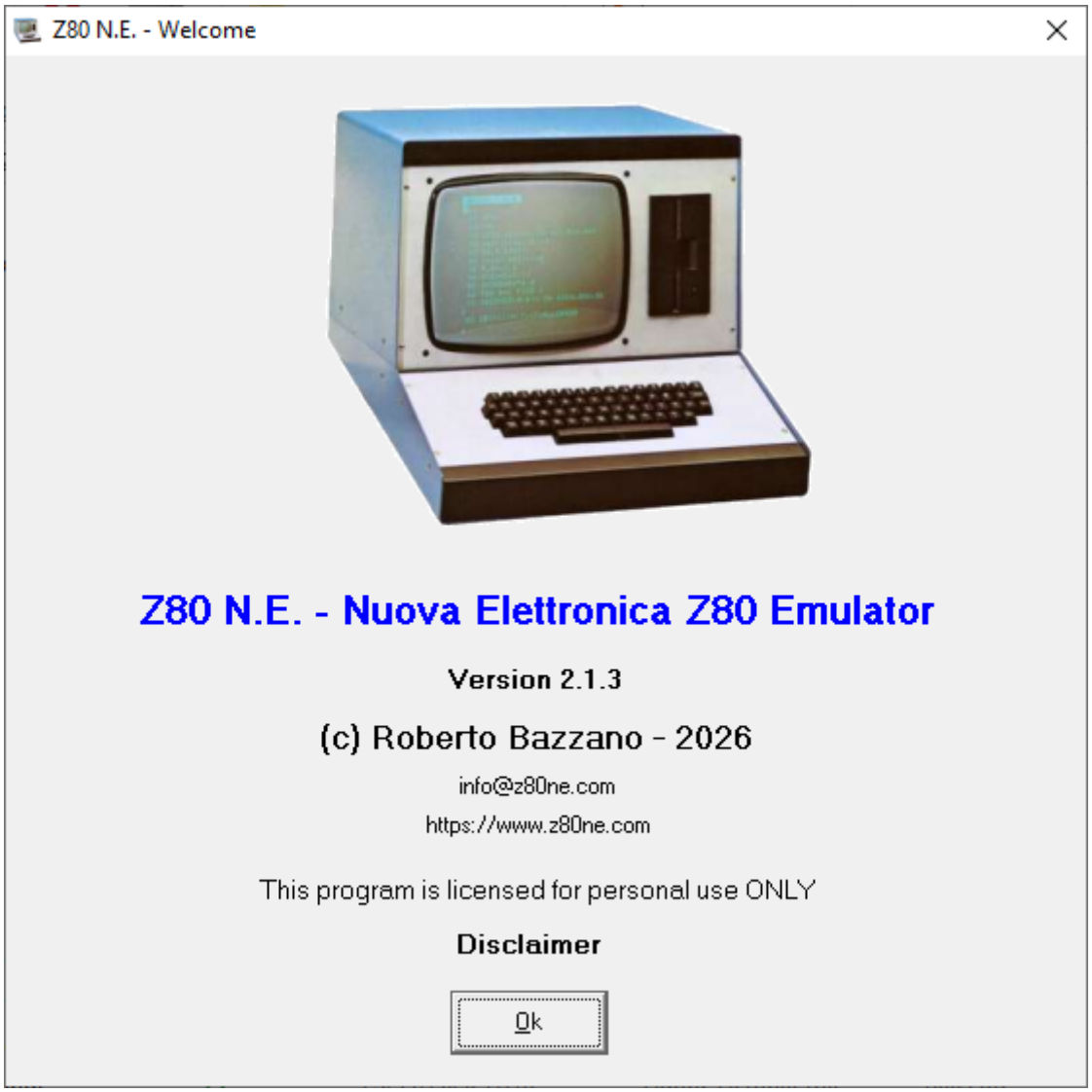


To uninstall the application, go to the Windows Control Panel or Apps settings and double-click “Z80 N.E.”: the uninstall wizard will start.
You can also uninstall Z80 N.E. by clicking the “Uninstall Z80 N.E.” icon in the Windows Start menu.

NOTE: this icon will appear after a reboot or after logging out and back in if you have just installed Z80 N.E.

Simply follow the wizard steps to uninstall the application.
The wizard will ask whether you want to remove the configuration file: select Yes if you no longer need it.
The wizard may then ask whether you want to remove some unused .OCX files. It is safe to remove them; however, if you think you may need them for other programs, you can leave them installed.

Splash screen



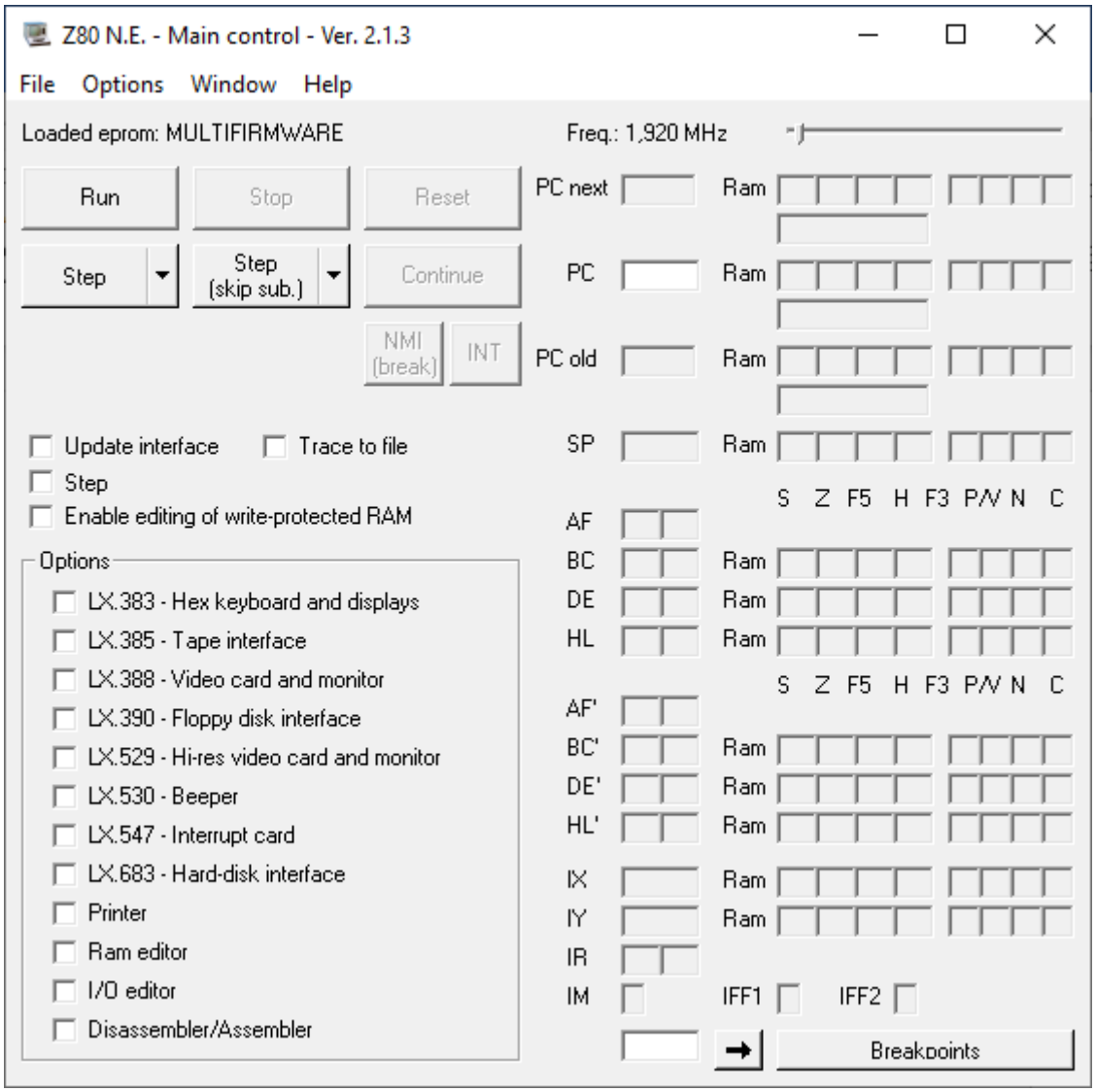
After you run Z80 N.E., a splash screen is displayed. Simply click OK to close it. You can also click the word “Disclaimer” to display the disclaimer, then click OK to accept and close it.

NOTE: when you run Z80 N.E. for the first time, the disclaimer is shown automatically. Please read it and then click OK to accept and close it.

NOTE: When the program starts, a folder named “Z80NE_Software” is created inside the user’s Documents folder.

This folder contains the tape, floppy disk, and hard disk images originally distributed by Nuova Elettronica. These images are used by the “File/Configuration presets” function in the “Main control” window. It is therefore recommended not to delete them.

Main control



After you close the splash screen, the Main control window is displayed, allowing you to run the CPU and manage the main application options.

Here is a brief description of the available options:

MENU ITEMS:

- File/Configuration presets: use this option to load a configuration preset, for example “EP.382 + LX.383 - Hex keyboard and displays”. The selected configuration is loaded, including the EPROM, the associated interfaces, and the software images. The CPU is then automatically set to the RUN state.

- File/Load binary files: use this option to load one or more raw binary files into RAM. A separate load address can be specified for each file, and an overall starting address can also be defined. See the relevant section later in this document.

- File/Save binary file: use this option to save a range of RAM into a raw binary file.

- File/Dump: use this option to save the full emulator state to a .Z80 file. You can restore it later and resume from the exact point at which it was saved.
  NOTE: only CPU registers, RAM, I/O and the internal CPU state are saved. States related to other interfaces are not included.

- File/Restore: use this option to restore the emulator state from a previously saved state file created with the “Dump” menu option.
  NOTE: only CPU registers, RAM, I/O and the internal CPU state are restored. States related to other interfaces are not restored.

- File/Trace configuration: use this option to open the Trace configuration window. See the relevant section later in this document.

- File/View text file: use this option to open a text file with the external program “SnakeTail”. You can use this option, for example, to view trace files created by the “Trace to file” function.

- File/Exit: use this option to close the program. You can also click the upper-right “X” of the window. A confirmation dialog will be displayed before exiting.

- Options/Select eprom firmware: use this option to load a default or custom EPROM firmware image.
  The application includes five default EPROM firmware images, corresponding to the EPROMs originally distributed by Nuova Elettronica for use with the Z80 N.E. computer.
  You can also use a custom EPROM file (raw binary), which can be loaded and/or run at a specified RAM address. See the relevant section later in this document.

- Options/Address space configuration: like the real machine, the Z80 N.E. emulator has a 64 KB address space. When you load an EPROM (default or custom), the address space used by the EPROM is marked as “protected” and is not writable, so programs cannot modify it. All remaining address space is writable and therefore behaves as RAM.
  Using this option, you can change the address ranges that are write-protected, allowing you to simulate missing RAM address space. See the relevant section later in this document.

- Options/Initialize ram on RUN: use this option to initialize RAM when you click the RUN button.
  If this menu option is checked, RAM is initialized according to the following rules after you click RUN:
  - all RAM address space is initialized to FFh except for:
    - address space occupied by loaded EPROM
    - address space occupied by a loaded binary file (loaded with "File/Load binary file" menu option)
    - address spaces selected as write-protected using "Address space configuration" menu option

  If this menu option is unchecked, RAM is not initialized when you click RUN. This is useful if, for example, you want to place some values in RAM before starting emulation for debugging purposes.

- Window/Reset windows position: all windows can be moved or resized. Using this option, all windows are restored to their default position and size.
  Window positions and sizes are preserved between sessions, so when you close the application they are saved to the configuration file. The next time you open the application, you will find the windows exactly as you left them.

- Window/Cascade windows: use this option to arrange all windows in a cascade.

- Window/"window name": open windows are listed below the "Cascade windows" menu option. Click a window name to bring that window to the front.

- Help/About: this menu option displays a window with information about the emulator. You can click the "Close" button to close it, or click the "Disclaimer" button to display the disclaimer. Click "OK" to close the Disclaimer window.

- Help/Check for updates: use this option to manually check for new versions of the emulator. Updates are checked on the official website: https://www.z80ne.com
  If a new update is available, a message asks whether you want to download and install it. If you click "OK", the download starts, the emulator closes automatically, and the setup wizard starts. You will then need to follow the wizard steps to install the update.
  NOTE: an automatic update check is performed once per day when you open the emulator.
  NOTE: for updates to work, your PC must be connected to the Internet and must be able to successfully contact https://www.z80ne.com

WINDOW ITEMS:

- Run button: click this button to power on the CPU.

- Stop button: click this button to power off the CPU.
  NOTE: you do not need to power off the CPU before exiting the application.

- Reset button: click this button to reset the CPU. This has the same effect as pressing the reset button on the real machine (on the CPU card or on the Hexadecimal keyboard card).

- Step button: click this button to enter step-by-step mode, or to advance to the next CPU instruction while already in step-by-step mode.
  In step-by-step mode, you can execute CPU instructions one at a time: the CPU stops before executing the next instruction, and you can view and modify CPU registers and RAM.
  Next to the Step button there is a small triangle that opens a menu where you can change the behavior of the Step button. If you select the "Step (auto)" menu option, clicking the "Step (auto)" button makes the CPU automatically execute instructions at the step rate selected by the "Select steps per second" option. Click the "Step (auto)" button again to stop auto-step.
  NOTE: interrupts (both NMI and INT, for example the NMI generated by pressing the BREAK keys on the alphanumeric keyboard) are serviced only after you click the STEP button.

- Step (skip sub.) button: similar to the "Step" button, but if the next instruction is a "CALL" (or a conditional variant such as "CALL NZ") or a "RST" (or a conditional variant such as "RST NZ"), the called subroutine is executed completely before step-by-step mode resumes. Step-by-step mode resumes after the CPU encounters the instruction following the "CALL" or "RST" instruction.

- Continue button: click this button to exit step-by-step mode and let the CPU run continuously.
  NOTE: after clicking this button, the user interface remains in "Update interface" mode, meaning that you will see all registers and RAM updating in near real time. To stop these updates, clear the "Update interface" checkbox.

- NMI (break) button: click this button to generate a non-maskable interrupt, just like pressing the BREAK keys on the emulated keyboards.
  NOTE: you can also press CTRL+SHIFT on your PC keyboard to simulate the BREAK keys and generate a non-maskable interrupt.

- INT button: click this button to generate a maskable interrupt.

- Running frequency label: below the "Step" button there is a label that displays information while the CPU is running. The following values are shown:
  - Current CPU frequency. It may differ from the selected CPU frequency if the PC running the emulator is not powerful enough to sustain the requested speed.
  - Opcodes per second. This is the average number of instructions executed by the emulated CPU every second.
  - Microseconds per opcode. This is the average time, expressed in microseconds, used by the CPU to execute an instruction.

- Stack depth and Max stack depth labels: below the “Running frequency” label there are two labels that display the current stack depth and the maximum stack depth. They are shown only if the “Update interface” or “Step” checkbox is checked.
  Stack depth is incremented by 1 each time the SP register is decremented, and decremented by 1 each time the SP register is incremented. It’s also reset to zero when SP is loaded by instructions such as “LD SP,xxx”, for example “LD SP,HL”.
  Max stack depth is the maximum value reached by Stack depth. Double-click either label to reset the value shown by that label.

- Update interface checkbox: select this checkbox to enable “Update interface” mode. In this mode, the user interface is updated in near real time, so you can see all registers and RAM changing while the CPU is running. Clear it to disable “Update interface” mode.

- Trace to file checkbox: select this checkbox to start tracing your program. When you select it, if the trace configuration has not yet been set, the configuration window is displayed, where you can enter a filename and configure the other trace parameters. See the relevant section later in this document.
  If the trace configuration has already been set, tracing starts immediately. Each instruction executed by the CPU, according to the Start and Stop trace ranges defined in the Trace configuration window, is written to the selected trace file together with the CPU register values until you clear this checkbox. A label below the Trace to file checkbox shows how many bytes have currently been written to the trace file.
  When you clear this checkbox, a save pop-up is shown while the trace file is being saved to disk. This operation may take some time.
  WARNING: the trace file can become very large. Please keep an eye on the number of bytes written, as this may lead to disk-full errors and/or long save times.

- Step checkbox: you can use this checkbox to enter or leave step-by-step mode, just like clicking the “Step” or “Continue” button.

- Enable editing of write-protected RAM: if this checkbox is not selected, and the current RAM location is write-protected (for example because it belongs to the address space used by the EPROM), RAM values are shown with a gray background and cannot be changed.
  To override the write lock, select this checkbox and you will be allowed to change RAM values even within write-protected address space.

- Frequency selector: drag this selector to choose the CPU operating frequency. This is essentially a limiter: if you move it all the way to the right, the emulator runs at the maximum frequency allowed by your PC CPU; if you move it to the left, the CPU frequency is limited accordingly.
  The selected frequency is shown to the left of the selector.
  The real machine frequency is 1.920 MHz. You can set the default frequency by double-clicking the selected-frequency label.
  Right-clicking the frequency label or the frequency selector lets you enter the desired frequency value in MHz.
  NOTE: when running at the maximum allowed frequency (selector all the way to the right), the CPU frequency will not be stable, because it also depends on internal emulator operations. This is normal and expected behavior.

- Registers labels: when in step-by-step mode, double-click one of these labels to automatically position the disassembler/assembler at the address shown by the corresponding register displayed to the right of the label. If the disassembler/assembler window is closed, it is automatically opened first.

- Ram labels: when in step-by-step mode, double-click one of these labels to automatically position the RAM editor at the address shown by the corresponding register displayed to the left of the label. If the RAM editor is closed, it is automatically opened first.

The following note applies to all register descriptions below, unless otherwise specified:

NOTE: To the right of each register there are four bytes showing the RAM contents at the address pointed to by the 16-bit register (or 8-bit register pair), first in hexadecimal format and then in ASCII format. While in step-by-step mode, you can modify the register value, and you can also modify the RAM contents shown beside it in either hexadecimal or ASCII format.

- PC next register: this is a "virtual" register, not present on the real CPU. It stores the 16-bit address of the instruction that will be executed after the one currently indicated by the "PC register" field. This register cannot be modified in step-by-step mode.
  Below the PC next register you can see the disassembled instruction at the next PC address.

- PC register: this is the 16-bit PC (Program Counter) CPU register. It indicates the instruction that will be executed by the CPU. Below the PC register you can see the disassembled instruction at the PC address.

- PC old register: this is a "virtual" register, not present on the real CPU. It stores the 16-bit address of the previous instruction executed by the CPU. This register cannot be modified in step-by-step mode.
  Below the PC old register you can see the disassembled instruction at the previous PC address.

- SP register: this is the 16-bit SP (Stack Pointer) CPU register.

- AF registers: these are the 8-bit A and F (Accumulator and Flags) CPU registers. In step-by-step mode, you can double-click the decoded flags of the F register (shown to its right) to toggle individual bit values between 0 and 1.

- BC registers: these are the 8-bit B and C CPU registers.

- DE registers: these are the 8-bit D and E CPU registers.

- HL registers: these are the 8-bit H and L CPU registers.

- AF' registers: these are the alternate 8-bit A and F (Flags) CPU registers. In step-by-step mode, you can double-click the decoded flags of the F register (shown to its right) to toggle individual bit values between 0 and 1.

- BC' registers: these are the alternate 8-bit B and C CPU registers.

- DE' registers: these are the alternate 8-bit D and E CPU registers.

- HL' registers: these are the alternate 8-bit H and L CPU registers.

- IX register: this is the 16-bit IX CPU register.

- IY register: this is the 16-bit IY CPU register.

- IR registers: these are the 8-bit I and R CPU registers.

- IM register: this is the Interrupt Mode state. Its value can be 0, 1 or 2.

- IFF1 and IFF2: these are the Interrupt Flip-Flop 1 and 2.

- NMI and INT labels: to the right of IFF2 there are two red labels, NMI and INT, normally hidden. These labels are shown in Step-by-Step mode while an NMI or INT interrupt is being serviced. They are hidden again when a RETN or RETI instruction is executed. For INT, the label is also hidden after the execution of the EI and RET instructions in sequence. Double-click these labels to reset them.

- Breakpoints field, RIGHT arrow and Breakpoints button: you can add one or more breakpoints, that is, addresses at which the CPU automatically enters step-by-step mode, by typing an address in this field and clicking the arrow button to the right, or by pressing ENTER on the keyboard.
  If you press the Breakpoints button, a configuration window is displayed where you can define more advanced breakpoint conditions or view the active breakpoints. See the relevant section later in this document.
  When a breakpoint is triggered during execution, move the mouse over the "Breakpoints" button to display a tooltip containing the condition that triggered the breakpoint and the number of CPU T-States elapsed since the previous breakpoint, or since the RUN button was pressed if this is the first breakpoint triggered after starting the CPU.

- Options:
  - LX.383 - Hex keyboard and displays: this checkbox opens/closes the window for LX.383 emulation.

  - LX.385 - Tape interface: this checkbox opens/closes the window for LX.385 emulation.

  - LX.388 - Video card and monitor: this checkbox opens/closes the window for LX.388 emulation.

  - LX.390 - Floppy disk interface: this checkbox opens/closes the window for LX.390 emulation.

  - LX.529 - Hi-res video card and monitor: this checkbox opens/closes the window for LX.529 emulation.

  - LX.530 - Beeper: this checkbox enables/disables LX.530 emulation. No window is displayed for this emulation.
    You can enable LX.530 even if LX.529 is not enabled, unlike the real machine where LX.530 is a daughter board of LX.529.

- LX.547 - Interrupt card: this checkbox enables/disables LX.547 emulation. No window is displayed for this emulation.
  NOTE: as on the real machine, LX.547 must be disabled if you want to use the SONE operating system, because SONE does not support it.

- LX.683 - Hard-disk interface: this checkbox opens/closes the window for LX.683 emulation.

- Printer: this checkbox opens/closes the printer emulation window.

- RAM editor: this checkbox opens/closes the RAM editor window, which lets you view and modify RAM contents.

- I/O: this checkbox opens/closes the input/output editor window, which lets you view and modify input and output ports.

- Disassembler/Assembler: this checkbox opens/closes the disassembler/assembler window.

HOW TO OPERATE MAIN CONTROL

When you start the application, the MULTIFIRMWARE preset is loaded by default.

This preset includes a custom-made EPROM containing all the original EPROM firmware images, which you can run using an on-screen menu, just like on the real machine.

You can also load different presets using the File/Configuration preset menu option, or set your desired configuration by enabling or disabling the different option checkboxes in the Main control window.

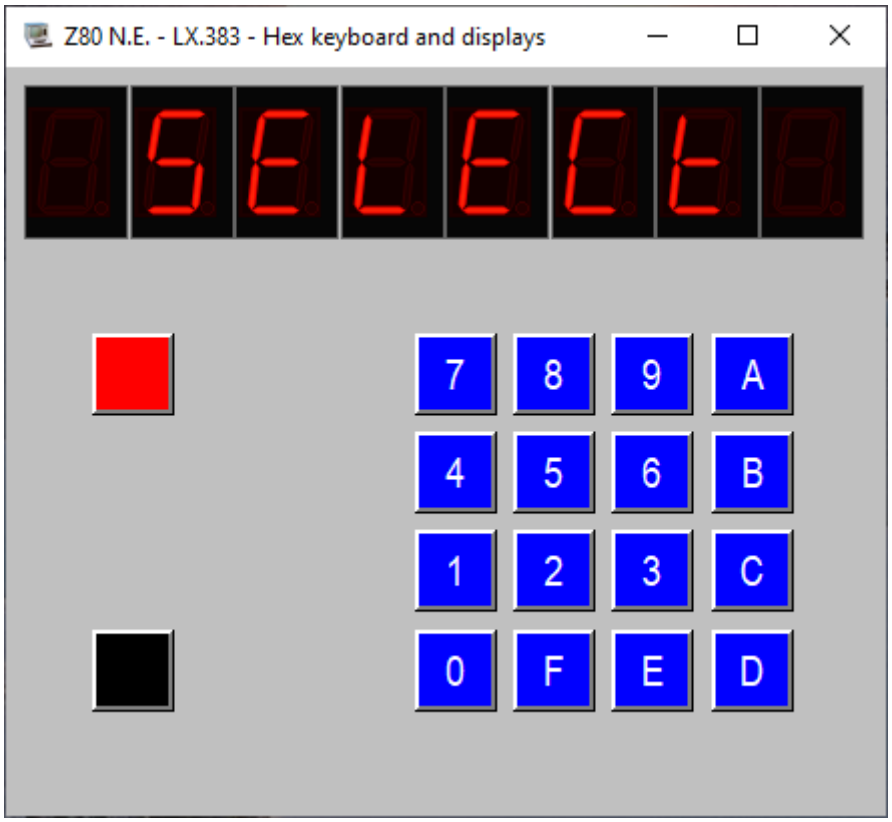
For example, you can open LX.388 and LX.529, which are the video interface cards on which the MULTIFIRMWARE main menu is displayed, or LX.383, the Hexadecimal keyboard and displays interface.

You can of course enable other options as desired, and you can also load and run a different EPROM using the Options/Select eprom firmware menu.

Many bootable floppy-disk images for use with the LX.390 interface, as well as many audio software images for the LX.385 tape interface, can be downloaded from:

https://www.z80ne.com/default.asp?page=/eng/software.asp

LX.383 – Hex keyboard and displays



This window emulates the LX.383 interface card and the LX.384 keyboard and display card.

You can operate the keys either by clicking them with the mouse or by typing on your PC keyboard.

The red button is the RESET button: it resets the CPU (it has the same effect as the “Reset” button in the “Main control” window).

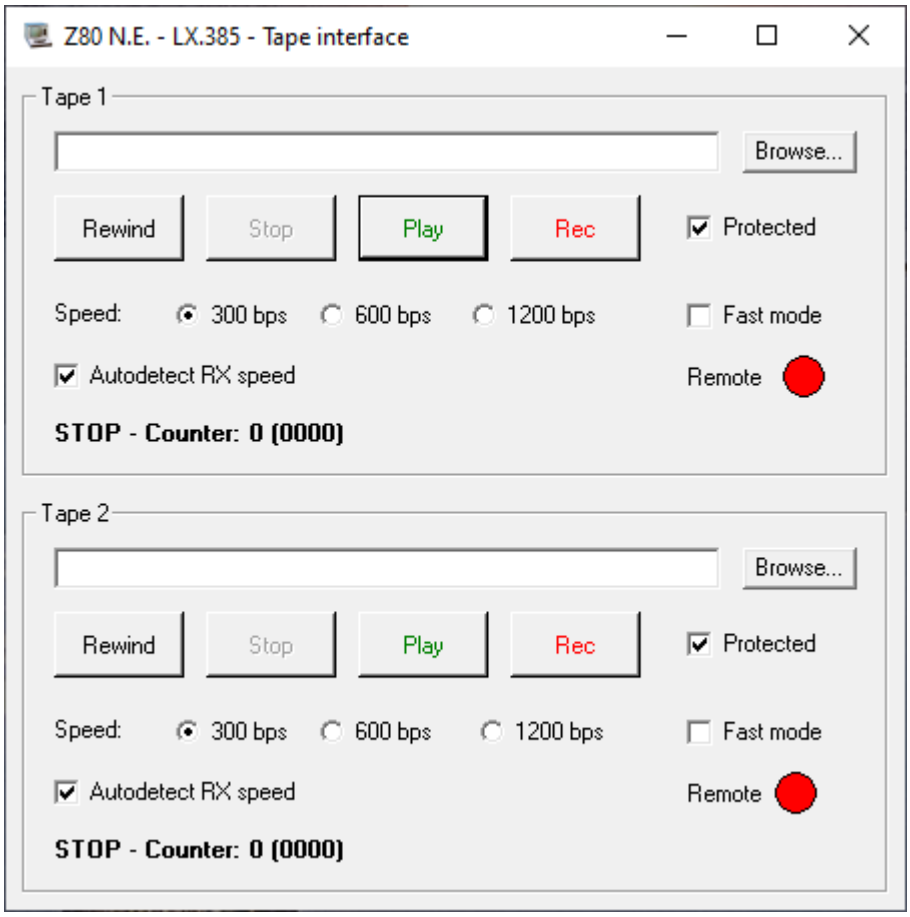
The black button is the CONTROL button. You must press it together with a value button (the blue buttons) to activate the related function, depending on the loaded EPROM. Press and hold the CONTROL button, then press a value button.

You can use the CONTROL button by pressing the CTRL key together with a value key on your PC keyboard, or by clicking the black button with the mouse. If you click it with the mouse, it behaves as a toggle button, so it stays pressed until you click it again or click a blue button.

As an addition to the real hardware, pressing CTRL+SHIFT on your PC keyboard generates an NMI interrupt, equivalent to pressing the BREAK keys on the real machine alphanumeric keyboards.

NOTE: to use your PC keyboard, this window must have focus, so click on it if you are unsure.

LX.385 – Tape interface



This window emulates the LX.385 interface card and two tape players/recorders.

Tape data is saved in .WAV files. You can create your own with the emulator, or download them from the Z80 N.E. website at: https://www.z80ne.com/eng/software.asp

To use LX.385 emulation, you need to load a .WAV file into the first and/or second virtual tape. You can then set the various options and press the tape control buttons (Rewind, Stop, Play, Rec) when instructed by the software running on the Z80 CPU.

NOTE: as on the real machine, when tape emulation is loaded both REMOTE motor controls are enabled (the LED is red). This means that if you click "Play", the tape starts playing immediately. This is the reason why the tape command buttons must be pressed only when instructed by the software.

Here is a brief description of the available options. They are identical for both tapes:

- Text field: in this field you can type the path and filename of the .WAV file containing tape data.
- Browse button: click this button to locate and load a .WAV file containing tape data. In the file-selection window you can also type a filename that does not yet exist, and the program will ask whether you want to create a new file.

- Rewind button: click this button to move the tape counter back to the start. The next play or record operation will then begin at the start of the .WAV file.

- Stop button: click this button to stop Play or Record.

- Play button: click this button to start playback. For playback to occur, the “Remote” LED must also be red. The Remote LED is controlled by the software running on the CPU.

- Rec button: click this button to start recording. To record data, the “Protected” checkbox must be cleared.

- Protected checkbox: if selected, this protects the tape from being written to. It is equivalent to removing the rear tab from a real cassette tape.

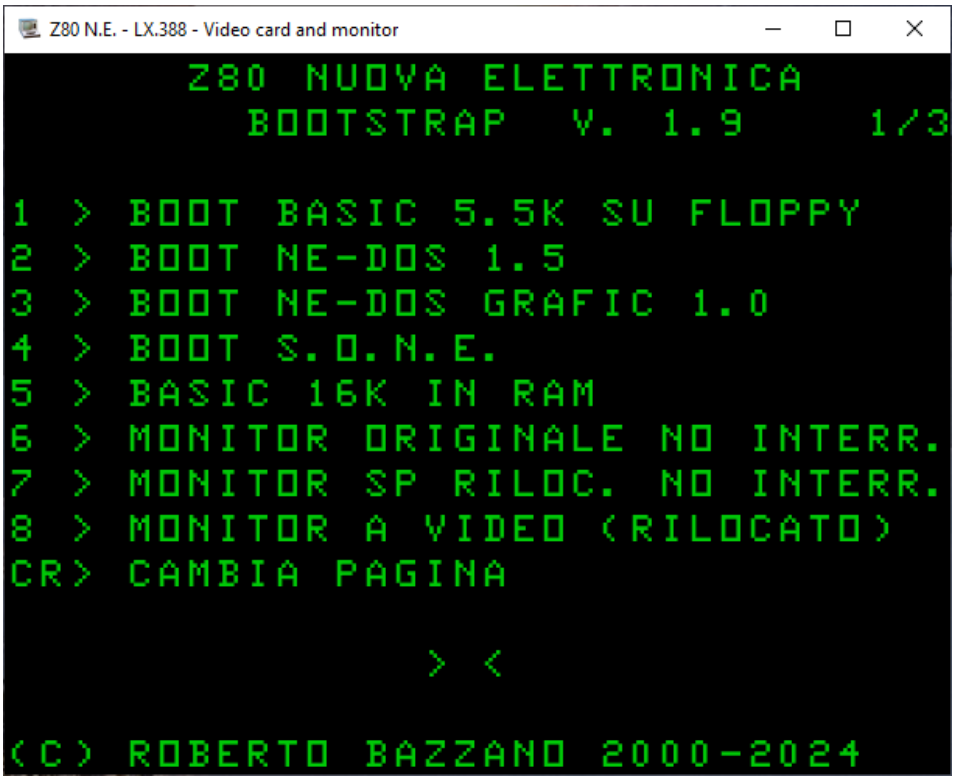
- Speed options: use these options to manually set the tape-data read or write speed. The available speeds are 300 bps, 600 bps and 1200 bps, as on the real machine.
  NOTE: To use the manual speed options during read, “Autodetect RX speed” must be cleared.

- Fast mode: If this option is enabled, read and write operations are performed at the maximum emulation speed available (depending on the PC CPU). The data being read or written is still processed using the correct baud rate, that is, 300, 600, or 1200 bps depending on the setting selected with the Speed options.
  NOTE: at the start and end of tape data there are leader and trailer sections. With Fast mode enabled, these sections are read/written at maximum speed by the emulator, but the firmware running on the CPU may include delays, so they can still take longer. This does not depend on the emulation itself, but on the EPROM firmware running on the CPU.

- Autodetect RX speed: with this option enabled, you do not need to manually select the reading speed. The emulator automatically sets the speed based on the data contained in the .WAV file.
  NOTE: writes are always performed at FAST speed.

- Status label: below “Autodetect RX speed” there is a status label. It displays the current tape-emulation status together with a counter. The counter represents the current byte being read or written, in both decimal and hexadecimal form (hexadecimal shown in brackets).

LX.388 – Video card and monitor



This window emulates the LX.388 interface card and a green, amber or white-phosphor monitor.

When this window has focus, you can type on your PC keyboard and all keystrokes are sent to the Z80 CPU. You can also use CTRL+V on your PC keyboard to paste text from Windows Notepad.

There are a few key mappings as follows:

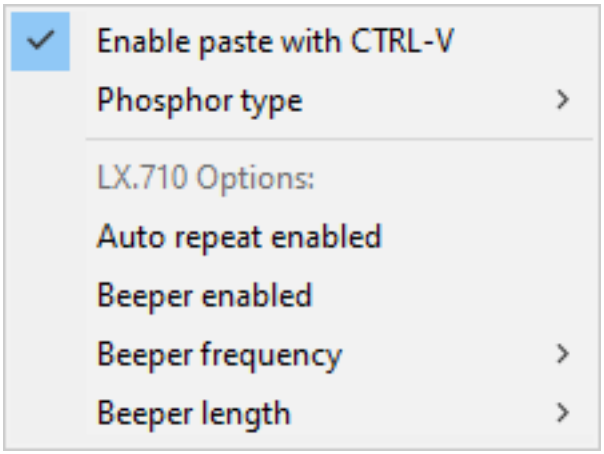
- BREAK key on the real machine: press CTRL+SHIFT. This generates an NMI on the CPU, just like on the real machine.
- LINE FEED: press the “Pg Down” key.

NOTE: when you use NE-DOS, to pause listing you must press SHIFT+@ on your keyboard, as on the real machine.

NOTE: when you use BASIC 5.5K, to delete characters (backspace) you must press the DEL key on your keyboard, as on the real machine.

NOTE: to use your PC keyboard this window must have focus, so click on it if you are unsure.

Right-click on the screen to open a pop-up menu for configuring additional settings, including the settings of the LX.710 auto-repeat add-on card.

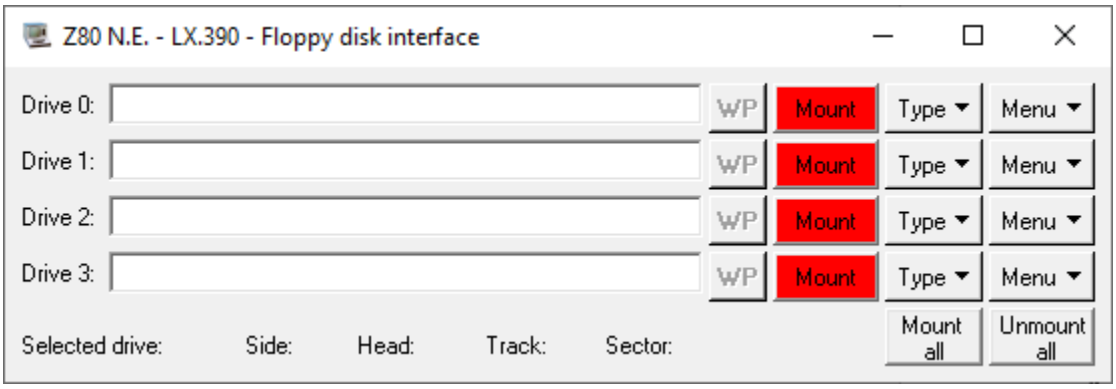


With the pop-up menu you can:

- enable or disable CTRL-V to paste text from Windows Notepad
- change the phosphor type to:
  - green
  - amber
  - white

See the relevant section later in this document for LX.710 settings.

LX.390 – Floppy disk interface



This window emulates the LX.390 interface card and four floppy disk drives.

Floppy disk data is saved in standard .IMD or .DMK image files, or on physical floppy disks. You can create your own image files with the emulator, or download them from the Z80 N.E. website at: https://www.z80ne.com/eng/software.asp

NOTE: image files downloaded from the https://www.z80ne.com/eng/software.asp website do not need to be converted as explained on the website; they can be used directly without datamark conversion.

If you have a Greaseweazle USB interface (https://github.com/keirf/greaseweazle), you can also use physical floppy-disk drives. Simply connect the Greaseweazle to a USB port on your PC, connect up to four drives to the Greaseweazle interface, select a physical drive using the Type button, and mount it with the Mount button. The emulator can then access the physical drive to read and write data just as it does with an image file. You can of course read, write, format, and copy data or entire disks between physical floppy disks, or between physical floppy disks and image files, using your standard operating system commands.

A list of places where you can purchase a Greaseweazle interface can be found here: https://github.com/keirf/greaseweazle/wiki/Purchase-a-Greaseweazle

To use LX.390 emulation, you need to load a disk image file into one of the virtual drives using the Menu/Browse button, and then mount it. Alternatively, you can select a physical floppy drive using the Type button.

Here is a brief description of the available options. They are identical for all drives:

- Text field: in this field you can type the path and filename of the disk-image file containing disk data. Both .IMD and .DMK image file types are supported.

- WP button: this is the “write protect” button. Click this button to write-protect an image file. This is equivalent to write-protecting a floppy disk on the real machine. The button background turns red when write-protect is active.

- Mount (Unmount) button: click this button to mount or unmount the image file. This is equivalent to physically inserting or removing a floppy disk from a disk drive on the real machine. If an image file is mounted, the button background turns green.

- Type button: this button opens a sub-menu with the following options:
  - Image: select this option if the floppy disk data is stored in an image file.

  - Physical: select this option if the floppy disk data is stored on a physical floppy disk drive. This option opens another sub-menu:
    - Greaseweazle: select this option to use the Greaseweazle interface. This option opens another sub-menu:

      - Verify writes: select this option to verify every write operation performed on the physical floppy disk. When this option is selected, all write operations are verified, and an error message is displayed if any errors occur.

      - Physical drive number: select this option to map a physical drive to the virtual drive. For example, if you click the Type button for Drive 0 (the virtual drive), you can assign physical drive 0, 1, 2 or 3 to virtual drive 0.

- Menu button: this button opens a sub-menu with the following options:
  - Browse: select this option to locate and load a disk-image file containing disk data. In the file-selection window you can also type a filename that does not yet exist, and the program will ask whether you want to create a new file. The selected file is automatically mounted (the Mount button background turns green).
    NOTE: a newly created image file is empty and must be formatted by the operating system running on the emulator CPU.

  - Move: select this option to move a floppy disk from one drive to another.

  - Swap: select this option to swap the floppy disks between two drives.

  - Edit disk or files: select this option to edit either the entire disk or individual files stored in the disk image or on the physical floppy disk. See the relevant section later in this document.

  - Disk info: select this option to display the Disk Info window, which shows information about the disk.

- Mount all button: click this button to mount the image files or physical disks of all four virtual drives together. Only disks already present in the text fields are mounted.

- Unmount all button: click this button to unmount the image files or physical disks of all four virtual drives together.

- Status labels: at the bottom of the window there are Selected drive, Side, Head, Track and Sector labels.
  These labels reflect the values of the currently selected disk.
  NOTE: Head is the position of the read/write head of the emulated drive, while Track is the current track loaded in the emulated FD1771 controller register. In some cases, they could theoretically differ.

A few notes about how image files and physical drives work.

The drives shown in the LX.390 user interface are virtual drives. The emulator "sees" these virtual drives and can read from and write to them.

A virtual drive can be mapped to an .IMD or .DMK image file, or to a physical drive through a Greaseweazle USB interface.

The Mount button serves exactly this purpose: it lets you map a virtual drive to an image file or a physical drive.

Virtual drives use an internal disk cache in the emulator. This means that when you mount a virtual drive, the entire image file is read from your hard disk and loaded into the emulator's internal disk cache. The emulator then accesses only its internal disk cache to read and write data.

The same applies to physical disks: when you mount a physical disk to a virtual drive, a preload phase starts. During this phase, the emulator reads the first 40 tracks of the physical disk and loads them into its internal disk cache.

This is a background process and is non-blocking: if the operating system requests access to a track that has not yet been loaded into the disk cache, the track is immediately read from the physical disk.

All writes made to the emulator disk cache are flushed to the image file or physical disk as follows:

- for image files, the entire disk cache is written to the image file after 500 ms of inactivity on all virtual drives mapped to an image file
- for physical disks, the disk cache is written to the physical disk after 500 ms of inactivity on the virtual drive mapped to that physical drive.

Please note that all read/write operations on the disk cache are synchronous. This means that the emulator CPU stops while the emulated controller (FD1771) is accessing the disk cache. This is generally not a problem, because access to the disk cache is very fast and the CPU is still able to maintain its designated clock frequency.

All access to physical drives is handled by a separate process (Z80NE_floppy.exe), which is launched automatically when you open the LX.390 user interface. This means that access to physical drives, which is typically much slower, is asynchronous and non-blocking.

Obviously, you can also use an HxC floppy emulator or a similar device in place of a physical floppy drive, allowing you to use SD cards or, depending on the device, USB storage instead of floppy disks (although simply mounting the images directly in the emulator is sufficient). Connect the device to the Greaseweazle interface in the same way as a physical floppy drive.

NOTE: the Greaseweazle interface must have firmware version 1.6 or later to be used with the emulator.

Note about the DISK CHANGE signal.

Many modern floppy drives (for example, 3.5-inch models) provide the DISK CHANGE signal on pin 34 of the drive data connector.

The emulator continuously checks the drive to determine whether this signal is present and whether the floppy disk has been manually ejected. This allows the emulator to detect the condition and react accordingly, limiting the problems that manual ejection may cause if the drive has not first been unmounted using the Unmount button in the LX.390 window (for example, data loss during write operations).

Many older drives, such as numerous 5.25-inch or 8-inch models, provide the READY signal instead of DISK CHANGE on pin 34 of the drive data connector. The READY signal cannot be used by the emulator to determine whether a floppy disk has been manually ejected. The emulator does its best to identify possible errors and limit any resulting problems, but without the DISK CHANGE signal, manual ejection cannot be detected.

Therefore, if the drive has a configuration DIP switch that lets you select which signal is provided on pin 34, it is recommended to configure it for the DISK CHANGE signal.

To check whether the drive is configured to provide the DISK CHANGE signal, proceed as follows:

- Insert a floppy disk into the drive.
- Select the physical drive using the Type button in the LX.390 window.
- Mount the physical drive using the Mount button in the LX.390 window: the Preload phase will start on the drive.
- Eject the floppy disk from the drive while the Preload phase is still in progress.

After about one second, if the DISK CHANGE signal is active, an error message will appear indicating that the floppy disk has been removed from the drive. If the Preload phase continues without any error message appearing, the drive is not providing the DISK CHANGE signal on pin 34 of the drive data connector.

WARNING – WARNING – WARNING

NEVER eject, remove, unmount, or disconnect a physical floppy disk, an SD card or USB storage device while the activity LED of the corresponding drive or device is lit. Doing so may cause catastrophic data loss and could corrupt or destroy the contents of the storage medium.

To remove a physical floppy disk, SD card, or USB storage device, **ALWAYS** unmount it first using the Unmount button in the LX.390 user interface. Only when the virtual drive is unmounted (the Unmount button changes its text to Mount and turns red) is it safe to remove the physical medium or storage device.

LX.529 – Hi-res video card and monitor



This window emulates the LX.529 interface card and a green, amber or white-phosphor monitor.

All text and graphics modes supported by the real machine are emulated.

When this window has focus, you can type on your PC keyboard and all keystrokes are sent to the Z80 CPU. You can also use CTRL+V on your PC keyboard to paste text from Windows Notepad.

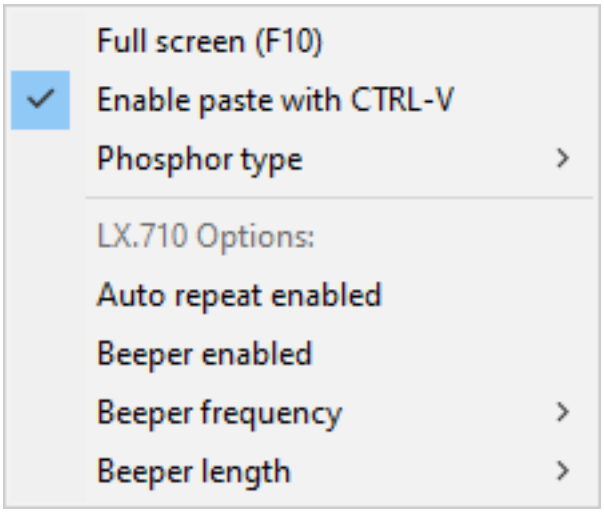
There are a few key mappings as follows:

- BREAK key on the real machine: press CTRL+SHIFT. This generates an NMI on the CPU, just like on the real machine.
- LINE FEED: press the “Pg Down” key.

NOTE: when you use NE-DOS, to pause listing you must press SHIFT+@ on your keyboard, as on the real machine.

NOTE: to use your PC keyboard this window must have focus, so click on it if you are unsure.

Right-click on the screen to open a pop-up menu for configuring additional settings, including the settings of the LX.710 auto-repeat add-on card.



With the pop-up menu you can:

- set the window to full-screen mode, hiding the title bar. You can also enable or disable full-screen mode by pressing F10 on the keyboard
- enable or disable CTRL-V to paste text from Windows Notepad
- change the phosphor type to:
  - green
  - amber
  - white

See the relevant section later in this document for LX.710 settings.

LX.530 – Beeper

This emulates the LX.530 beeper daughter board.

There is no window for this emulation.

Unlike the real hardware, where LX.530 is a daughter board of LX.529 and therefore can only exist when LX.529 is present, in this emulator LX.530 can be enabled independently of LX.529.

LX.547 – Interrupt card

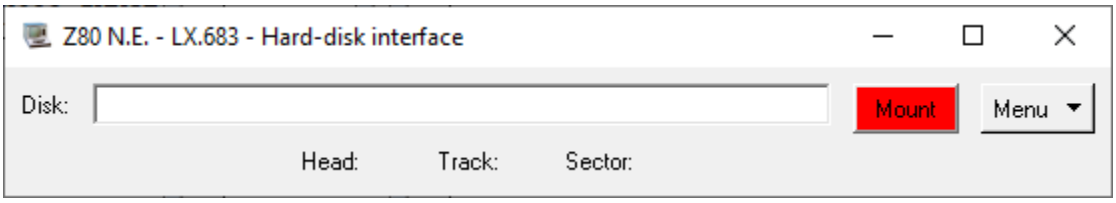
This emulates the LX.547 interrupt card.

There is no window for this emulation.

The interrupt card is used by NE-DOS to update its internal time counters and to enable trace functions in BASIC.

NOTE: if you use the SONE operating system, do not enable LX.547, because SONE does not support it and will not work correctly.

LX.683 – Hard-disk interface



This window emulates the LX.683 interface card and one 10 MBytes hard-disk drive.

Hard-disk data is saved in a proprietary raw .IMG image file, with sectors stored in CHS order (cylinder, head, sector). The hard-disk geometry is: 360 cylinders, 4 heads, 32 sectors per track, 256 bytes per sector, for a total of 11,796,480 data bytes.

Empty image files can be created directly with the emulator, and must then be formatted by the software running on the CPU.

To use LX.683 emulation, you need to load an image file into the virtual disk and then mount it.

Here is a brief description of the available options:

- Text field: in this field you can type the path and filename of the image file containing hard-disk data. The image file must have the .IMG extension.

- Mount (Unmount) button: click this button to mount or unmount the image file. This is equivalent to physically powering on or off the hard disk. If an image file is mounted, the button background turns green.

- Menu button: this button opens a sub-menu with the following options:
  - Browse button: click this button to locate and load a disk-image file containing disk data. In the file-selection window you can also type a filename that does not yet exist, and the program will ask whether you want to create a new file. The selected file is automatically mounted (the Mount button background turns green).
    NOTE: a newly created image file is empty and must be formatted by the operating system running on the emulator CPU.

  - Edit disk or files: select this option to edit either the entire disk or individual files stored in the hard-disk image. See the relevant section later in this document.

  - Disk info: select this option to display the Disk Info window, which shows information about the disk.

- Status labels: at the bottom of the window there are Head, Track and Sector labels.
  These labels reflect the values of the currently accessed head, track and sector of the hard disk.

NOTE: there are a few points to keep in mind:

- If LX.683 is not enabled in the “Main control” window, it is as if the interface card were not installed in the real machine. That means, for example, that the LX.683 EPROM firmware will boot SONE directly from floppy disk.

- If LX.683 is enabled in the “Main control” window but the image file is not mounted, it is as if the interface card were installed but the hard disk were not ready (for example because it is powered off). In that case, the LX.683 EPROM firmware will ask whether you want to boot from floppy disk, but it will not proceed until the hard disk becomes ready.

- If LX.683 is enabled in the “Main control” window and the image file is mounted but not formatted, then the hard disk is still not ready.

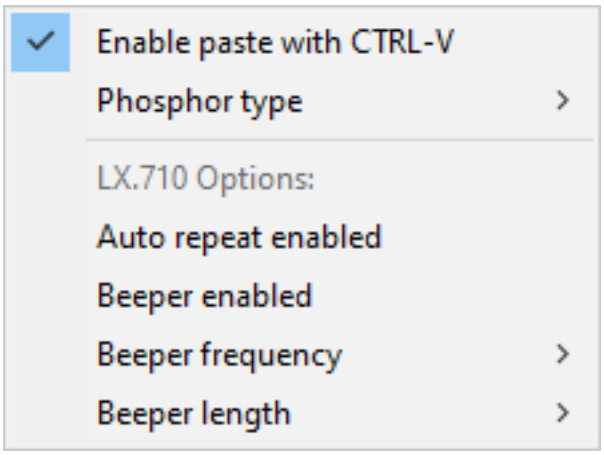
- If LX.683 is enabled in the “Main control” window and the image file is mounted and formatted, then the hard disk is ready. However, to boot SONE for example, you must first run HPUTSYS to load the IPL onto the disk; otherwise the LX.683 EPROM firmware will not be able to boot from the hard disk.

To format the hard-disk and install SONE on it, you can boot using the SONE_HD.DMK floppy disk image, which can be downloaded from here:

https://www.z80ne.com/software/sone/sone_hd.dmk

LX.710 – Auto-repeat add-on card

Right-click on the LX.388 or LX.529 screen to open a pop-up menu to configure the settings of the LX.710 auto-repeat add-on card.



LX.710 is an add-on card for the LX.388 and LX.529 keyboards that provides auto-repeat and beeper functions. Auto-repeat is useful when using CP/M, since NE-DOS already includes this function in the operating system.

With the pop-up menu you can:

- enable or disable auto repeat
- enable or disable beeper
- set the beep frequency to 250 Hz or 2200 Hz
- set the beep length to short or long.

Printer



This window emulates a printer connected either to the LX.389 printer interface or to the LX.529 printer output connector.

In this emulation there is no difference between the LX.389 printer interface and the LX.529 printer output connector: all commands/data sent to either are redirected to this printer.

Here is a brief description of the available options:

- Output/Video: select this option to print to the on-screen window.
- Output/File: select this option to print to a text file on your PC. When you select this option, a window is shown where you can type the filename to create. If you select an existing file, it is overwritten.
- Output/Printer: select this option to print to a real printer connected to your PC.
- Output/Select real printer: select this option to choose the real printer to print to. If a printer is already selected, it is displayed in this menu option.
  NOTE: all three printers can be selected at the same time.
- Command/Online: select this option to put the printer online and enable printing. While the printer is online, it is ready to receive and print data from the emulated CPU. This is equivalent to pressing the Online button on a real printer.

- Command/Offline: select this option to put the printer offline and temporarily disable printing. If a print operation is already in progress, it is paused. While the printer is offline, it signals to the emulated CPU that it is busy by setting the BUSY bit on the I/O port. If the Online option is selected again, printing resumes from the point where it was paused. This is equivalent to pressing the Offline button on a real printer.

- Command/Cut paper: select this option to clear the text printed on screen. It is equivalent to removing the paper from a real printer and starting on a new page.

- Commands/Clear buffer: this printer emulation is buffered. This means that data sent from the CPU to the virtual printer is first stored in a buffer before being printed. Clearing the buffer deletes all pending output.

- Commands/Copy to real printer: this function lets you send all text printed in the on-screen printer window to a real printer connected to your PC.

- Layout/80-132 columns: use these options to select the paper width.

- Layout/Continuous (“alternate green”, “alternate light blue”, “alternate grey”, “white”): use these options to select the type of continuous virtual paper shown on screen.

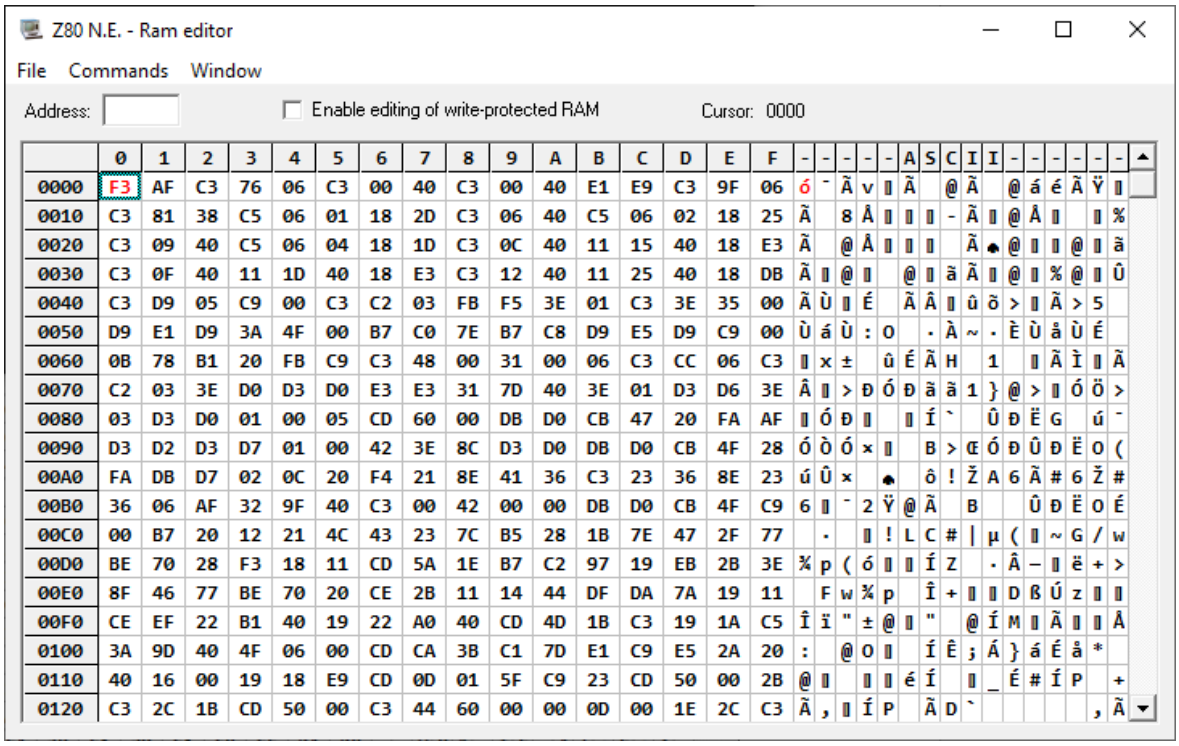
- Layout/A4 paper: use this option to select A4 paper for on-screen display.

- Options/Auto CR+LF: if this option is enabled, the printer automatically generates an LF (line-feed) character after receiving a CR (carriage-return). If it is not enabled, the software must send both CR and LF in order to return to the left margin and advance the paper by one line.

- Options/Turbo mode: use this option to print at the maximum speed allowed by the emulation, without simulating the speed of a real printer.
  NOTE: the software running on the CPU may be slow or contain delays, so printing can still be relatively slow even in Turbo mode.

- Options/Real printer type/WINDOWS - RAW: use this option to select how printing is sent to a real printer. WINDOWS mode prints through the Windows printer driver. RAW mode sends characters directly to the physical printer, without Windows interpreting them. RAW mode can be useful if you want to use ESC/P control codes directly.

Ram editor



This window is a RAM viewer/editor. It shows the contents of the CPU address space, both as hexadecimal values and as ASCII characters.

Here is a brief description of the available options:

- Menu File/Load ram: select this option to open Load ram window.
- Menu File/Save ram: select this option to open Save ram window.
- Menu Commands/Fill ram: select this option to open Fill ram window.
- Menu Commands/Copy ram: select this option to open Copy ram window.
- Menu Window/Open new window: select this option to open a new "Ram editor" window.
  You can open up to 16 windows. The list of open windows is shown below the "Open new window" menu item. Each window has a number displayed after its name for identification.
  The first available window is the one that receives position commands when you double-click the "Ram" labels in the "Main control" window.
- Address field: type a hexadecimal address to position the view.
- Enable editing of write-protected RAM: if the displayed RAM location is write-protected, enabling this option lets you change the RAM value anyway.
  NOTE: write-protected RAM locations have a gray background.

- Cursor label: this shows the currently selected address.

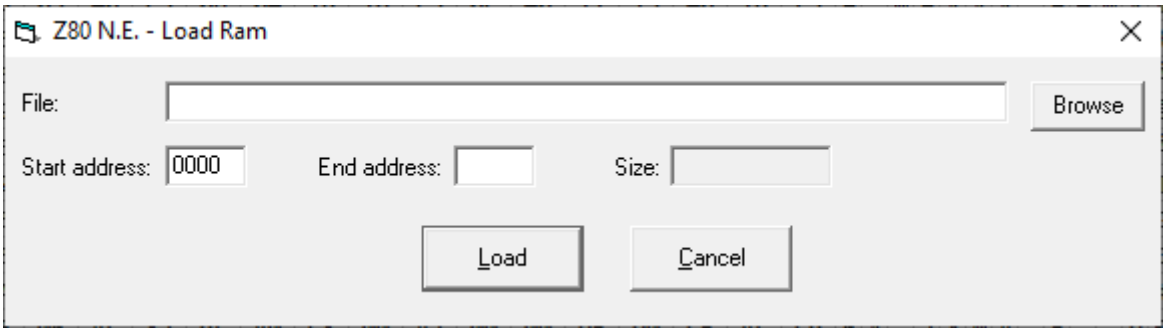
- Double-click a hexadecimal or ASCII RAM location with the left mouse button: double-click a location with the left mouse button to enter edit mode. Press ESC on the keyboard to leave edit mode without saving the value. Leaving edit mode in any other way saves the value.

- Double-click a hexadecimal or ASCII RAM location with the right mouse button: double-click a location with the right mouse button to write-protect or unprotect the selected RAM location.
  NOTE: this function is not a temporary write-protect override like the "Enable editing of write-protected RAM" checkbox. By right-clicking a RAM location, you are actually write-protecting or unprotecting that RAM location, so if it is write-protected, the CPU cannot write to it, just like an EPROM location.

- Select values: you can select multiple hexadecimal or ASCII values in three ways:
  - Drag the mouse while holding down the left mouse button.
  - Click while holding down the SHIFT key: this selects all values between the current cell and the clicked cell.
  - Click while holding down the CTRL key: this adds or removes the clicked cell from the current selection.

- CTRL+C/CTRL+V: you can copy and paste hexadecimal or ASCII values by pressing CTRL+C or CTRL+V.

Load ram



This window lets you load RAM from a binary file.

Here is a brief description of the available options:

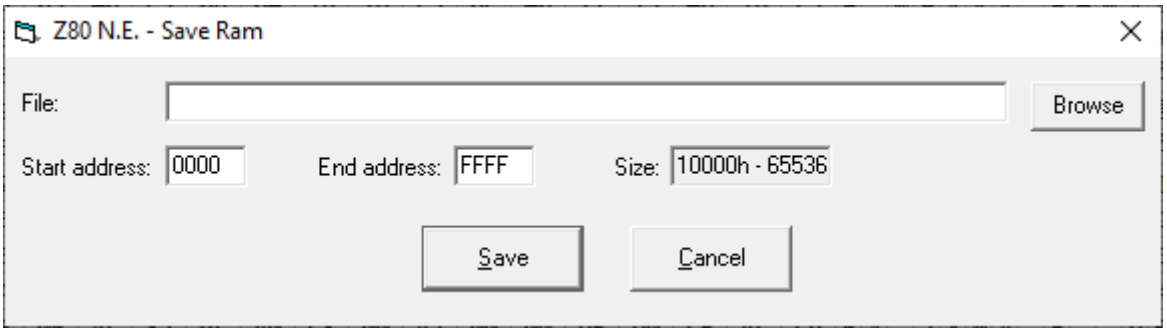
- File field: type the path and filename of the .RAM binary file to be loaded in this field.
- Browse button: click this button to select a .RAM binary file from your PC.
- Start address field: this is the starting address of the RAM address range to be loaded.
- End address field: this is the ending address of the RAM address range to be loaded. It is automatically filled in when you select a file to load. You can edit it as required.
- Size label: this is the size of the RAM address range to be loaded. It is displayed in both hexadecimal and decimal format.
- Load button: click this button to load the selected file into the selected RAM range.
- Cancel button: click this button to close this window.

NOTE: the file is loaded only up to address FFFFh, even if its size would extend beyond it.

NOTE: the file to be loaded is treated as a raw binary file without any headers.

NOTE: typically, you would load binary files created with the Save ram menu option; however, any file can be loaded.

Save ram



This window lets you save RAM to a binary file.

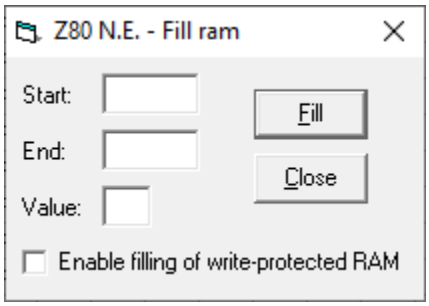
Here is a brief description of the available options:

- File field: type the path and filename of the .RAM binary file to be saved in this field.
- Browse button: click this button to select a folder on your PC and enter the filename of the .RAM binary file to be written.
- Start address field: this is the starting address of the RAM address range to be saved. It is automatically set to 0000h when you open this window. You can edit it as required.
- End address field: this is the ending address of the RAM address range to be saved. It is automatically set to FFFFh when you open this window. You can edit it as required.
- Size label: this is the size of the RAM address range to be saved. It is displayed in both hexadecimal and decimal format.
- Save button: click this button to save the selected RAM range to the selected file.
- Cancel button: click this button to close this window.

NOTE: the file saved is a raw binary file without any headers.

NOTE: saved files can be loaded using the Load ram menu option.

Fill ram

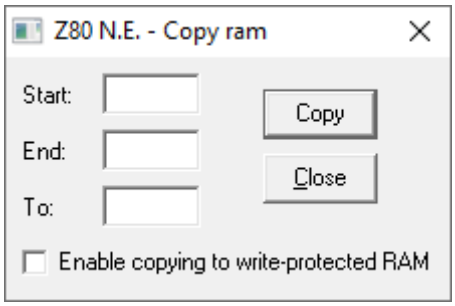


This window lets you fill RAM with value.

Here is a brief description of the available options:

- Start field: this is the starting address of the address space to fill.
- End field: this is the ending address of the address space to fill.
- Value: this is the value to write into RAM.
- Enable filling of write-protected ram: enabling this option lets you write the value into RAM even if a RAM address is write-protected.
- Fill button: click this button to fill RAM.
- Close button: click this button to close this window.

Copy ram

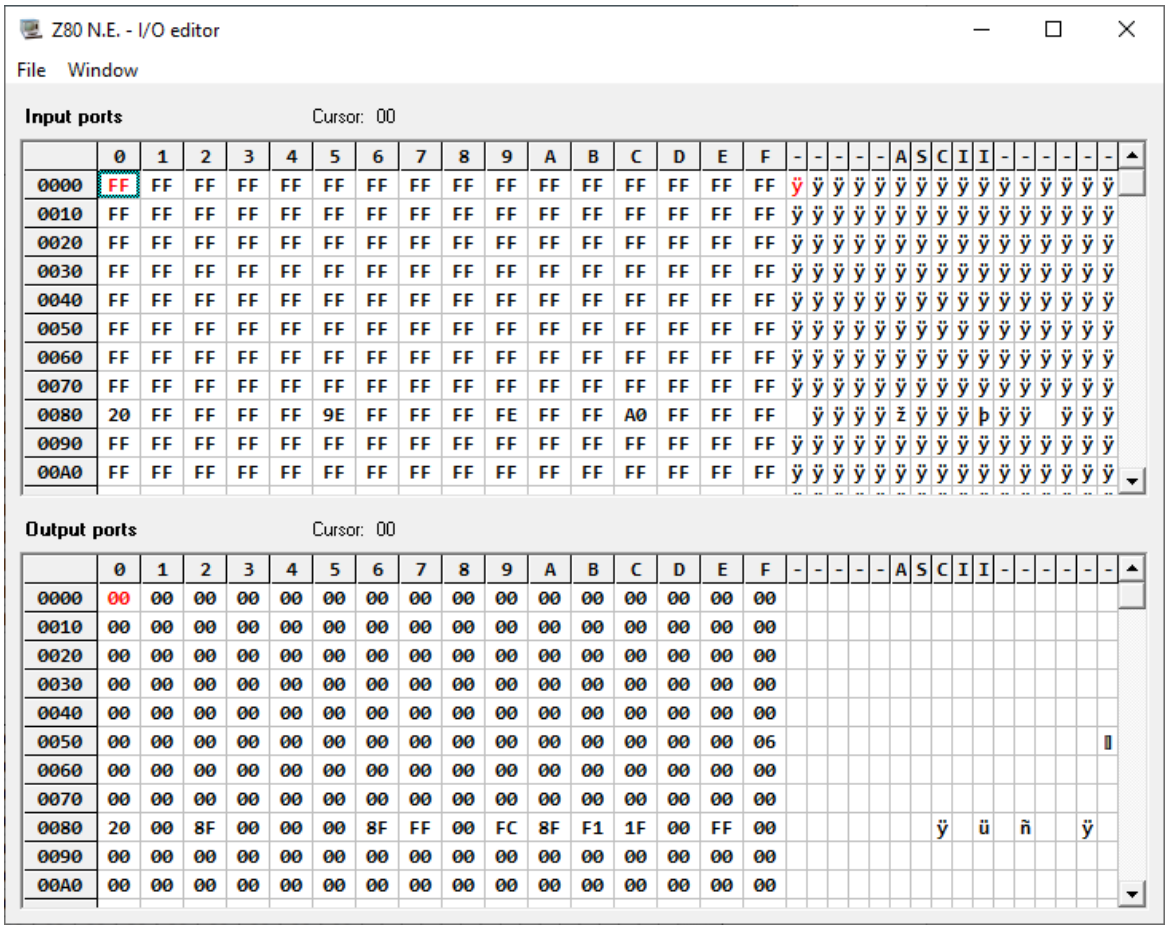


This window lets you copy RAM.

Here is a brief description of the available options:

- Start field: this is the starting address of the address range to be copied.
- End field: this is the ending address of the address range to be copied.
- To field: this is the destination address to which the selected address range will be copied.
- Enable copying of write-protected ram: enabling this option lets you copy data into RAM even if the destination address is write-protected.
- Copy button: click this button to copy the selected RAM range.
- Close button: click this button to close this window.

I/O editor



This window is a viewer/editor of the INPUT/OUTPUT ports. It shows the contents of the CPU input/output ports, both as hexadecimal values and as ASCII characters.

Here is a brief description of the available options:

- Menu File/Load input ports - Load output ports: select this option to open Load input ports or Load output ports window.

- Menu File/Save input ports – Save output ports: select this option to open Save input ports or Save output ports window.

- Menu Window/Open new window: select this option to open a new "I/O editor" window.
  You can open up to 16 windows. The list of open windows is shown below the "Open new window" menu item. Each window has a number displayed after its name for identification.
  The first available window is the one that receives position commands when you double-click the register labels in the "Main control" window.

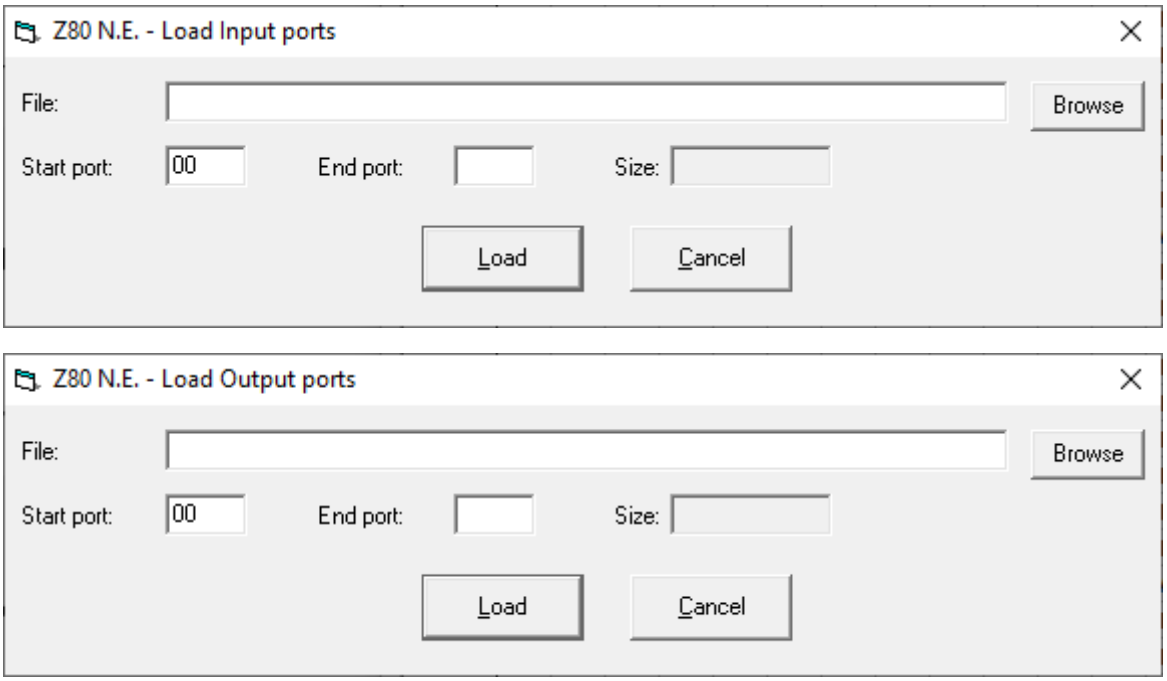
- Cursor label: this shows the currently selected address.

- Double-click a hexadecimal or ASCII I/O location: double-click a location to enter edit mode. Press ESC on your keyboard to leave edit mode without saving the value. Any other way of leaving edit mode saves the value.

- Select values: you can select multiple hexadecimal or ASCII values in three ways:
  - Drag the mouse while holding down the left mouse button.
  - Click while holding down the SHIFT key: this selects all values between the current cell and the clicked cell.
  - Click while holding down the CTRL key: this adds or removes the clicked cell from the current selection.

- CTRL+C/CTRL+V: you can copy and paste hexadecimal or ASCII values by pressing CTRL+C or CTRL+V.

Load input ports – Load output ports



These windows let you load I/O input or output ports from a binary file.

Here is a brief description of the available options:

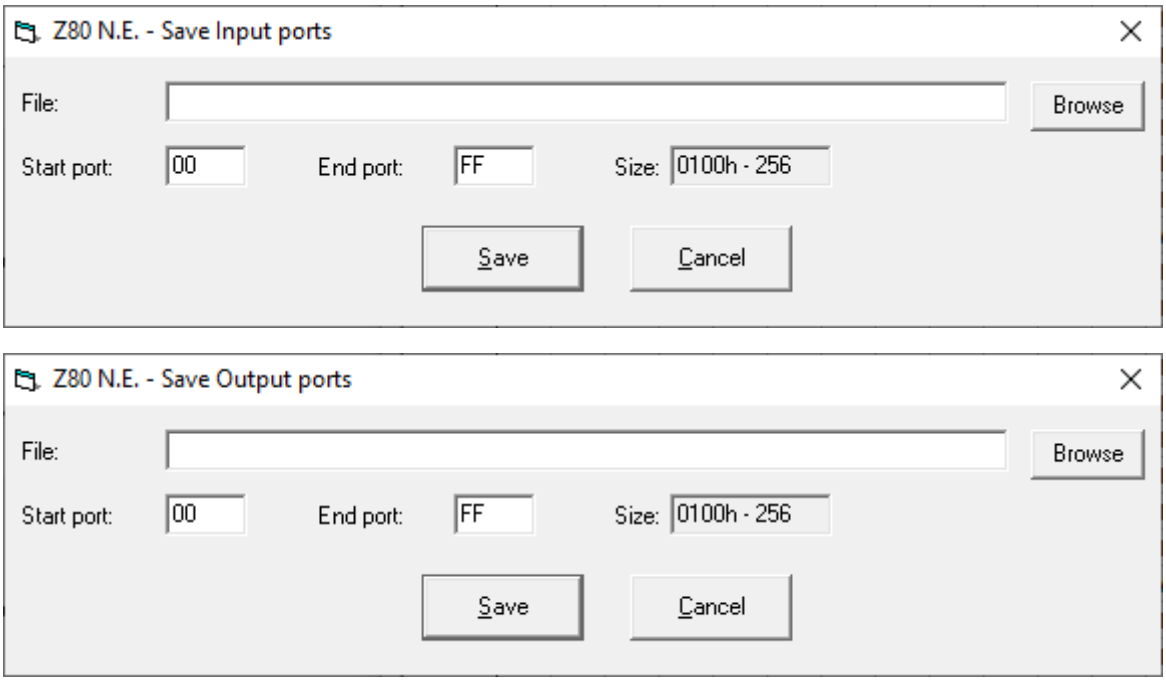
- File field: type the path and filename of the I/O port binary file to be loaded in this field.
- Browse button: click this button to select an I/O port binary file from your PC.
- Start port field: this is the starting address of the I/O port range to be loaded.
- End port field: this is the ending address of the I/O port range to be loaded. It is automatically filled in when you select a file to load. You can edit it as required.
- Size label: this is the size of the I/O port range to be loaded. It is displayed in both hexadecimal and decimal format.
- Load button: click this button to load the selected file into the selected I/O port range.
- Cancel button: click this button to close this window.

NOTE: the file is loaded only up to address FFh, even if its size would extend beyond it.

NOTE: the file to be loaded is treated as a raw binary file without any headers.

NOTE: typically, you would load binary files created with the Save I/O input ports or Save I/O output ports menu options; however, any file can be loaded.

Save input ports – Save output ports



These windows let you save I/O input or output ports to a binary file.

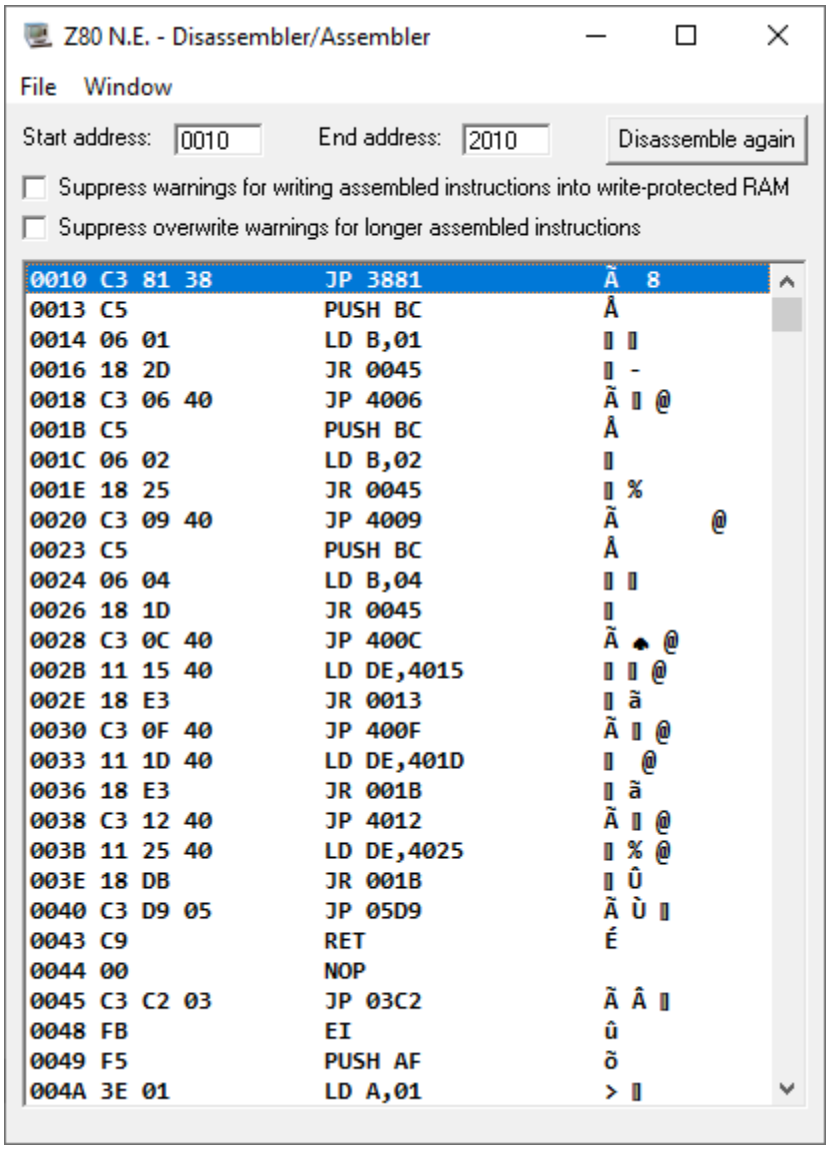
Here is a brief description of the available options:

- File field: type the path and filename of the I/O port binary file to be saved in this field.
- Browse button: click this button to select a folder on your PC and enter the filename of the I/O port binary file to be written.
- Start address field: this is the starting address of the I/O port range to be saved. It is automatically set to 00h when you open this window. You can edit it as required.
- End address field: this is the ending address of the I/O port range to be saved. It is automatically set to FFh when you open this window. You can edit it as required.
- Size label: this is the size of the I/O port range to be saved. It is displayed in both hexadecimal and decimal format.
- Save button: click this button to save the selected I/O port range to the selected file.
- Cancel button: click this button to close this window.

NOTE: the saved file is a raw binary file without any headers.

NOTE: saved files can be loaded using the Load I/O input ports or Load I/O output ports menu options.

Disassembler/Assembler



This window is a disassembler/assembler. It shows the CPU instructions currently present in RAM in disassembled form. You can double-click an instruction to enter assembly mode.

The list contains:

- RAM address
- Instruction opcode
- Disassembled instruction mnemonic
- ASCII conversion of the opcode byte values

Here is a brief description of the available options:

- File/Save: select this option to save the disassembled listing to a file on your PC.

- Window/Open new window: select this option to open a new “Disassembler/Assembler” window.
  You can open up to 16 windows. The list of open windows is shown below the “Open new window” menu item. Each window has a number displayed after its name for identification.
  The first available window is the one that receives position commands when you click the “Step” button in the “Main control” window.

- Start address field: this is the starting address for disassembly.

- End address field: this is the ending address for disassembly.

- Disassemble again button: click this button to force a new disassembly.

- Suppress warning for writing assembled instructions into write-protected RAM: select this checkbox to suppress warnings when assembling instructions that will be written to a write-protected RAM address range.

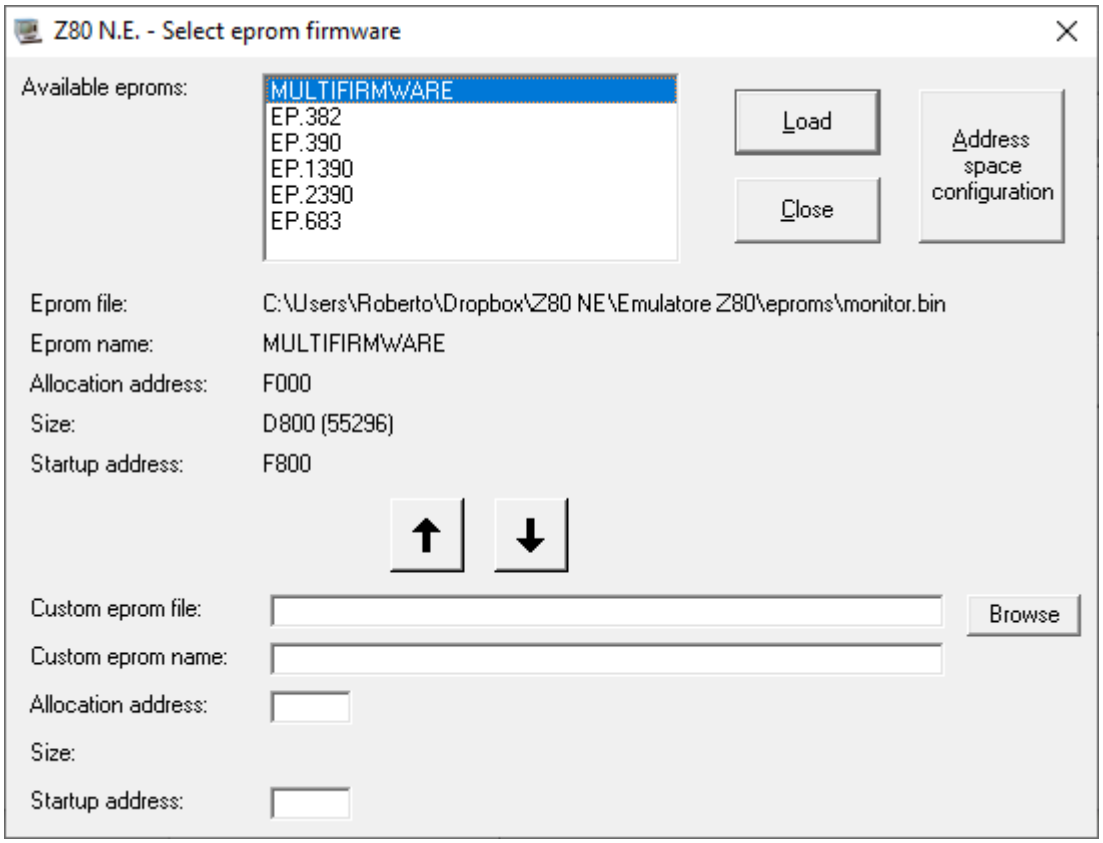
- Suppress overwrite warnings for longer assembled instructions: select this checkbox to suppress warnings when an assembled instruction is longer than the instruction it replaces and therefore overwrites the bytes of the following instruction.

- Double-click a line: double-click a line to enter assembly mode. In assembly mode, you can type a mnemonic instruction, which is then assembled and written to RAM. Press ENTER to assemble the instruction and write it to RAM. Press ESC to exit assembly mode.

- Right-click a line: right-click a line to open a menu where you can choose an option to add the corresponding address to the “Breakpoint” list in the “Main control” window. This allows you to quickly set a breakpoint.

NOTE: when the “Main control” window is in step-by-step mode, the disassembler/assembler is synchronized with the Program Counter address. This means that whenever you press the “Step” button, the highlighted disassembly line moves to the current Program Counter address. If the address falls outside the Start and End range, disassembly is performed automatically with Start address = PC address minus 1000h, and End address = PC address plus 1000h.

Select eprom firmware



This window lets you choose which EPROM to use to boot the virtual CPU. You can also add custom EPROMs to the list of available EPROMs.

Here is a brief description of the available options:

- Available EPROMs list: to select an EPROM, click the desired EPROM and then click the “Load” button, or simply double-click the desired EPROM.
- UP arrow button: click this button to add the custom EPROM data to the list of available EPROMs. This addition is saved immediately.
- DOWN arrow button: click this button to remove the EPROM from the list and place it back into the custom EPROM data fields. This removal is saved immediately.
- Custom EPROM file: type the path and filename of the custom EPROM file in this field.
- Browse button: click this button to select a custom EPROM file from your PC.
- Custom EPROM name: type a custom EPROM name of your choice in this field. This name will be displayed in the “Main control” window when this EPROM is in use.

- Allocation address: type a hexadecimal address. This is the address at which the custom EPROM file is loaded.

- Startup address: type a hexadecimal address. This is the address from which the CPU starts after a reset. It may differ from the allocation address.

- Load button: click this button to load the selected EPROM into RAM. The address space will be protected according to the Address space configuration defined for the selected EPROM.

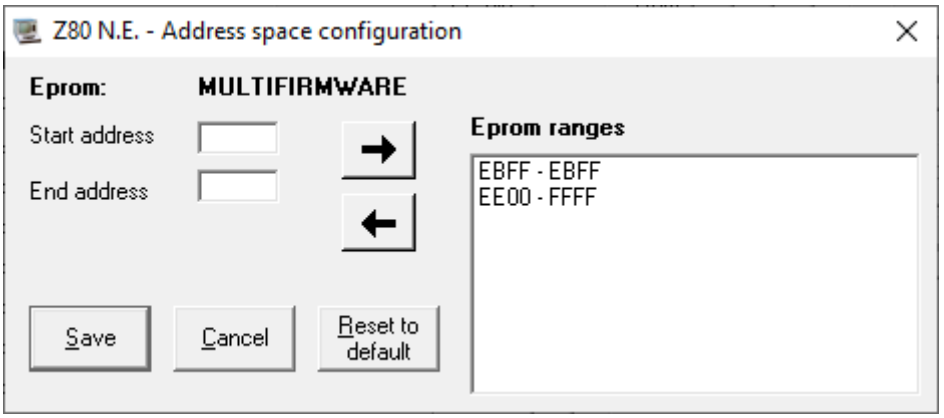
- Close button: click this button to exit without changing the loaded EPROM.

- Address space configuration button: click this button to open the Address space configuration window for the selected EPROM.

NOTE: custom EPROM files must be raw binary files.

NOTE: the address space used by the loaded EPROM, whether factory or custom, is write-protected by default. You can change its protection settings by using the “Options/Address space configuration” menu in the “Main control” window or the “Address space configuration” button. Each EPROM has its own address space configuration.

Address space configuration



This window lets you edit the write-protected address space, usually occupied by EPROMs.

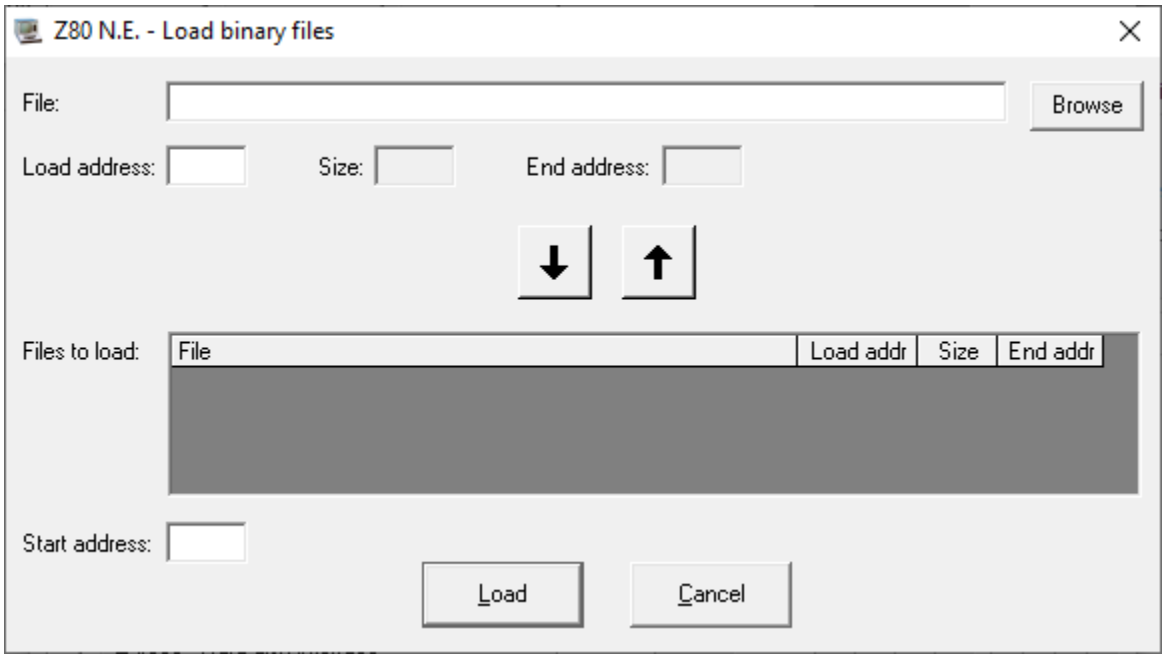
The write-protected address space is set up when an EPROM is loaded. These address ranges cannot be written to by CPU instructions. However, you can change them to protect or unprotect address ranges as desired.

The address space can be different for each available EPROM and can be configured independently.

Here is a brief description of the available options:

- Start address: type a hexadecimal address. This is the starting address of the range to protect from writing.
- End address: type a hexadecimal address. This is the ending address of the range to protect from writing.
- RIGHT arrow button: click this button to add the address range to the list on the right.
- LEFT arrow button: click this button to remove the address range from the list and place it back into the address input fields.
- Save button: click this button to save all changes. The address ranges listed on the right become write-protected.
- Cancel button: click this button to discard all changes.
- Reset to default button: click this button to restore the address space configuration of an EPROM to its default values. For custom EPROMs, the default value is the address space occupied by the EPROM.

Load binary files



This window lets you load one or more raw binary files into RAM.

Here is a brief description of the available options:

- File text field: type the path and filename of the file to be added to the “Files to load” list.
- Browse: click this button to locate a raw binary file to be loaded.
- Load address: type a hexadecimal address. This is the address at which the binary file will be loaded.
- Size label: this is the size of the binary file.
- End address label: this is the last address occupied by the binary file when it is loaded.
- DOWN arrow button: click this button to add the file to the “Files to load” list.
- UP arrow button: click this button to remove the selected file from the “Files to load” list and place it back into the input fields.
- Files to load list: this is the list of binary files to be loaded.
- Start address: type a hexadecimal address. This is the address at which execution will start. When the emulator is in the STOP state, this value is loaded into the PC field of the “Main control” window. Leave this field empty if you do not want execution to start.

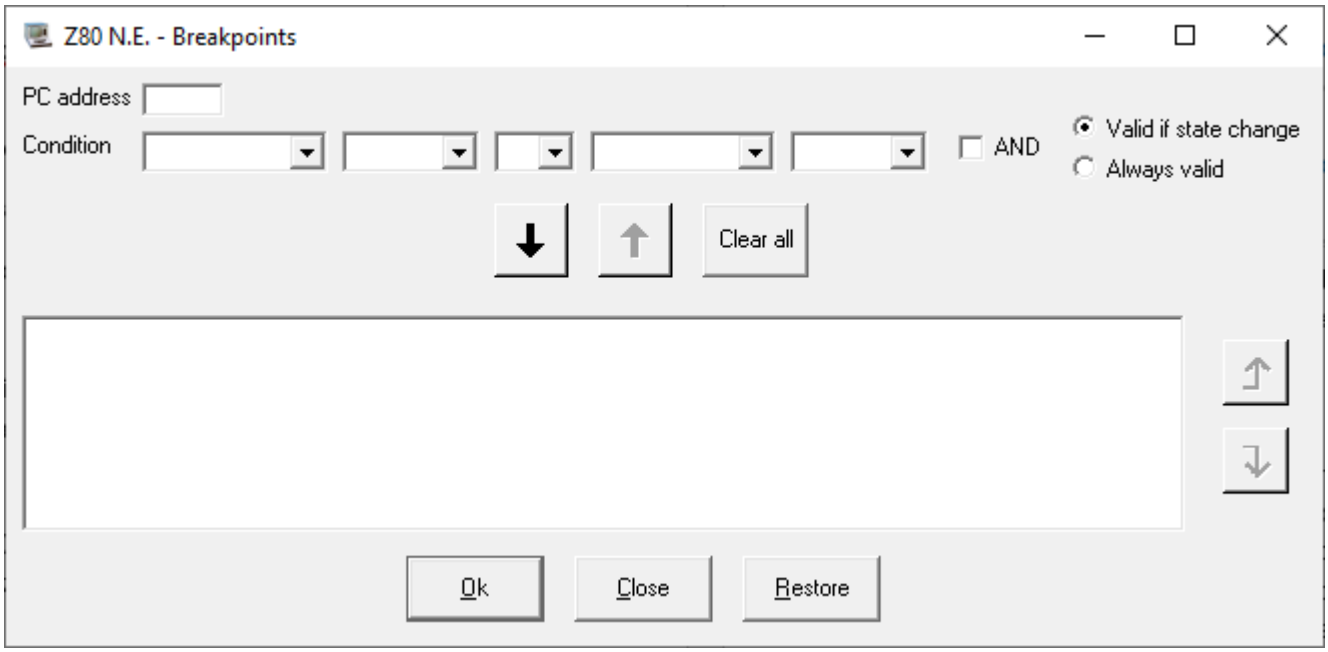
- Load button: click this button to load the files in the “Files to load” list. If “Start address” is not empty and the emulator is in the running state, execution starts immediately.

- Cancel button: click this button to exit without loading any files.

NOTE: no checks are performed for overlapping addresses. Files are loaded in the order in which they are displayed (sorted by load address).

NOTE: the address space used by the loaded files is not initialized when you press the “Run” button in the “Main control” window.

Breakpoints



This window lets you configure advanced breakpoint conditions.

You can add multiple conditions, which are evaluated from first to last. If a condition is true, step-by-step mode is automatically activated during program execution.

Here is a brief description of the available options:

- PC address: enter the hexadecimal Program Counter address of the breakpoint.
- Condition fields: select or enter values in each field to define a condition that must be matched to trigger the breakpoint. The fields are:
    - 1st field options:
        - Register: compare a CPU register
        - Ram: compare a value at a RAM address
        - I/O_port_IN: compare a byte value at an input I/O port
        - I/O_port_OUT: compare a byte value at an output I/O port
        - Int.mode: compare the interrupt mode
        - Int.flag: compare an interrupt flag
        - Int.active: compare the active interrupt state
        - StackDepth: compare the emulator Stack Depth value

- 2nd field options: depending on the value selected in the 1st field, this field may contain one of the following values:
  - A, F, B, C, D, E, H, L, I, R, IX, IY, SP: the corresponding CPU register
  - BC, DE, HL: the corresponding CPU register pair
  - A', F', B', C', D', E', H', L': the corresponding alternate CPU register
  - BC', DE', HL': the corresponding alternate CPU register pair
  - PC: the PC register value of the current instruction to be executed
  - PC_Old: the PC register value of the last executed instruction
  - PC_Next: the PC register value of the next instruction
  - IFF1, IFF2: interrupt flag IFF1 or IFF2
  - NMI, INT: the active interrupt type
  - value field: a field in which you can enter a value, for example a RAM address

- 3rd field options: this is the comparison operator field. The available values are:
  - =: equal to
  - <>: not equal to
  - >: greater than
  - <: less than
  - >=: greater than or equal to
  - <=: less than or equal to

- 4th field options: this field contains the same options as the 1st field, plus:
  - Value: a value to be entered in the 5th field

- 5th field options: this field contains the same options as the 2nd field.

- AND checkbox: if this checkbox is selected, the next condition must also be true for the breakpoint to be triggered. You can add as many AND conditions as required. Conditions not linked by the AND flag are treated as OR conditions.

- Valid if state change/Always valid options: when "Valid if state change" is selected, a condition that has triggered a breakpoint must become false before it can trigger another breakpoint.
  For example, if the condition is "Register A = Value 10", the breakpoint is triggered when register A becomes 10. After execution resumes, the breakpoint is not triggered again until register A first becomes different from 10, making the condition false, and then becomes equal to 10 again.
  If "Always valid" is selected, the condition is evaluated immediately after execution resumes. If it is still true, another breakpoint is triggered immediately.

- DOWN arrow button: click this button to add a condition to the list below.

- UP arrow button: click this button to remove a condition from the list and place it back into the Condition fields.

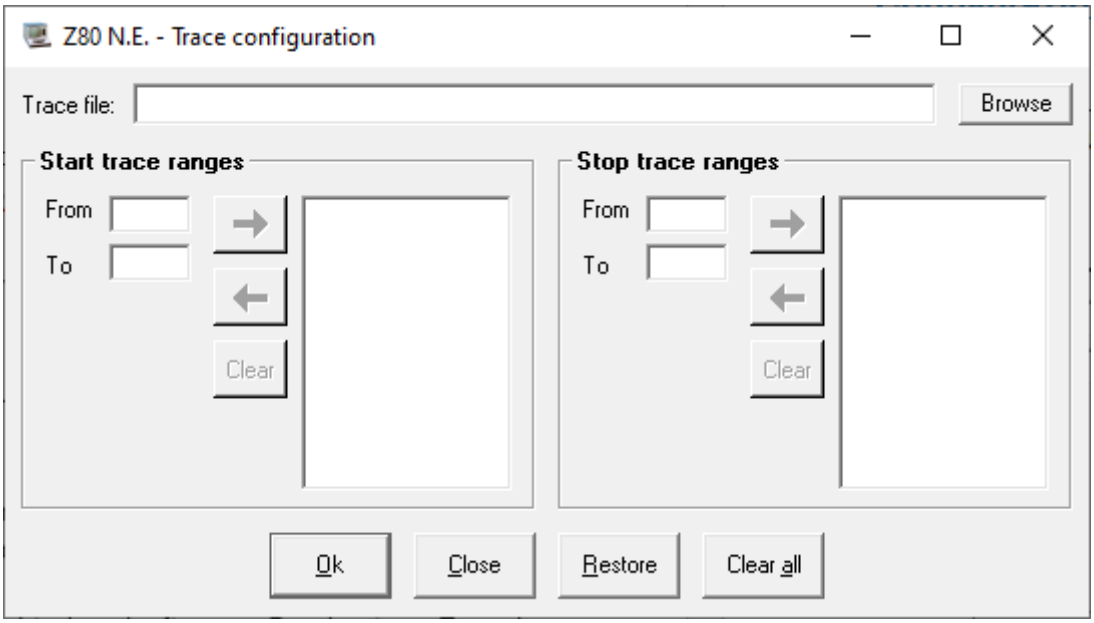
- Clear all: click this button to remove all conditions from the list and clear all Condition fields.

- List field: this list contains all the entered conditions. Double click a row to remove it from the list and place it back into the condition fields.
- RIGHT-UP arrow button: click this button to move the selected list row up by one position.
- RIGHT-DOWN arrow button: click this button to move the selected list row down by one position.
- Ok button: click this button to save the conditions. The entered conditions are not evaluated until this button is pressed.
- Close button: click this button to close the Breakpoints window. Any unsaved changes will be discarded. To save them, press the Ok button before closing the window.
- Restore button: click this button to restore the conditions to the values they had before the Ok button was last pressed.

NOTE: when a breakpoint is triggered during execution, the condition that triggered it is automatically selected in the condition list.

NOTE: when a breakpoint is triggered during execution, click the Main control window to select it, then move the mouse over the "Breakpoints" button to display a tooltip showing the condition that triggered the breakpoint and the number of CPU T-States elapsed since the previous breakpoint, or since the RUN button was pressed if this is the first breakpoint triggered after starting the CPU.

Trace configuration



This window lets you configure the "Trace to file" options.

You can add ranges that determine when tracing starts and stops. When the PC register value falls within a Start range, trace data starts being written to the trace file.

When the PC register value falls within a Stop range, trace data is no longer written to the trace file until the PC falls within a Start range again.

You can add as many ranges as required. You can also enter only the From address; in this case, the range is valid only when the PC exactly matches that address.

In all cases, you must select the "Trace to file" checkbox in the Main control window to start tracing according to the configured trace ranges.
To stop tracing and save the collected data to the final trace file, clear the "Trace to file" checkbox.

NOTE: during program execution, trace data is initially written to an emulator memory buffer. When the memory buffer is full, its contents are flushed to a temporary trace file in an internal binary format. When you clear the "Trace to file" checkbox to stop tracing, the temporary trace file is read, and its internal binary data is converted and saved to the final trace file.
Please be aware that this operation may take a long time. Always check the number of bytes written to the temporary file, which is displayed in the Main control window below the "Trace to file" checkbox. Use the Start and Stop trace ranges to limit the amount of recorded data to the addresses of interest.

Here is a brief description of the available options:

- Trace file: enter the path and filename of the trace file.
  WARNING: the trace file is overwritten each time you select the "Trace to file" checkbox to start tracing.

- Browse button: click this button to select a folder on your PC and enter the filename of the trace file to be written. The trace file can be saved in .TXT or .CSV (comma-delimited) format.

- Start trace ranges fields:
  - From: enter the hexadecimal start address of the range
  - To: enter the hexadecimal end address of the range
  - RIGHT arrow button: click this button to add the range to the list
  - LEFT arrow button: click this button to remove the range from the list and place it back into the From and To fields
  - Clear button: click this button to remove all ranges from the list
  - List field: this list contains all the entered ranges. Double click a row to remove it from the list and place it back into the From and To fields.

- Stop trace ranges fields: these fields work in the same way as the Start trace ranges fields.

- Ok button: click this button to save the ranges. The entered ranges are not used until this button is pressed.

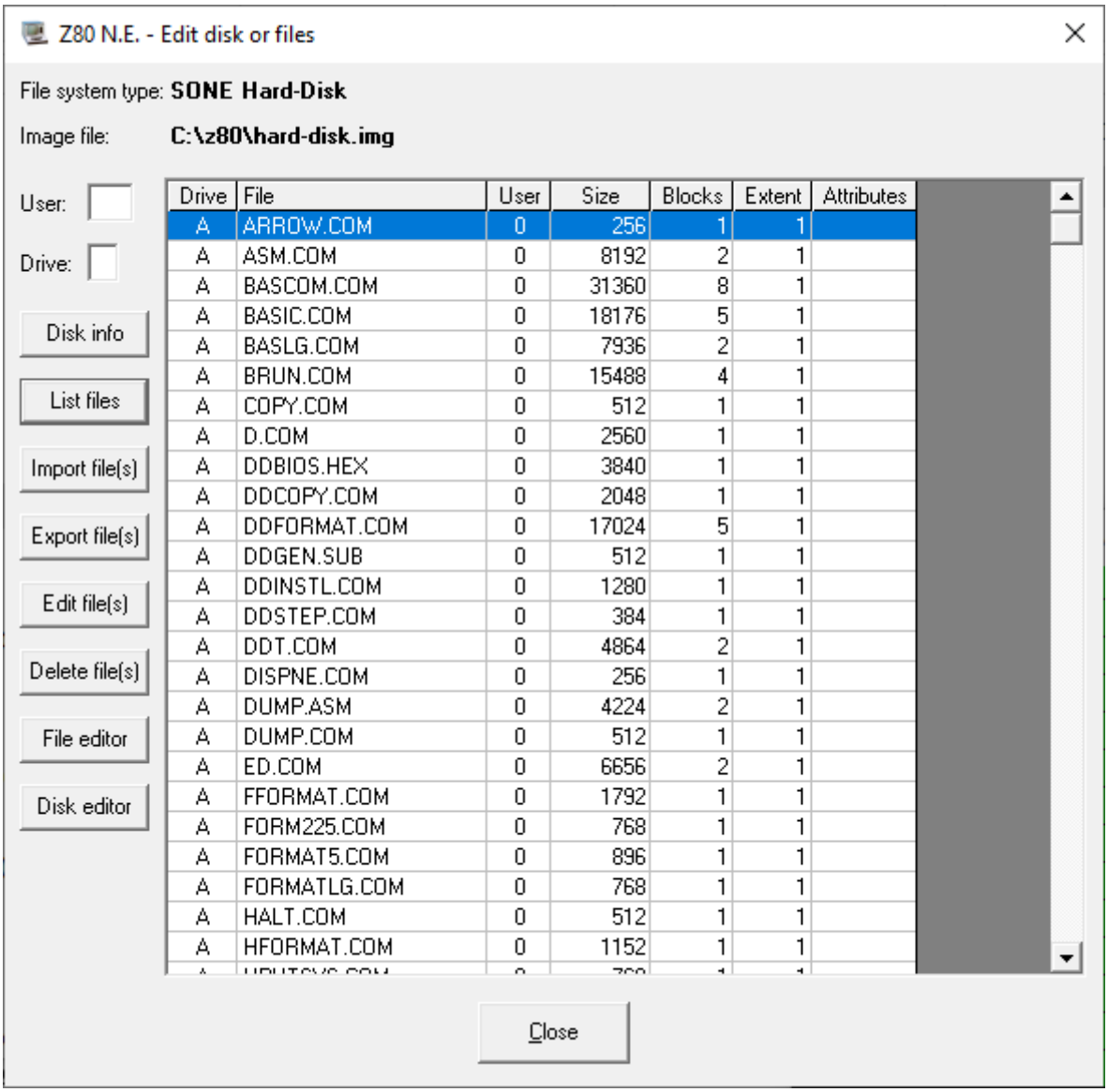
- Close button: click this button to close the Trace configuration window. Any unsaved changes will be discarded. To save them, press the Ok button before closing the window.

- Restore button: click this button to restore the ranges to the values they had before the Ok button was last pressed.

- Clear all button: click this button to remove all ranges from the lists and clear the Trace file field.

NOTE: if you clear all fields, tracing is disabled.
If you enter only the trace filename without defining any ranges, tracing is active for all PC addresses.
If you define only Start ranges, tracing never stops automatically. You must stop it manually by clearing the "Trace to file" checkbox in the Main control window.
If you define only Stop ranges, tracing starts immediately when you select the "Trace to file" checkbox in the Main control window.

Edit disk or files



This window lets you manage files stored in a disk image or on a physical floppy disk.

You can edit filenames or attributes and file contents, and you can also edit the entire disk contents.

The following file systems are supported, using .IMD or .DMK image files or physical floppy disks:

- NE-DOS 40 tracks
- NE-DOS 80 tracks
- SONE SSSD, 10 sectors × 256 bytes
- SONE DSSD, 10 sectors × 256 bytes
- SONE SSSD, 17 sectors × 128 bytes
- CP/M SSSD, 18 sectors × 128 bytes
- CP/M DSSD, 18 sectors × 128 bytes (40+40 tracks)

- CP/M SSDD, 10 sectors × 512 bytes (.IMD or .DMK image files only, read-only)
- CP/M DSDD, 10 sectors × 512 bytes (.IMD or .DMK image files only, read-only)
- Micro Design CP/M 8", DSDD, 16 sectors × 512 bytes (.IMD or .DMK image files only, read-only)
- Micro Design CP/M SSSD, 17 sectors × 128 bytes (80 tracks)
- SONE Hard-Disk (.IMG image files only)

Legend:

- SSSD: Single-Sided, Single-Density
- DSSD: Double-Sided, Single-Density
- SSDD: Single-Sided, Double-Density
- DSDD: Double-Sided, Double-Density

Here is a brief description of the available options:

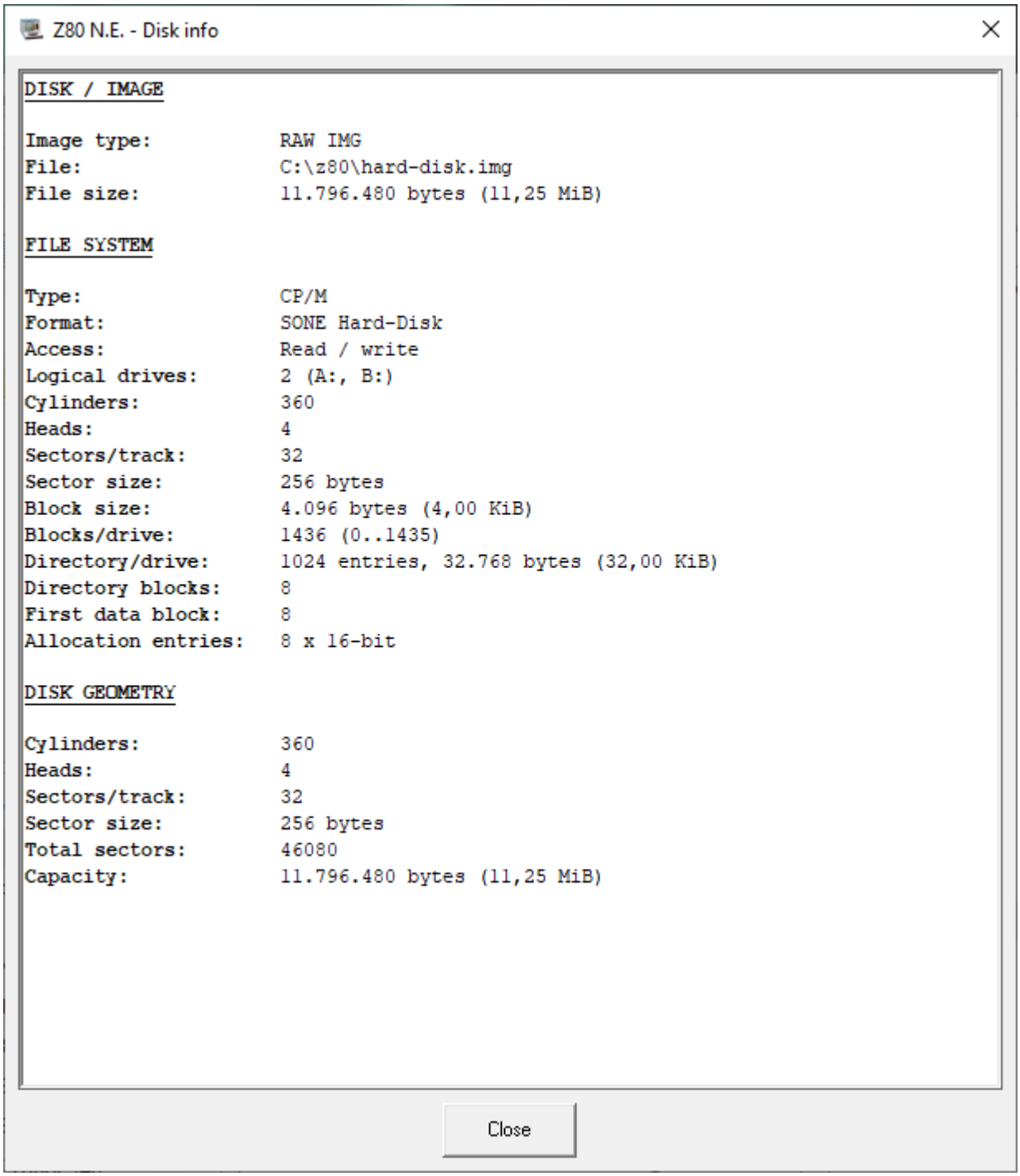
- File system type label: this is the type of file system detected.
- Image file label: this is the path and filename of the image file or the physical disk drive.
- User field: type the user number to filter by that user (SONE only). Leave this field blank to show all users.
- Drive field: type the drive letter, “A” or “B”, to filter by that drive (SONE only). Leave this field blank to show all drives.
- Disk info button: click this button to open the Disk info window, which displays information about the disk.
- List files button: click this button after changing the User or Drive field to update the file list.
- Import file(s) button: click this button to import one or more files from your PC into the image file or the physical disk. If you click this button, the Assign file names window is displayed.
- Export file(s) button: click this button to export one or more files from the image file or the physical disk to your PC. If you click this button, the Assign file names window is displayed.
- Edit file(s) button: click this button to edit one or more filenames or file attributes in the image file or the physical disk. If you click on this button, the Edit file window is displayed.
- Delete file(s) button: click this button to delete one or more files from the image file.
- File editor: click this button to open the file editor for editing the currently selected file. If you click this button, the File editor window is displayed.

- Disk editor: click this button to open the disk editor for editing the entire disk content. If you click this button, the Disk editor window is displayed.

- Close button: click this button to close this window.

- Right mouse click: click with the right mouse button to open a context menu containing the same commands available from the buttons on the left.

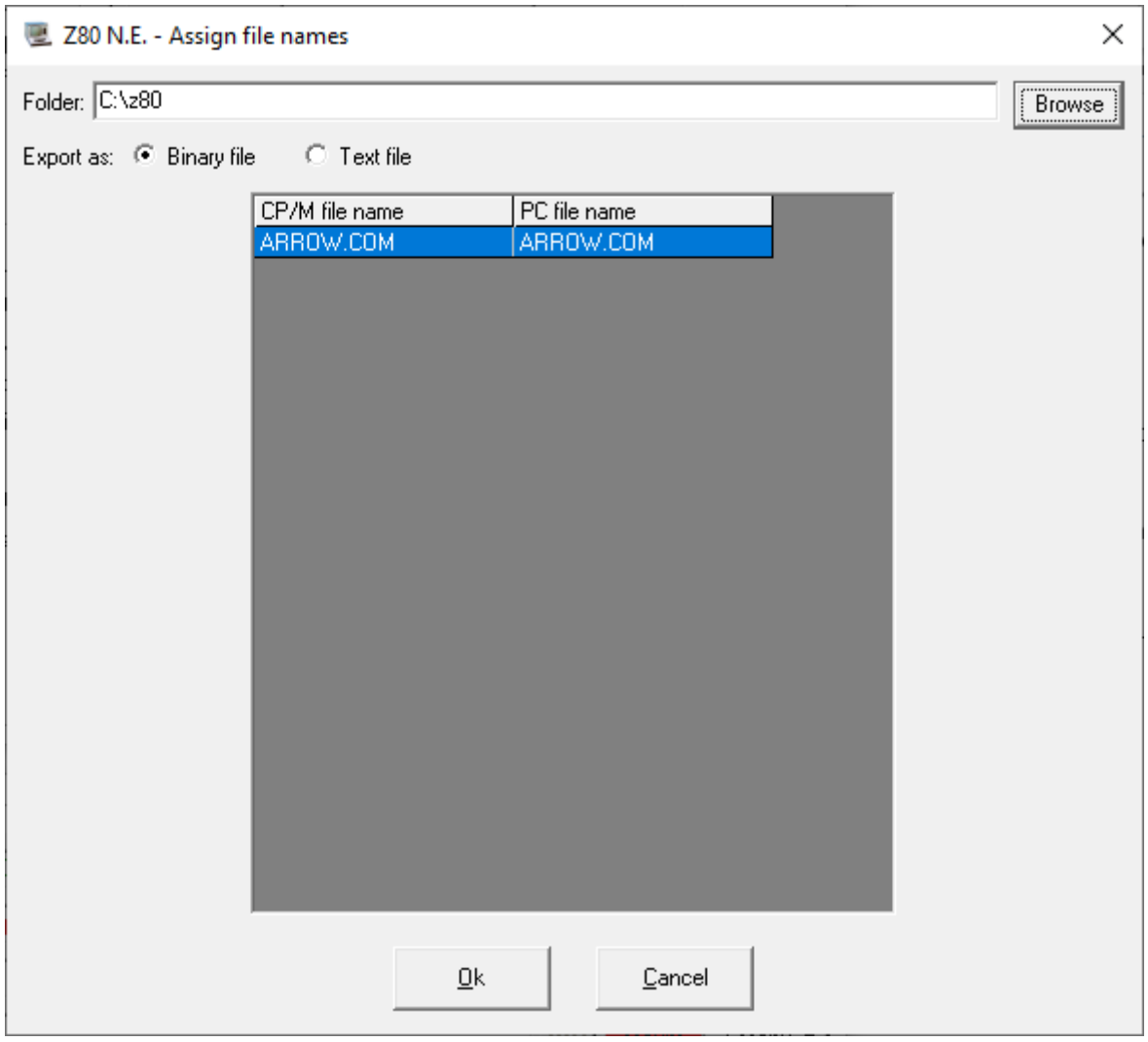
Disk info



This window displays information about the selected disk.

Assign file names

If you click the Import or Export button in the Edit disk or files, the “Assign file names” window is displayed.



Use this window to check or change the folder and destination filename for each file to be imported or exported.

Here is a brief description of the available options:

- Folder field: type the name of the folder on your PC where you want to export files. This field is editable only for export operations.

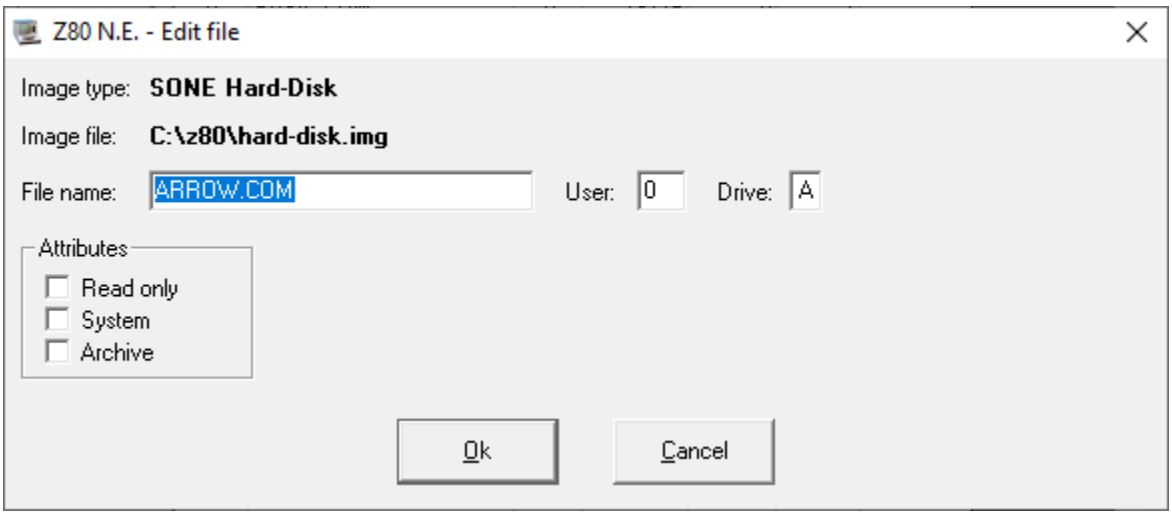
- Browse button: click this button to select a folder on your PC. This button is enabled only for export operations.

- Export as: select how you want to export files (SONE only).
  SONE treats files as using a 128-byte block size.
  If “Binary file” is selected, files are exported without any changes and aligned to the 128-byte block size. This means that if a file does not end on a 128-byte boundary, 1Ah byte values are added to the end of the file to reach that boundary.
  If “Text file” is selected, the exported file is truncated at the first occurrence of the 1Ah character (CTRL+Z, which means EOF). This is useful for exporting text files without any garbage at the end.

- File list: this is the list of source and destination filenames. To change a filename, double-click the relevant name or type directly in the relevant cell.
  Press ENTER to confirm the filename change, or press ESC to cancel edit mode.

Edit file

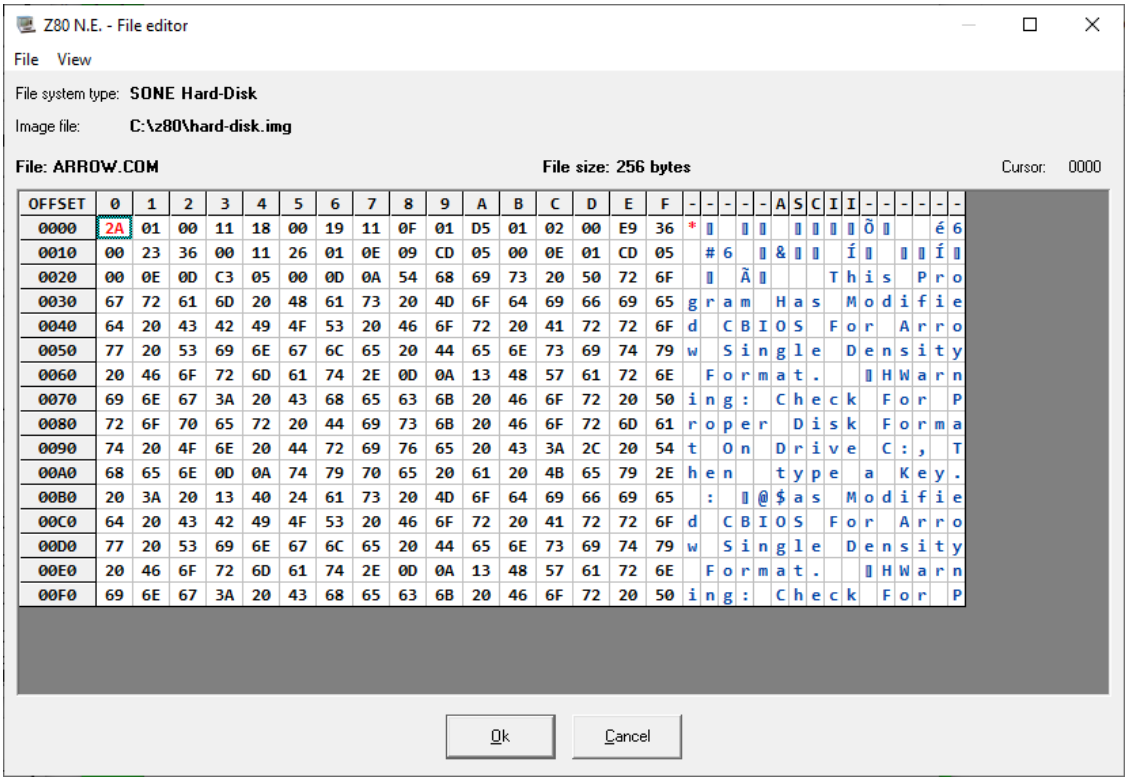
If you click on the Edit file(s) button in the Edit disk or files window, the “Edit file” window is displayed.

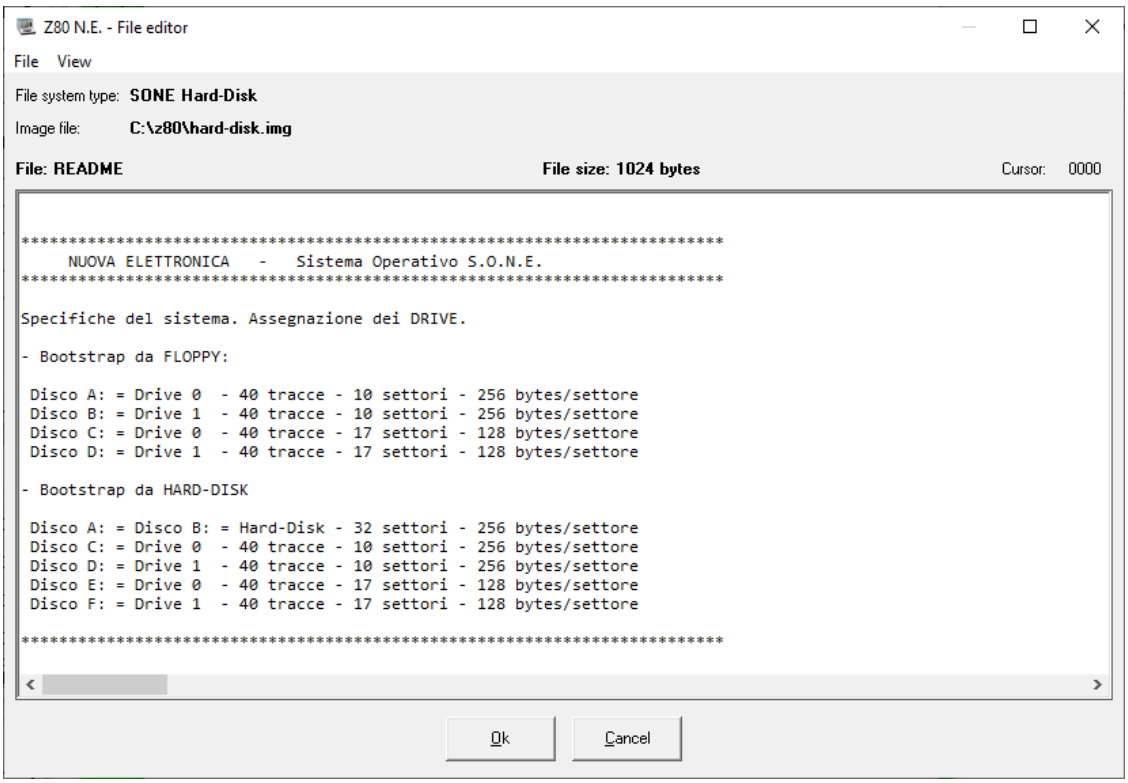


Here you can change the filename, user number (SONE only), drive letter for the file (SONE only), and/or file attributes.

File editor

Clicking the File editor button in the Edit disk or files window displays the “File editor” window.





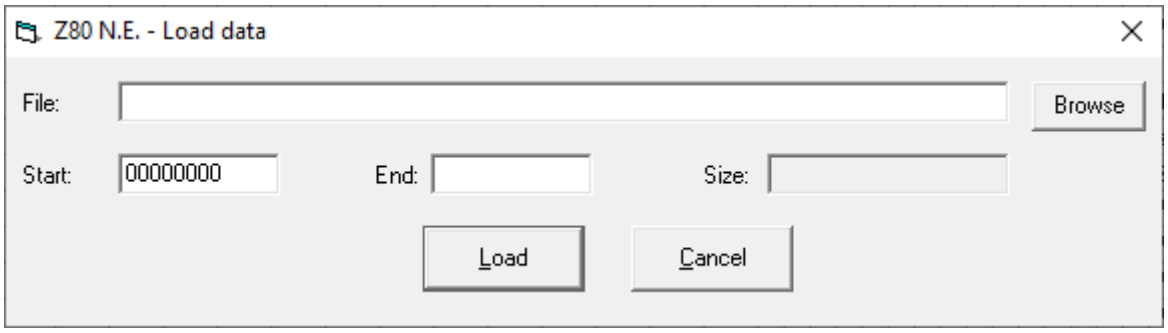
Use this window to edit the entire contents of the currently selected file.

This window lets you edit the contents of an entire file in both hexadecimal and ASCII format. The file contents can be displayed and edited in hexadecimal/ASCII format or in text format.

Here is a brief description of the available options:

- File/Load data menu: select this option to open the Load data (file mode) window.
- File/Save data menu: select this option to open the Save data (file mode) window.
- View/Hex/ASCII menu: select this option to display or edit the file in hexadecimal or ASCII format.
- View/Text menu: select this option to display or edit the file in text format.
- File system type label: indicates the type of file system detected.
- Image file label: indicates the path and filename of the image file or the physical disk.
- File label: shows the name of the currently selected file.
- File size label: shows the size of the currently selected file.
- Cursor label: shows the currently selected address.
- Ok button: click this button to save the changes and close the window.
- Cancel button: click this button to discard the changes and close the window.
- Double-click a hexadecimal or ASCII I/O location: double-click a location to enter edit mode. Press ESC on the keyboard to leave edit mode without saving the value. Leaving edit mode in any other way saves the value.
- Select values: you can select multiple hexadecimal or ASCII values in three ways:
  - Drag the mouse while holding down the left mouse button.
  - Click while holding down the SHIFT key: this selects all values between the current cell and the clicked cell.
  - Click while holding down the CTRL key: this adds or removes the clicked cell from the current selection.
- CTRL+C/CTRL+V: you can copy and paste hexadecimal or ASCII values by pressing CTRL+C or CTRL+V.

Load data (file mode)



This window lets you load data into the currently selected file by reading it from a binary file on the PC.

Here is a brief description of the available options:

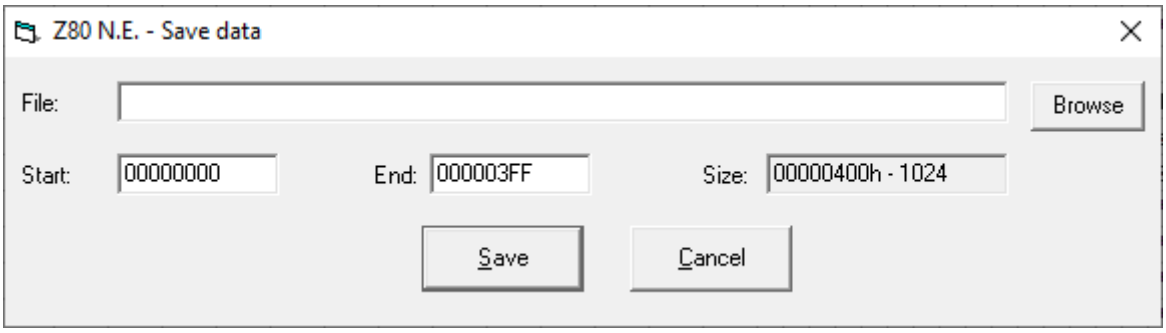
- File field: type the path and filename of the binary file to be loaded in this field.
- Browse button: click this button to select a binary file from your PC.
- Start field: this is the starting address of the destination range for the data to be loaded.
- End field: this is the ending address of the destination range for the data to be loaded. It is automatically filled in when you select a file to load. You can edit it as required.
- Size label: indicates the size of the data range to be loaded. It is displayed in both hexadecimal and decimal format.
- Load button: click this button to load the selected file into the specified range.
- Cancel button: click this button to close the window.

NOTE: the file is loaded only up to the last available address of the destination file, even if its size would extend beyond that address. The destination file is never extended.

NOTE: the file to be loaded is treated as a raw binary file without any headers.

NOTE: any file can be loaded.

Save data (file mode)



This window lets you save data from the currently selected file to a binary file on the PC.

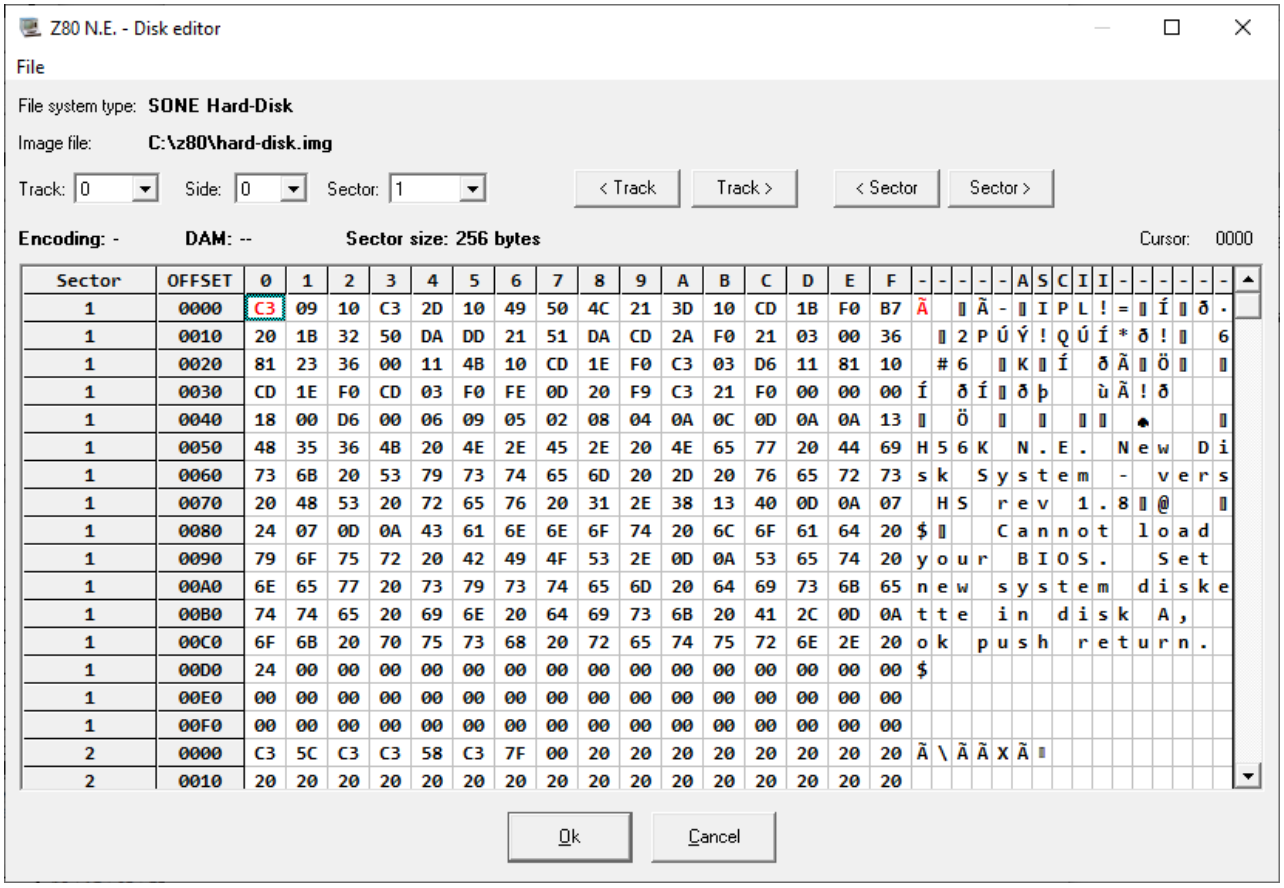
Here is a brief description of the available options:

- File field: type the path and filename of the binary file to be saved in this field.
- Browse button: click this button to select a folder on your PC and specify the name of the binary file to be written.
- Campo Start: this is the starting address of the data range to be saved. When the window is opened, it is automatically set to 0000h. You can edit it as required.
- End address field: this is the ending address of the data range to be saved. When the window is opened, it is automatically set to the maximum file length. You can edit it as required.
- Size label: indicates the size of the data range to be saved. It is displayed in both hexadecimal and decimal format.
- Save button: click this button to save the specified data range to the selected file.
- Cancel button: click this button to close the window.

NOTE: the saved file is a raw binary file without any headers.

Disk editor

Clicking the Disk editor button in the Edit disk or files window displays the “Disk editor” window.



Use this window to edit the entire contents of a disk.

This window lets you edit the contents of an entire disk in both hexadecimal and ASCII format. The displayed data is organized by track, side, and sector.

Here is a brief description of the available options:

- File/Load data menu: select this option to open the Load data (disk mode) window.
- File/Save data menu: select this option to open the Save data (disk mode) window.
- File system type label: indicates the type of file system detected.
- Image file label: indicates the path and filename of the image file or the physical disk.
- Track field: select the track whose data you want to display.

- Side field: select the disk side whose data you want to display.
- Sector field: select the sector whose data you want to display.
- < Track button: click this button to select the previous track.
- Track > button: click this button to select the next track.
- < Sector button: click this button to select the previous sector.
- Sector > button: click this button to select the next sector.
- Encoding label: shows the FM or MFM encoding type.
- DAM label: shows the Data Address Mark used on the selected track.
- Sector size label: shows the size of the selected sector.
- Cursor label: shows the currently selected address.
- Ok button: click this button to save the changes and close the window.
- Cancel button: click this button to discard the changes and close the window.
- Double-click a hexadecimal or ASCII I/O location: double-click a location to enter edit mode. Press ESC on the keyboard to leave edit mode without saving the value. Leaving edit mode in any other way saves the value.
- Select values: you can select multiple hexadecimal or ASCII values in three ways:
  - Drag the mouse while holding down the left mouse button.
  - Click while holding down the SHIFT key: this selects all values between the current cell and the clicked cell.
  - Click while holding down the CTRL key: this adds or removes the clicked cell from the current selection.
- CTRL+C/CTRL+V: you can copy and paste hexadecimal or ASCII values by pressing CTRL+C or CTRL+V.

Load data (disk mode)



This window lets you load data into the disk by reading it from a binary file on the PC.

Here is a brief description of the available options:

- File field: type the path and filename of the binary file to be loaded in this field.
- Browse button: click this button to select a binary file from your PC.
- From/To fields: indicate the destination range on the disk where the data will be loaded. In particular:
  - Track: destination track.
  - Side: destination side of the disk.
  - Sector: destination sector.
  - Offset: offset within the destination sector.
- Load button: click this button to load the selected file into the specified range.
- Cancel button: click this button to close the window.

NOTE: the file is loaded only up to the To address on the destination disk (Track/Side/Sector/Offset), even if its size would extend beyond that address.

NOTE: if the file to be loaded is shorter than the destination range, only the data contained in the file is loaded. The remaining bytes in the destination range are not modified on the destination disk.

NOTE: the file to be loaded is treated as a raw binary file without any headers.

NOTE: any file can be loaded.

Save data (disk mode)



This window lets you save data from the disk by writing it to a binary file on the PC.

Here is a brief description of the available options:

- File field: type the path and filename of the binary file to be saved in this field.
- Browse button: click this button to select a folder on your PC and specify the name of the binary file to be written.
- From/To fields: indicate the range on the source disk whose data will be saved. In particular:
  - Track: source track.
  - Side: source side of the disk.
  - Sector: source sector.
  - Offset: offset within the source sector.
- Save buttone: click this button to save the specified range to the selected file.
- Cancel button: click this button to close the window.

NOTE: the saved file is a raw binary file without any headers.

File references and general architecture

This emulator has been written completely from scratch in Microsoft Visual Basic 6.

Every function has been implemented from the ground up, including CPU emulation, hardware emulation of the various interfaces, and more.

CPU emulation is cycle-exact when running at a fixed selected frequency (provided the host computer is fast enough) and supports the full Z80 instruction set, including undocumented opcodes.

The architecture is multi-process. The CPU core and all hardware emulations run as separate processes (different .exe files) because Visual Basic 6 is not thread-safe and therefore cannot safely use threads. This means that while the emulator is running, you will see different processes and different taskbar icons (one for each emulated hardware window). This is normal and intentional.
The only exceptions to this rule are LX.390 and LX.683, which run in the CPU process.

Inter-process communication is handled through shared memory maps, Windows inter-process events and mutexes. All of this is transparent to the user.

This emulator uses three internal files to work properly:

- Configuration file
- Temporary file
- Temporary trace file

The **configuration file** is a text file created and maintained by the emulator itself. It is located at:

%LOCALAPPDATA%\Z80NE\Z80NE.ini

If you encounter problems with the settings, close the emulator if it is running, then safely delete this file: the emulator will recreate it automatically.

The **temporary file** is used to pass parameters when the emulator starts external processes, such as LX.388 emulation or LX.529 emulation. It is located at:

%TEMP%\Z80NE\Z80NE.tmp

The **temporary trace file** is used to store temporary binary data for the “Trace to file” function. It is automatically deleted when the function ends. It is located at:

%TEMP%\Z80NE\Z80NE_trace_(process_id).tmp

where (process_id) is the Windows process ID of the current emulator instance.

This means that, for the emulator to work properly, the user running it must have write permissions for the directories mentioned above. In most cases this is already the default, so there should be no problems.