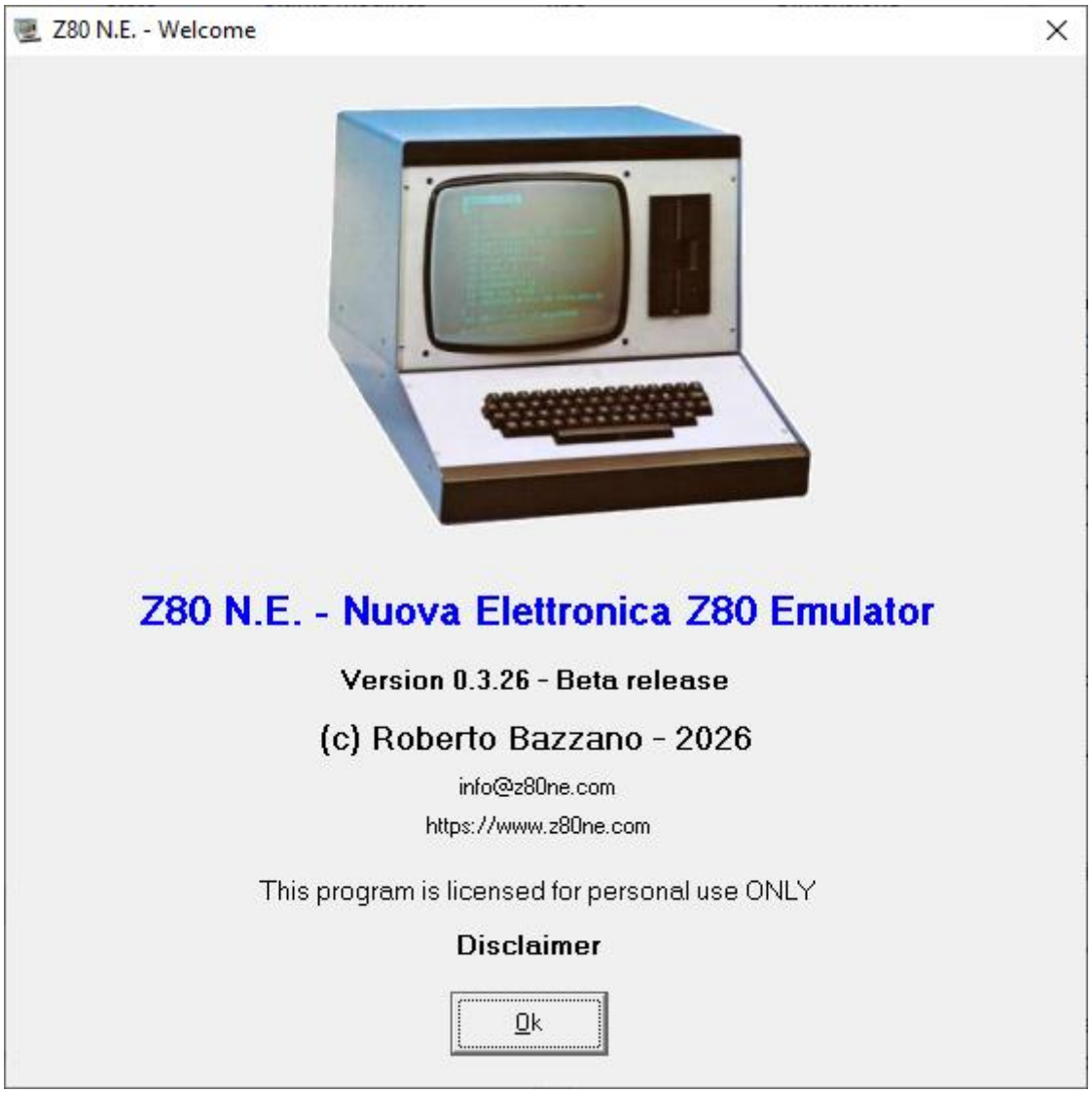
Z80 N.E.

Nuova Elettronica Z80 Emulator



Contents:

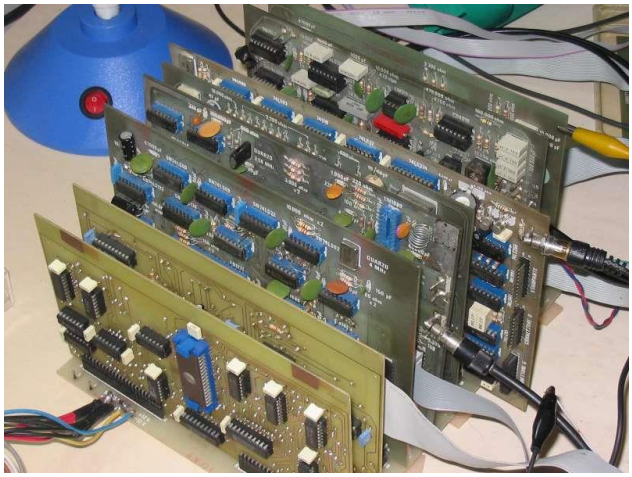
General description

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.

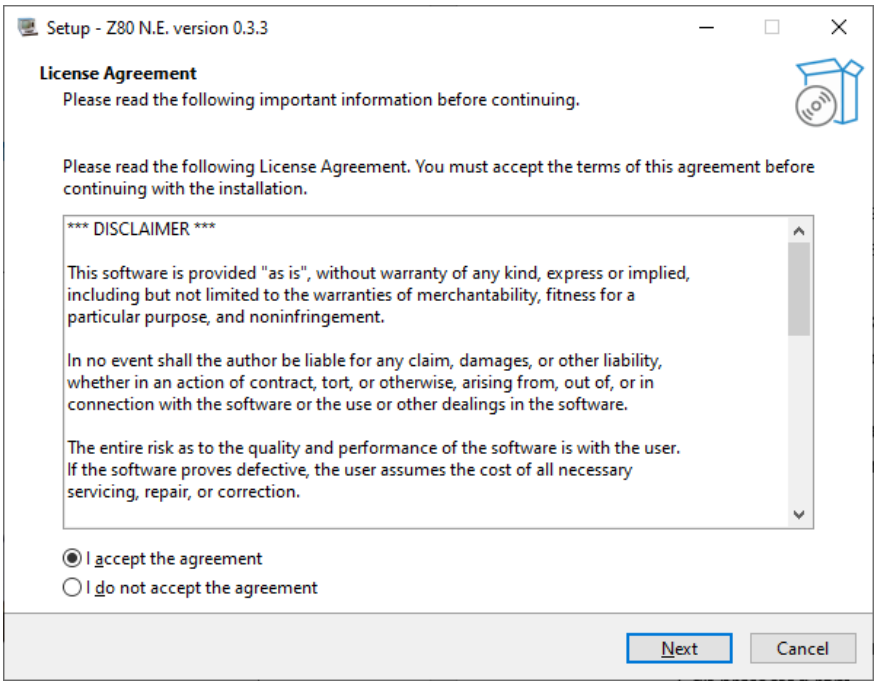
The emulator not only reproduces the CPU itself, but also the main peripherals and expansion boards of the original system, including the hexadecimal keyboard and displays, tape interface, video cards, floppy-disk interface, beeper, interrupt card, hard-disk interface, printer, RAM editor, I/O editor and disassembler. 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 **Nuova Elettronica** beginning in October 1979. 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, image-file-based storage and automatic update support.



Install



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

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

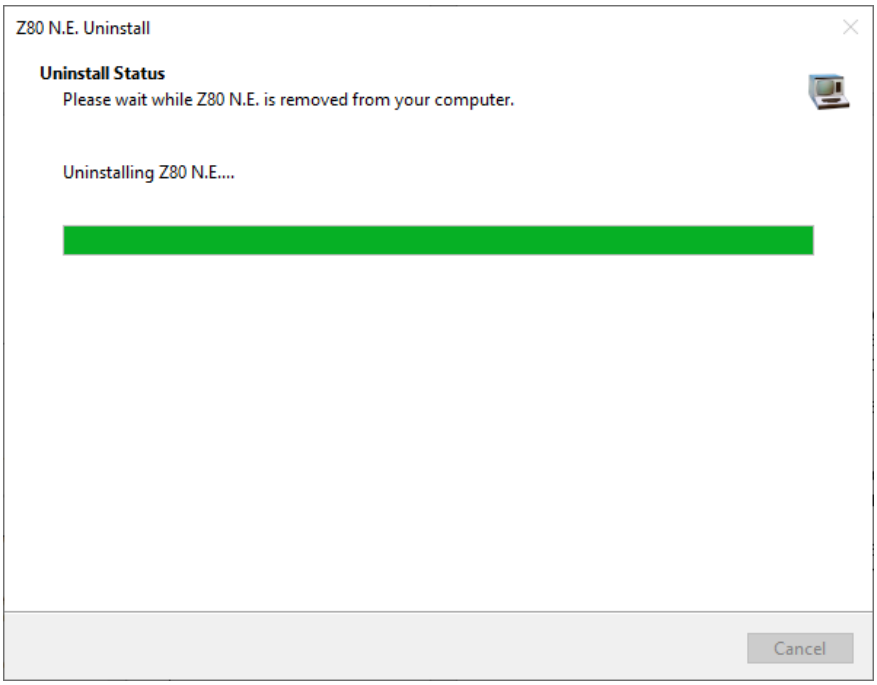
After downloading it, double-click the setup icon.

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.

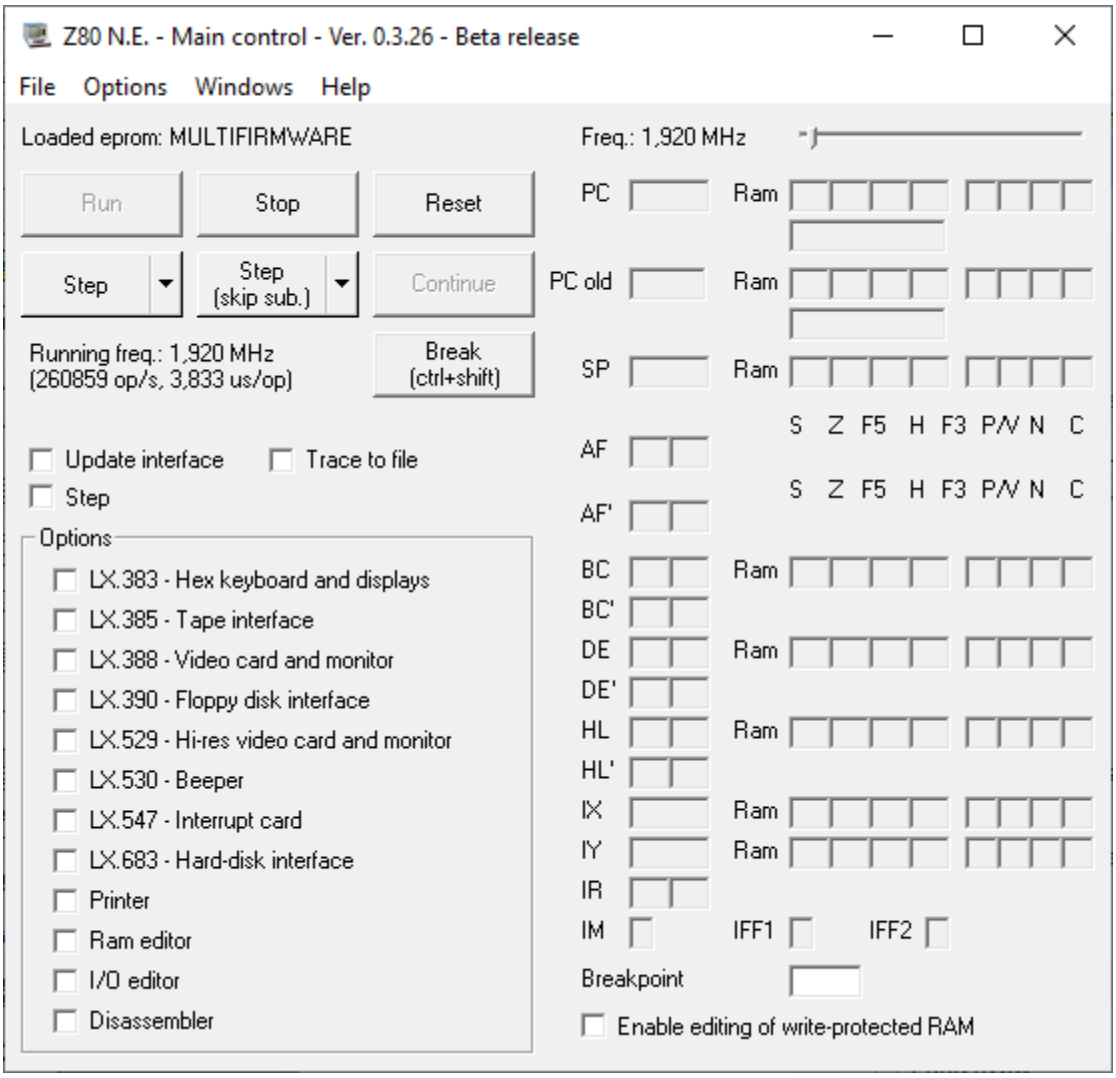
Simply follow the wizard steps to uninstall the application.
The wizard will ask whether you want to remove the configuration file: select Yes if you no longer need it.
The wizard may then ask whether you want to remove some unused .OCX files. It is safe to remove them; however, if you think you may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

Main control

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.

After you close the splash screen, the main control window is displayed.



This is the main control window, which lets you run the CPU and manage the main application options.

Here is a brief description of the available options:

MENU ITEMS:

- File/Load binary file: 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.

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

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

- 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.

- Windows/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.

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

- Windows/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"), the called subroutine is executed completely before step-by-step mode resumes. Step-by-step mode resumes after the CPU executes a "RET" instruction or a similar conditional return such as "RET NZ".

- 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.

- Break button: click this button to simulate pressing the BREAK keys on the emulated keyboards.
  NOTE: you can also press CTRL+SHIFT on your PC keyboard to simulate the BREAK keys.

- 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.
  SP 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 SP depth is the maximum value reached by **SP 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.

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

- Trace to file checkbox: select this checkbox to start tracing your program. When you select it, a file selection window is shown, where you can enter a filename. The trace file can be saved either as plain text or in comma-separated format (.txt or .csv).
  Every instruction executed by the CPU is then written to the selected trace file together with the CPU registers, until you clear this checkbox. A label below the Trace 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: it is very easy for the trace file to 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.

- 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.
  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.

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

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 register: this is the 16-bit PC (Program Counter) CPU register. 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 simply 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 (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.

- AF' registers: these are the alternate 8-bit A and F (Flags) CPU registers.

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

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

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

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

- HL registers: these are the 8-bit H and L 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, which are normally hidden. These labels are shown while an NMI or INT interrupt is being serviced. They are hidden again when a RETN or RETI instruction is executed. In the case of INT, the label is also hidden after the execution of the EI and RET instructions in sequence.

- 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.

- 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: this checkbox opens/closes the disassembler window.

HOW TO OPERATE MAIN CONTROL

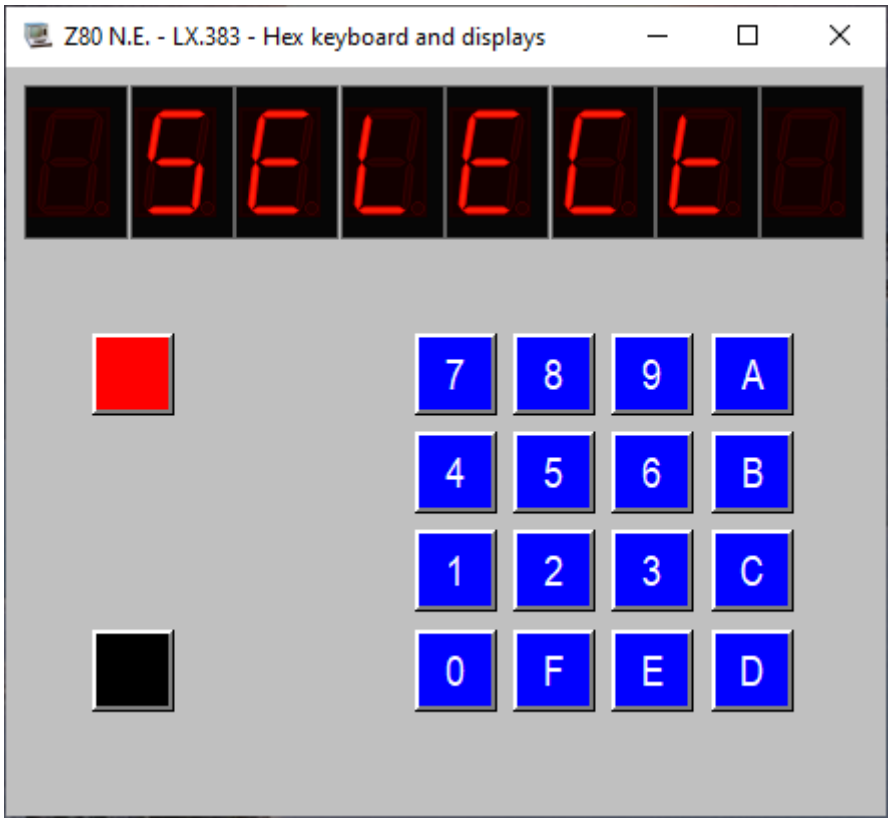
When you start the application, the MULTIFIRMWARE EPROM is loaded by default.

This is a custom-made EPROM that contains all original EPROM firmware images, which you can run by using an on-screen menu, just like on the real machine.

To use the emulator, simply click the “Run” button to power on the CPU, then select the desired option checkboxes. A good starting point is to open LX.388 and LX.529, which are the video interface cards on which the MULTIFIRMWARE main menu is displayed.

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.

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).

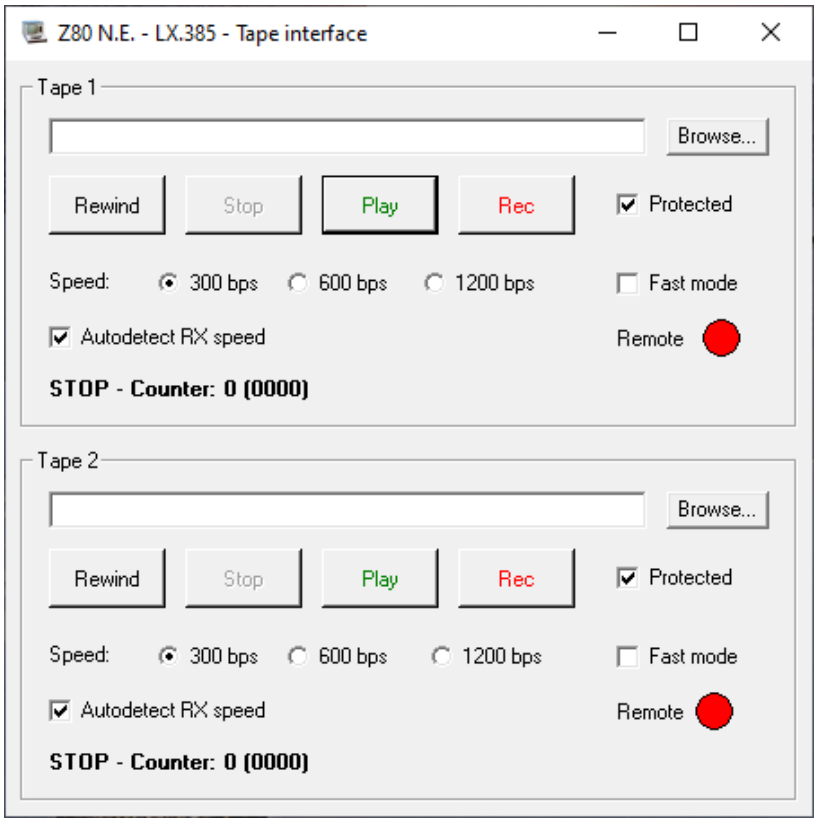
The black button is the CONTROL button. You must press it together with a value button (the blue buttons) to activate the related function, depending on the loaded EPROM. Press and hold the CONTROL button, then press a value button.

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 original machine 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/ita/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.

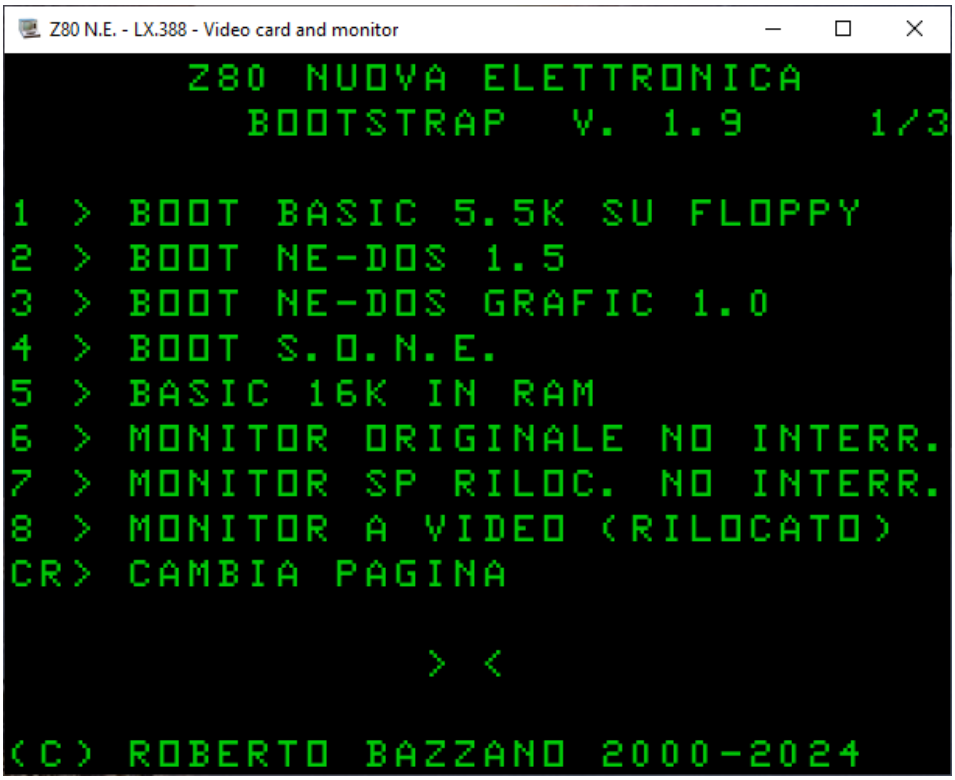
- Speed options: use these options to manually set the tape-data read speed. The available speeds are 300 bps, 600 bps and 1200 bps, as on the real machine.
  To use the manual speed options, “Autodetect RX speed” must be cleared.

- Fast mode: if this option is enabled, reads and writes are performed at the maximum available emulation speed, which is much faster than 1200 bps (depending on the PC CPU).
  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 speed. The emulator automatically sets the speed based on the .WAV data.
  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-phosphor monitor.

When this window has focus, you can type on your PC keyboard and all typed keys are sent to the Z80 CPU.

There are a few key mappings as follows:

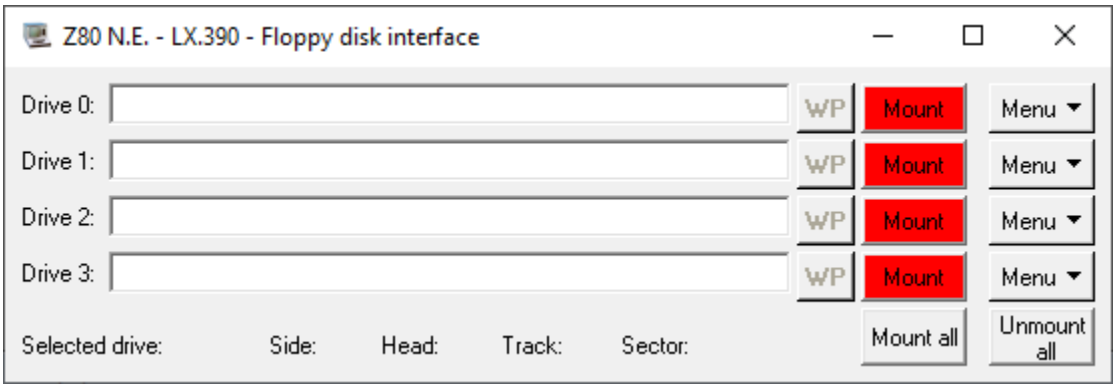
- BREAK key on the real machine: press CTRL+SHIFT. This generates an NMI on the CPU, just like on the real machine.
- LINE FEED: press the “Pg Down” key.

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 NE-DOS 2.1, 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.

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. You can create your own with the emulator, or download them from the Z80 N.E. website at: https://www.z80ne.com/ita/software.asp

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

To use LX.390 emulation, you need to load a disk-image file into one of the virtual drives and then mount it.

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.

- Write protect button: click this button to write-protect an image file. This is equivalent to write-protecting a physical floppy disk. 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. 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.

  - Move button: click this button to move a floppy disk image file from one drive to another.

  - Swap button: click this button to swap the floppy disk image files between two drives.

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

- Unmount all button: click this button to unmount the image files 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 physical emulated drive, while Track is the current track loaded in the emulated FD1771 controller register. In some cases, they could theoretically differ.

LX.529 – Hi-res video card and monitor

This window emulates the LX.529 interface card and a green-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 typed keys are sent to the Z80 CPU.

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.

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

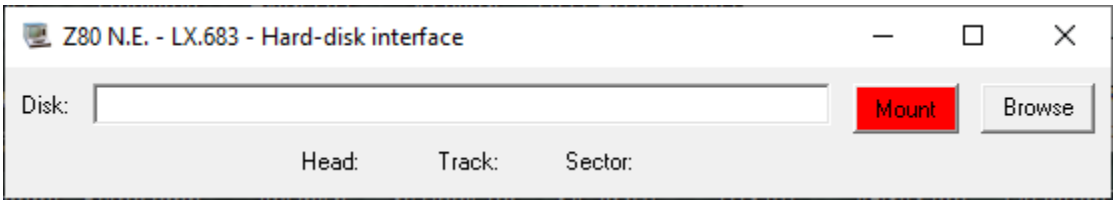
This emulates the LX.547 interrupt card.

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 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.

- Browse button: click this button to locate and load an image file containing hard-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.

- 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.

- 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.

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/Cut paper: select this option to clear the on-screen virtual printer. 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.

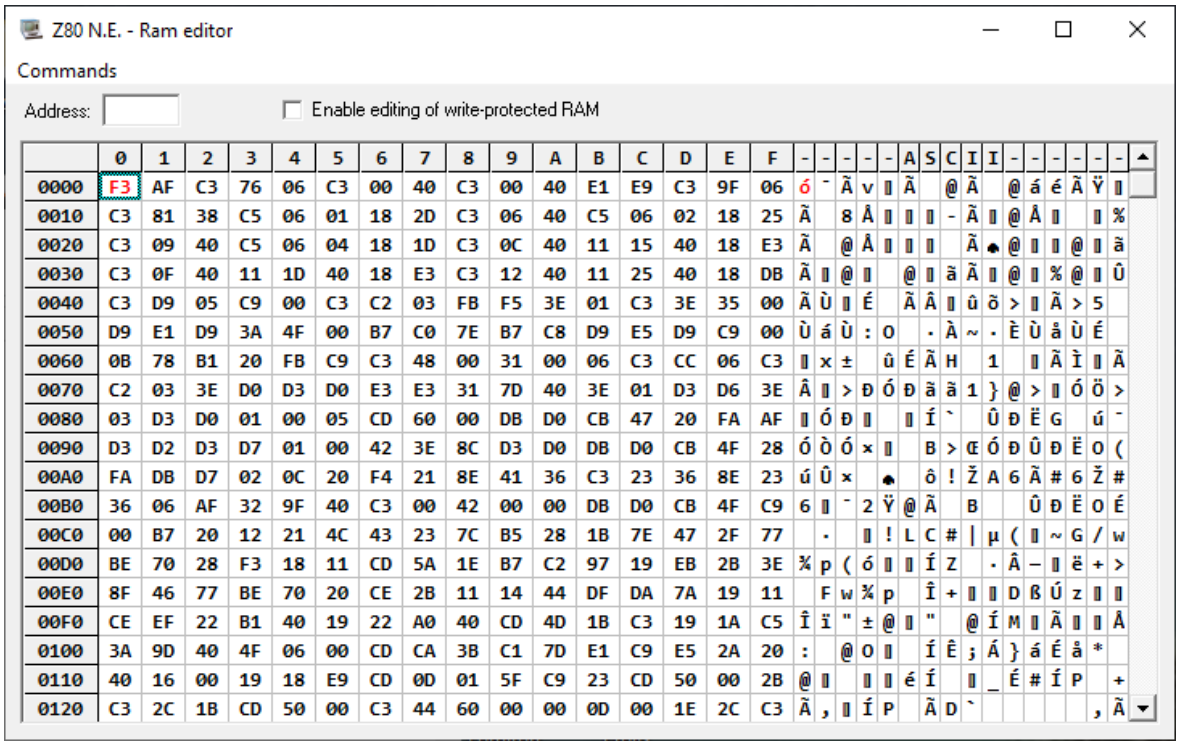
- Layout/A4 paper: use this option to select A4 paper for on-screen display.

- 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 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. This 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:

- Commands/Fill ram menu: select this option to open “Fill ram” window.

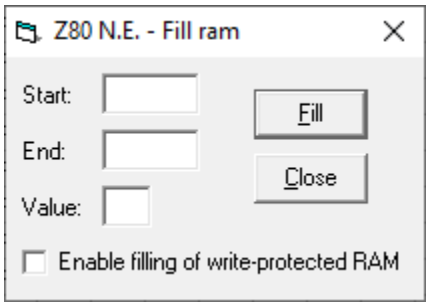
- Address field: type a hexadecimal address to position the view.

- 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.

- 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 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.

Fill ram

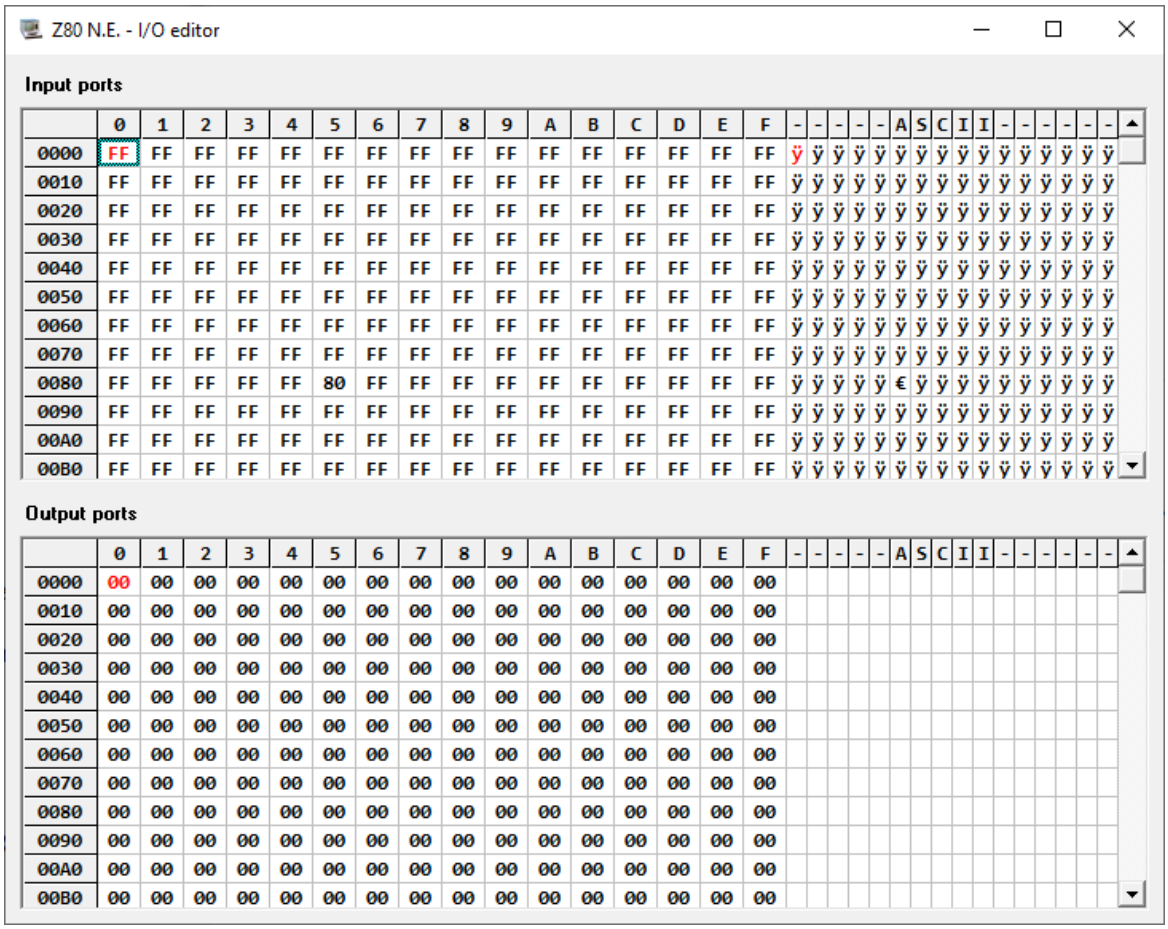


This window lets you fill RAM with value.

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.

I/O editor

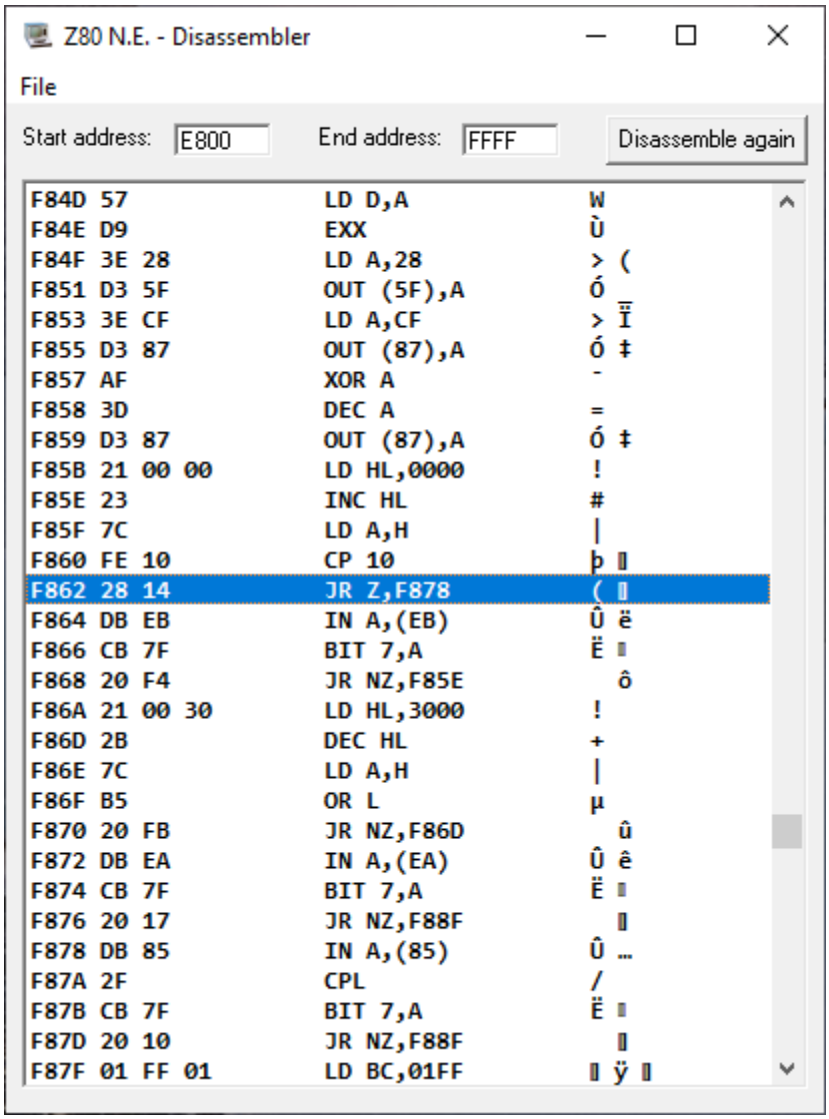


This window is an INPUT/OUTPUT viewer/editor. It shows the contents of the CPU input/output ports, both as hexadecimal values and as ASCII characters.

Here is a brief description of the available options:

- 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.

Disassembler



This window is a disassembler. It shows the CPU instructions currently present in RAM in disassembled form.

The list contains:

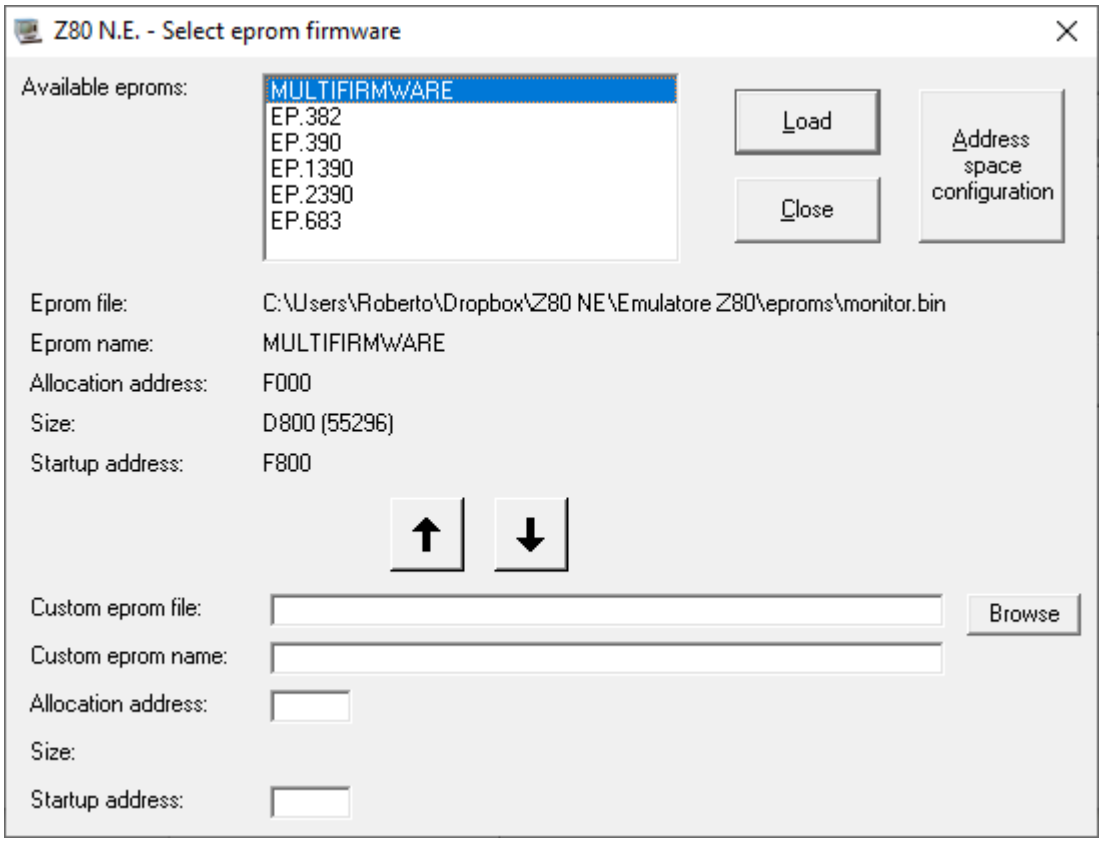
- RAM address
- Instruction opcode
- Disassembled instruction mnemonic
- ASCII conversion of the opcode byte values

Here is a brief description of the available options:

- 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.
- Double-click a line: double-click a line to copy the corresponding address into the “Breakpoint” field of the “Main control” window. This lets you quickly set a breakpoint.

NOTE: when the “Main control” window is in step-by-step mode, the disassembler 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.
- Arrow UP button: click this button to add the custom EPROM data to the list of available EPROMs. This addition is saved immediately.
- Arrow DOWN 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.

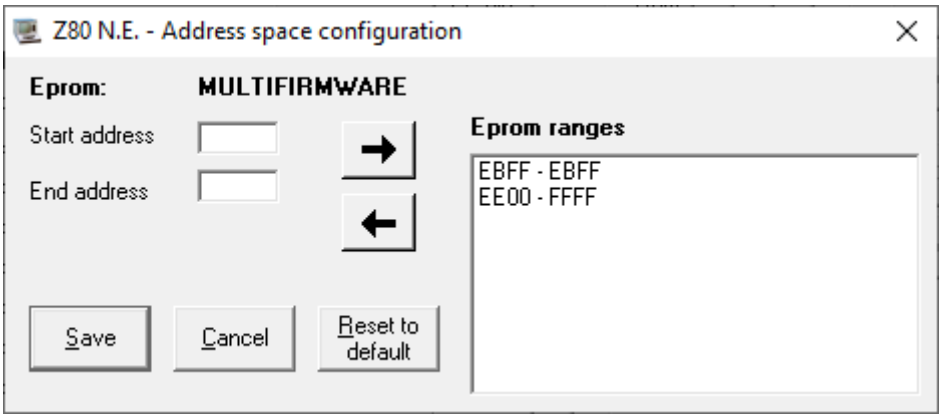
- Close button: click this button to exit without changing the loaded EPROM.

- 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 (factory or custom) is marked as write-protected by default. You can change address space protection by using the “Options/Address space configuration” menu in the “Main control” window.

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.
- Arrow RIGHT button: click this button to add the address range to the list on the right.
- Arrow LEFT 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.

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.