

# fMSX

## version 5.4

MSX Home Computer Emulator  
by [Marat Fayzullin](#)

## Contents

- [New in This Version](#)
  - [Introduction](#)
  - [fMSX Ports](#)
  - [Registered Users](#)
  - [Keyboard Assignments](#)
  - [Command Line Options](#)
  - [Frequently Asked Questions](#)
  - [History](#)
  - [Thanks to...](#)
- 

## New in This Version

- Made replay save states approximately every 170ms.
- Added ability to browse through paused replay.
- During replay, press [UP] to pause or resume.
- When replay paused, press [LEFT] and [RIGHT] to browse.
- Press any other button to continue playing.
- Now cancelling replay if any key or button is pressed.
- Made `WaitJoystick()` exit when window closed.
- 

## Introduction

*fMSX* is a program that emulates MSX, MSX2, and MSX2+ home computers. It runs the majority of MSX software and mimics most popular hardware extensions, such as SCC, OPLL, etc. You can always get the latest fMSX source code, binaries, and support files from

<http://fms.komkon.org/fMSX/>

*fMSX* has a very long history for a piece of software. It has been in continuous development since 1993, when I wrote the first version to run on Unix-based DEC Alpha workstations. Because fMSX is written in C, it is a very portable program. It can run on any sufficiently fast 32bit hardware platform, be it personal computer, PDA, videogame console, cell phone, set-top box, or a DVD player. There are fMSX versions for Unix, Windows, Macintosh, Symbian, MSDOS, Amiga, and many other platforms.

fMSX source code is open for everyone to see but it is not in public domain. You can look and learn from it, but you cannot change it or copy it without giving a credit to the original author and a few other conditions. If you would like to port fMSX to another platform or make changes to the code, please,

contact me by email or some other means.

You cannot use fMSX source code for commercial purposes unless you contact me to arrange the conditions of such usage. If your company intends to use MSX software in its products and you are considering using fMSX source code, please, email me about licensing.

**MSX** is an old 8bit family of home computers created in 1982 as an attempt to establish a single standard in home computing similar to VHS in video. MSX computers have been popular in Asia (Korea, Japan) and South America (Brazil, Chile) as well as in Europe (Netherlands, France, Spain) and former Soviet Union, although they are virtually unknown in the USA. Although the MSX platform quietly died around 1988, the world got to see MSX2, MSX2+, and TurboR extensions of the MSX platform.

The MSX has been mainly designed by a Japanese company called [ASCII](#) in cooperation with [Microsoft](#), who provided the firmware BASIC for the machine. There is a widespread rumor that "MSX" stands for "MicroSoft eXtended". MSX machines were produced by such giants as Sony, Yamaha, Panasonic, Toshiba, Daewoo, and Philips. The only MSX model ever sold in the US appears to be an early SpectraVideo machine though.

In spite of its short history, MSX is a very nice computer, especially good for education, as seen in an example of the Soviet Union. Soviet Ministry of Education bought hundreds of MSXes (and later MSX2s) grouped into "computerized classroom systems" of 10-16 machines connected with a simple network. A whole generation of programmers has grown up using these computers.

Hardware-wise, MSX represents a hybride of a videogame console and a generic CP/M-80 machine. Its heart is a Z80 CPU running at 3.58MHz in the base model. The clock frequency has been doubled in the TurboR. The video subsystem is built around a TI9918 or TI9928 VDP chip also used in Texas Instruments' TI-99/4 computers, ColecoVision, and Coleco Adam. In the late MSX models, this chip has been upgraded to V9938 (MSX2) and then to V9958 (MSX2+ and TurboR). The latest version of this chip is known as V9990. The audio system is handled by an AY-3-8910 chip from General Instruments, same as the one used in Sinclair ZX Spectrum 128. AY-3-8910 provides three channels of melodic sound, a noise channel, volume envelopes, and two general purpose parallel IO ports, which MSX uses for joysticks and a few other things. Due to their hardware architecture, MSX machines were perfectly suitable for games and there was a lot of good games either written for or ported to them.

## fMSX Ports

Because fMSX is a very portable program, it can run on many different platforms: Unix, Macintosh, MSDOS, Windows, Symbian, PocketPC, Amiga, etc. The complete up-to-date list of fMSX ports is available at the fMSX [distribution site](#). Following are the major ports:

### fMSX-Windows

Starting December 2013, *fMSX-Windows* is free for everyone to use, in binary form. You can download it from the fMSX [distribution site](#). Also see [my other emulators for Windows](#).

### fMSX-Android

*fMSX-Android* is available from the [Google Play Store](#). You can download the [full version](#) or the feature-limited [free demo](#). Also see [my other emulators for Android](#).

### fMSX-Unix

*fMSX-Unix* is available freely in the source code form from the fMSX [distribution site](http://fms.komkon.org/fMSX/fMSX.html).

## fMSX-Linux

*fMSX-Linux* (ELF, GLIBC, X11) is compiled from the same source code as all other Unix versions. You can get it for free from the fMSX [distribution site](http://fms.komkon.org/fMSX/fMSX.html).

## Registered Users

If you've registered fMSX-Windows, please do not give your copy to anybody. And I do mean anybody. There were cases when registered users gave fMSX away to their friends, relatives, or just some shady characters on the Net, and then I found it pirated, put onto the Web, and even posted to USENET newsgroups. If I find your personalized copy of fMSX being spread around, your registration gets automatically cancelled which means no support and no more updates.

I understand that the previous paragraph may sound threatening to some people, but this kind of piracy really hurts my profits and feelings. I've put a lot of effort into fMSX, and can only continue working on it if people do not try to cheat on me.

## Keyboard Assignments

```
[CONTROL]      - CONTROL (also: joystick FIRE-A button)
[SHIFT]        - SHIFT (also: joystick FIRE-B button)
[ALT]          - GRAPH (also: swap joysticks)
[INSERT]       - INSERT
[DELETE]       - DELETE
[HOME]         - HOME/CLS
[END]          - SELECT
[PGUP]         - STOP/BREAK
[PGDOWN]       - COUNTRY
[F6]           - Load emulation state from .STA file
[F7]           - Save emulation state to .STA file
[F8]           - Rewind emulation back in time
[F9]           - Fast-forward emulation
[F10]          - Invoke built-in configuration menu
[F11]          - Reset hardware
[F12]          - Quit emulation
[CONTROL]+[F8] - Toggle scanlines on/off
[ALT]+[F8]     - Toggle screen softening on/off
[CONTROL]+[F10] - Go to the built-in debugger
```

## Command Line Options

Usage: `fmsx [-option1 [-option2...]] [filename1] [filename2]`

```
[filename1] = name of file to load as cartridge A
[filename2] = name of file to load as cartridge B
```

When compiled with `#define ZLIB`, fMSX will transparently uncompress singular GZIPped and PKZIPped files.

```
[-option] =
-verbose <level> - Select debugging messages [1]
                  0 - Silent           1 - Startup messages
                  2 - V9938 ops       4 - Disk/Tape
                  8 - Memory          16 - Illegal Z80 ops
-skip <percent>  - Percentage of frames to skip [25]
-pal/-ntsc       - Set PAL/NTSC HBlank/VBlank periods [NTSC]
```

```

-help                - Print this help page
-home <dirname>     - Set directory with system ROM files [off]
-printer <filename> - Redirect printer output to file [stdout]
-serial <filename> - Redirect serial I/O to a file [stdin/stdout]
-diska <filename>  - Set disk image used for drive A: [DRIVEA.DSK]
                   (multiple -diska options accepted)
-diskb <filename>  - Set disk image used for drive B: [DRIVEB.DSK]
                   (multiple -diskb options accepted)
-tape <filename>   - Set tape image file [off]
-font <filename>   - Set fixed font for text modes [DEFAULT.FNT]
-logsnd <filename> - Set soundtrack log file [LOG.MID]
-state <filename>  - Set emulation state save file [automatic]
-auto/-noauto      - Use autofire on SPACE [off]
-ram <pages>       - Number of 16kB RAM pages [4/8/8]
-vram <pages>      - Number of 16kB VRAM pages [2/8/8]
-rom <type>        - Select MegaROM mapper types [8,8]
                   (two -rom options accepted)
                   0 - Generic 8kB      1 - Generic 16kB (MSXDOS2)
                   2 - Konami5 8kB     3 - Konami4 8kB
                   4 - ASCII 8kB      5 - ASCII 16kB
                   6 - GameMaster2   7 - FMPAC
                   >7 - try guessing mapper type
-msx1/-msx2/-msx2+ - Select MSX model [-msx2]
-joy <type>        - Select joystick types [0,0]
                   (two -joy options accepted)
                   0 - No joystick
                   1 - Normal joystick
                   2 - Mouse in joystick mode
                   3 - Mouse in real mode
-simbdos/-wd1793   - Simulate DiskROM disk access calls [-wd1793]
-sound [<quality>] - Sound emulation quality (Hz) [44100]
-nosound           - Same as '-sound 0'
-sync <frequency> - Sync screen updates to <frequency> [60]
-nosync           - Do not sync screen updates [-nosync]
-static/-nostatic  - Use static color palette [-nostatic]
-tv/-lcd/-raster   - Simulate TV scanlines or LCD raster [off]
-soft/-eagle       - Scale display with 2xSaI or EAGLE [off]
-epx/-scale2x      - Scale display with EPX or Scale2X [off]
-cmy/-rgb          - Simulate CMY/RGB pixel raster [off]
-mono/-sepia       - Simulate monochrome or sepia CRT [off]
-green/-amber      - Simulate green or amber CRT [off]
-4x3               - Force 4:3 television screen ratio [off]

With #define DEBUG:
-trap <address>    - Trap execution when PC reaches address [FFFFh]
                   (when keyword 'now' is used in place of the
                   <address>, execution will trap immediately)

With #define MITSHM:
-shm/-noshm        - Use MIT SHM extensions for X [-shm]

With #define UNIX:
-saver/-nosaver    - Save/don't save CPU when inactive [-saver]
-scale <factor>    - Scale window by <factor> [2]

With #define MSDOS:
-vsync             - Sync screen updates to VBlank [-vsync]
-480/-200          - Use 640x480 or 320x200 VGA mode [-200]

```

## Frequently Asked Questions

### 1. Where do I get MSX software?

Go to <http://fms.komkon.org/MSX/> and follow links from there.

### 2. What do I do with .BAS,.GMB,.CRC,.LDR files?

These are BASIC programs. You run them from MSX BASIC with

```
RUN "filename"
```

### 3. What do I do with .BIN,.OBJ,.GM files?

These are binary files with programs, also known as BLOADable files. You can run them from MSX BASIC with

```
BLOAD "filename",R
```

### 4. What do I do with .COM files?

These are MSXDOS command files. You can run them from MSXDOS by typing their names sans the .COM extension.

### 5. What are the .ROM files?

These are binary images of cartridge ROMs that you can load into fMSX. There are "small" cartridge ROMs of 8kB, 16kB, or 32kB, and the MegaROMs, which can be 128kB, 256kB and even 512kB.

### 6. What are the .ROM files included with fMSX?

Following .ROM files may not be normal cartridges, but they are used by fMSX:

```
MSX.ROM      - Standard MSX BIOS and BASIC code
MSX2.ROM     - MSX2 BIOS and BASIC code
MSX2EXT.ROM  - MSX2 ExtROM containing system extensions
MSX2P.ROM    - MSX2+ BIOS and BASIC code
MSX2PEXT.ROM - MSX2+ ExtROM containing system extensions
DISK.ROM     - MSX DiskROM containing BDOS and Disk BASIC (optional)
RS232.ROM    - RS232 BIOS and BASIC extensions (optional)
FMPAC.ROM    - FM-PAC BIOS and BASIC extensions (optional)
MSXDOS2.ROM  - MSXDOS2 system core (optional)
PAINTER.ROM  - Yamaha Painter, graphical editor found in Russian MSX
              machines from Yamaha (optional)
GMASTER.ROM - Konami GameMaster, a game cheating tool (optional).
GMASTER2.ROM - Konami GameMaster2, a game cheating tool (optional).
KANJI.ROM    - ROM with Kanji character images (optional)
CMOS.ROM     - Non-volatile memory used in MSX2 and MSX2+. This file gets
              overwritten on exit if non-volatile memory has been changed.
```

Please note that not all of these may be included with your fMSX copy.

### 7. How do I use disks with fMSX?

fMSX includes disk support starting with version 0.9. If you have an earlier version, you cannot use disks. The following instructions assume that you have a version supporting disks.

First, make sure that the DISK.ROM file containing MSX DiskROM is in the current directory. Then, use an MSDOS program called DCOPY.EXE to create disk images of your MSX disks:

```
DCOPY <drive>: <filename>.DSK
```

These images are just raw files with all disk blocks written in a sequence. They can also be created on a Unix machine with

```
cp /dev/rfd0 <filename>.DSK
```

or a similar command. If you have a 1.44MB HD floppy formatted on MSX for 720kB, don't forget to stick a piece of tape on the HD/DD indicator hole.

After you have created disk image files, run fMSX in the following way:

```
fmsx -diska <filename1>.DSK -diskb <filename2>.DSK
```

where two image filenames will become your drives A: and B:. You can also have default disk images called DRIVEA.DSK and DRIVEB.DSK and located in the current directory.

## 8. Is there an easier way to work with disk images?

In fact, there is. Look at two programs that come with fMSX, called `wrdsk` and `rddsk`. The `wrdsk` program allows you to create a disk image and add files to it:

```
wrdsk <filename>.DSK <file> <file> ...
```

The `rddsk` program will read files from a given disk image:

```
rddsk <filename>.DSK [-d <dir>] [<file> <file> ...]
```

## 9. I can't get some BASIC programs to work with the emulator.

Many MSX programs have BASIC loaders written for machines not quite compatible with each other. Other loaders expect the machine to have only one floppy disk drive and use the memory dedicated to the second drive. To make such programs work, perform two "magic passes" on fMSX before running a loader:

- When booting MSX, press [CTRL]+[DEL] keys to switch off the second disk drive.
- Before running a program, type `POKE &hFFFF, &hAA` to set MSX memory manager into the mode expected by most loaders.

## 10. I can't get some ROM images to work with the emulator.

If the ROM image is bigger than 32kB, try using `-rom <N>` parameter with different `<N>`s (see `fmsx -help` for a complete list). If you still can't get ROM image to work, send it to me for analysis.

## 11. Is it legal to spread cartridge ROMs?

NO. Nobody seems to care though, mainly because there is no profit to be made from MSX software any longer. Nevertheless, be aware of the fact that by using commercial software you haven't bought you are commencing an act of piracy.

## 12. When compiling fMSX-Unix, I get "undefined name" errors.

This means that your linker cannot find the libraries necessary for the emulator (namely, `libX11.a` and `libXext.a`) or some additional libraries (like `libsocket.a` and `libnsl.a`) are required. Find these libraries in your system and modify the Makefile so that the final invocation of the C compiler has `-L<path_to_libs>` options. If you have no `libXext.a` library, try #undefining `MITSHM` option.

## 13. When starting fMSX-Unix, I get X\_ShmAttach error.

You are probably trying to run the emulator on a remote Xterminal while it attempts to use shared memory for interfacing with X. Use the `-noshm` option to disable the shared memory usage.

## 14. The emulation starts under Unix, but then I get X\_PutImage error.

Unix/X version of fMSX can currently be compiled for 8bit, 16bit, or 32bit Xterminals. Neither 1bit nor 4bit Xterminals will work with the drivers included into official fMSX distribution. Arnold Metselaar developed the drivers supporting any Xterminal. These drivers are available from the [fMSX distribution site](#).

# History

## New in fMSX 5.3

- Added remaining time display during replay.
- Stopping replay when emulation reset or state loaded.
- Fixed replay rollover mechanism.
- Added shadow underneath the time display.
- Compiled fMSX-Unix with `-Wall` and eliminated warnings.
- Replaced `-DNO_WAVE_INTERPOLATION` with `-DWAVE_INTERPOLATION`, off by default.
- Removed old `LoadSTA()` and `SaveSTA()` code.
- Finally deprecated `-DNEW_STATES`.

## New in fMSX 5.2

- Fixed DirectInput joysticks support in Speccy-Windows.
- Made directional pads work on XBox gamepads.

## New in fMSX 5.1

- Added multiple monitor support to fMSX-Windows.
- Fixed disappearing window in fMSX-Windows.
- Fixed best full-screen mode detection in fMSX-Windows.
- Fixed window flicker that appeared after recent Win10 updates.
- Moved disk and cartridge options to "Hardware" menu in fMSX-Windows.
- Switched fMSX-Linux to using PulseAudio sound.
- Updated old PulseAudio driver for 64bit Linux.

## New in fMSX 5.0

- Switched to microseconds in the AY8910 PSG emulation.
- Now updating, rendering, and playing sound every 8 scanlines.
- The `PLAY "S8M9006G"` MML test works now.
- Added support for DirectInput joysticks to fMSX-Windows.
- Fixed crash when changing audio sampling rate in fMSX-Windows.

## New in fMSX 4.9

- Moved input settings to the "Input" menu in fMSX-Windows.
- Added *"Draw 65% Frames"* option to fMSX-Windows.
- Fixed screen updates after changing scaling algorithm in fMSX-Windows.
- Fixed *"Force 4:3 Screen"* option in TEXT80 screen mode.
- Fixed *"Load MSXDOS2 ROM"* option in fMSX-Windows.
- Fixed *"Hit MIDI Drums"* option in fMSX-Windows.
- Now computing MIDI volume by waveform analysis.
- Corrected MIDI drums volume in AY8910 emulation.
- Fixed MIDI keyboard click in fMSX-Windows.

## New in fMSX 4.8

- Fixed loading of BASIC-only cartridges (*Danger X4*, *Crazy Bullet*, etc).
- Added a hack to ignore bad writes to ASCII16 MegaROM mapper (*Vaxol*).
- Added "*Interpolate Video | Nearest Neighbor*" option to fMSX-Windows.
- fMSX-Windows will not overwrite last loaded file name with other names.
- Added "*Force 4:3 Screen*" option to fMSX-Windows.
- Added `-4x3` command line option to fMSX-Unix.
- Fixed file associations on Windows 8+.
- Slightly rearranged Windows menus.

## New in fMSX 4.7

- Added Scale2X scaling algorithm.
- Added Monochrome, Green, Amber, and Sepia CRT emulation.
- Added `-scale2x` and `-raster` command line options.
- Added `-mono`, `-sepia`, `-green`, and `-amber` command line options.
- EMULib-specific command line options now processed inside EMULib.
- Added "*Scale2x Algorithm*" and "*LCD Raster*" options on Windows.
- Renamed "*Color Raster*" menu to "*Color Filter*" on Windows.
- Deprecated `-notv` and `-nolcd` command line options.

## New in fMSX 4.6

- Enabled magnified sprites (*Jawbreaker*, *Stray Cat*, *F-Zeru*, *Triumph*).
- Fixed sprite collision detection, especially in BASIC games.
- Added collision detection even on transparent sprites.
- Now stopping collision detection when Y=216 in SCREEN 4-8.
- Now setting lower VDPStatus bits to the last tested sprite.

## New in fMSX 4.5

- Changed VDP VRAM access logic to comply with TMS9918 datasheet.

## New in fMSX 4.4

- Added special effects simulating individual pixel components.
- Added EPX and EAGLE image scaling algorithms.
- Merged scanline options into "*Simulate Scanlines*" in fMSX-Windows.
- Merged scaling options into "*Interpolate Video*" in fMSX-Windows.
- Added "*Clear File Associations*" option to fMSX-Windows.
- Added "*Play Sound When Inactive*" option to fMSX-Windows.
- Added "*Apply Color Raster*" submenu to fMSX-Windows.
- Added `-rgb` and `-cmy` command line options.
- Added `-epx` and `-eagle` command line options.

## New in fMSX 4.3

- Added instant replay function (press [F8]) that "rewinds" gameplay.
- Switched to the new state saving code.
- Built-in menu now available when pressing [F10].
- Built-in debugger now available when pressing [CTRL]+[F10].
- Fixed a memory corruption bug in the instant replay recorder.
- Fixed instant replay not recording the first state.



## New in fMSX 4.2

- Added Cheat Hunter tool, press F8 and select "Search Cheats".
- Removed waveform interpolation for more realistic sound.
- Changed PSG noise to be more realistic.
- Switched fMSX-Windows to Direct3D textures-based rendering.
- Added `-lcd` and `-nolcd` command line options.
- Added "Video | Stretch Full Screen" option to fMSX-Windows.
- Added "Hardware | Debugger" option to fMSX-Windows.
- Added "Help | MSX.ORG Site" option to fMSX-Windows.
- Added "Help | MSX.ORG Forums" option to fMSX-Windows.

## New in fMSX 4.0

- Added custom palettes. To make a custom palette for `Game.rom`, create `Game.pal` containing 16 `#RRGGBB` values, one per line. This palette file will be loaded automatically.
- Added cheat codes. To make cheat codes for `Game.rom`, create `Game.cht` containing codes in `00AAAAAA-DD` and `00AAAAAA-DDDD` formats, one per line. The cheat file will be loaded automatically.
- `AAAAAA` is the ROM address and `DD` is the value to write there. For 16bit values, use `DDDD`.
- To enable cheat codes, select "*Hardware | Cheats*" in fMSX-Windows, or press F8 and select "*Cheats | Enable*".
- To edit cheat codes, press F8 and select "*Cheats*".

## New in fMSX 3.9

- Majorly improved AY8910 PSG emulation.
- Added support for `.MX1/.MX2` ROM files.
- Set precise CPU, VDP, and PSG frequencies.
- Switched to table-based PSG envelopes.
- Switched to logarithmic PSG volumes.
- Fixed white noise frequency computation.
- Adjusted noise to be 50% of the melodic volume.
- Adjusted PSG envelopes length.
- Fixed recognition of multiple file extensions.

## New in fMSX 3.8

- Added support for tape images, supplied in `*.CAS` files.
- Implemented 5th/9th sprite fields in the VDP status register.
- No longer clearing VRAM address latch when reading VDP status.
- Added tape operations to the built-in menu.
- Added "*Simulate LCD Scanlines*" menu option to fMSX-Windows.
- Added "*Load Tape Image*" menu option to fMSX-Windows.
- Added "*Rewind Tape*" menu option to fMSX-Windows.
- Fixed "*Documentation*" menu option to fMSX-Windows.

## New in fMSX 3.7

- **fMSX-Windows is now free!**
- Added accurate R register emulation, many protected boot loaders run now.
- Added accurate LDIR/LDDR emulation, more protected boot loaders run now.

- Added undocumented opcode 71h to the Z80 emulation.
- Added more realistic "LD R,A" opcode to the Z80 emulation.
- Added cartridge type lookup by its SHA1 checksum, as listed in the `CARTS.SHA` file.
- Updated EMULib to a newer, more stable version.
- Switched fMSX-Windows version to use Direct3D in full-screen mode.
- Added proper Windows joystick support.
- Now properly restoring full-screen mode in fMSX-Windows.
- Fixed a bug in fMSX-Unix that corrupted memory on most platforms but, for some reason, not on Ubuntu Linux.
- Added a separate `Rules.Solaris` make rules file. Include this file into your `Makefile` instead of `Rules.Unix` to compile for Solaris and don't forget about removing `-DLSEB_FIRST` on SPARC hardware.
- Fixed compilation warnings when compiling fMSX with CLANG.

## New in fMSX 3.5

- After a long pause, finally updated *fMSX-Unix* port. fMSX-Unix source code is once again available as part of the core source code distribution. It has been debugged on the Ubuntu Linux. Let me know of any incompatibilities with other Unix flavors.
- Extended *fMSX-Symbian* port to support *UIQ3* phones from Sony Ericsson and Motorola. As I only have key-operated Motorola Z8, testers with pen-based SE phones are badly needed.
- Added run-time compatibility with 8bpp, 16bpp, 24bpp (32bit pixels) and 32bpp screen depths to fMSX-Unix. IMPORTANT: At this moment, built-in config menu, debugger, and NetPlay dialog will only work at X11 screen depth that matches your compile-time setting (i.e. `-DBPPxx`) in Makefile. Make sure you always compile fMSX-Unix for your default screen depth.
- Disabled secondary slots for primary slot #0 in MSX1 mode. DiskROM boots in MSX1 mode again.
- Now syncing all sound channels after loading state, sound does not get "stuck" any more.
- Renamed command line option `-zoom/-nozoom` into `-soft/-nosoft`.
- Moved a few key definitions, so that `"!@#$$%^..."` characters can now be used as indices into `Keys[]`. This is done for compatibility with GDK key events.
- Renamed `KeyMap[]` to `KeyState[]` to avoid name clash with MacOSX system API.
- ALT swaps joysticks 1/2 on all platforms where applicable.
- CONTROL/SHIFT keys now act as FIRE-A/FIRE-B joystick buttons on all platforms.
- Improved fMSX-Symbian sound code, although sound problems still persist on some UIQ3 phones.
- Split fMSX-Symbian/S60 configuration menu into four tabs, making "Video" and "Audio" settings separate.
- Added "Audio Latency", "Skip Frames", and "Sync Updates" settings to fMSX-Symbian.
- Added to fMSX-Symbian ability to use Nokia E61/E62 [FUNC] key in the built-in menu.
- Fixed an fMSX-Symbian bug where pieces of virtual keyboard remained on screen after closing it.
- Pressing [8] key on fMSX-Symbian now invokes virtual keyboard, if enabled, for the phones which do not have the [EDIT] key.

## New in fMSX 3.4

- Added **network play** option to fMSX-Windows and fMSX-Symbian!
- Now not opening printer file until the first character is printed.
- Fixed OUTD/OUTDR opcodes to modify B register before the OUT operation.
- No longer trying to load state files from LoadFile().
- Defined `"INLINE"` properly, depending on the C standard supported by the compiler.
- Finally retired `#ifdef SOUND` as sound is always compiled in anyway.
- Added new, much better, scanline simulation and video softening effects to fMSX-Windows.
- fMSX-Windows full-screen mode now runs in 640x480 resolution.

- Fixed problem with full-screen SCREEN 6/7 in fMSX-Windows.
- fMSX-MSDOS has got complete (non-truncated) emulation for SCREEN 6/7 and TEXT80 screen modes.
- fMSX-MSDOS now always runs in 640x480x15bpp VESA screen mode.
- Added options to simulate TV scanlines (`-tv/-notv`) and zoom display (`-zoom/-nozoom`) to fMSX-MSDOS.
- Options `-sync/-nosync/-vsync` work properly again in fMSX-MSDOS.
- Added new "Fill & Soften" zoom mode to fMSX-Symbian. It is rather slow though.
- Added ability to use virtual keyboard in fMSX-Symbian to enter filename when saving files from the built-in menu.
- Fixed problem with gameplay being too fast in fMSX-Symbian.
- Fixed possible source of instabilities in the S60 open file dialog in fMSX-Symbian.
- Fixed exit via Symbian-specific menu.
- Fixed premature termination of the emulation thread when exiting fMSX-Symbian, soundtrack recording and config saving work again.
- The [C] ("Clear") key is no longer used to toggle sound in fMSX-Symbian. It acts as [BACKSPACE] instead.

## New in fMSX 3.3

- Save state format has changed due to serious changes to the emulator architecture!
- Added `OpZ80()` function and the `FAST_RDOP` macro to the Z80 emulation to optimize code access in architectures with memory-mapped I/O.
- Added subslot support to all slots.
- Moved `PAINTER.ROM` to slot 0:1, `FMPAC` to slot 0:2, `MSXDOS2.ROM` to slot 3:0.
- Reduced the number of user cartridges back to two (A/B). All the other slots are taken by system cartridges (`MSXDOS2`, `FMPAC`, etc.).
- Locked user cartridges in slots 1/2 to subslot 0, as external cartridges often do not use subslots.
- If Konami's `GMASTER2.ROM` or `GMASTER.ROM` is present in the program directory, fMSX will now load it as a system cartridge.
- Fixed state file name autogeneration.
- Fixed `-rom` option behaviour.
- Fixed hangup when switching from MSX1 to MSX2 mode.
- Fixed `MSXDOS2.ROM` and `FMPAC.ROM` support.
- Fixed `StateID()` function to take all cartridges and system ROMs into account.
- Fixed non-working joystick autofire.
- Now checking joysticks and mice once per frame.
- Now accelerating horizontal mouse movement in 512-wide screen modes.
- Mouse buttons are now merged with joystick buttons.
- Fixed mouse-as-joystick option.
- Improved file type detection, now automatically detecting saved states, disk images, cartridge ROMs, fixed font files.
- Readded proper support for fixed text mode fonts with `LoadFNT()`.
- Added proper emulation for the WD1793 FDC (`WD1793.c`).
- Added unified `.FDI` disk image support (`FDIDisk.c`).
- Added support for standard, SV738, Arabic, and Brazilian WD1793-based disk interfaces.
- Added support for GZIPped disk images.
- Made disk geometry discovery smarter (`CP/M-80 v2.2` now boots).
- Moved 720kB floppy utilities (`Floppy.c`) and SCC sound chip emulation (`SCC.c`) into `EMULib`.
- Added command line option to patch DiskROM calls (`-simbdos`). With this option on, fMSX takes over DiskROM disk access routines. When `-wd1793` is used, fMSX will simulate real WD1793 FDC instead.
- Rehashed built-in menu to make screen space for more options.
- Added menu options to select RAM and VRAM sizes, create new disk images, save disk images,

patch DiskROM calls, load and use fixed text mode fonts.

- Added virtual keyboard to fMSX-Symbian, invoked by pressing [EDIT] or left [SHIFT] key. The keyboard can work in three modes: "Off", "Single Key Entry", and "Multiple Keys Entry".
- Added display orientation settings to fMSX-Symbian ("Landscape", "Portrait", and "Auto").
- Finally fixed opendir() problem in EMULib-Symbian. All built-in menu file dialogs work now!
- Further extended and optimized Symbian screen rendering routines.
- Fixed and optimized TV scanline simulation in fMSX-Symbian, made scanline effect stronger.
- Now saving RAM and VRAM size settings in fMSX-Symbian.
- fMSX-Symbian now uses true 512-wide display drivers for SCREEN6, SCREEN7, and TEXT80.
- fMSX-Symbian will now save printer output to E:\Others\fMSX\PRINTER.OUT.
- Fixed system warnings in when flipping Nokia E70 open/closed.
- Fixed possible crash when selecting an initial file in fMSX-Symbian.
- Improved "About" box in fMSX-Symbian.
- Moved fMSX-Symbian file selection root to E:\.
- When initial file selection cancelled, fMSX-Symbian will now boot into plain BASIC (i.e. no cartridges and no disks).
- Renamed fMSX-Symbian binary and resource files to avoid name clash with fMSX/S60 port by Juha Riihimaki.
- Fixed mouse support in fMSX-MSDOS.

## New in fMSX 3.2

- Added ChangePrinter() function to set printer redirection file.
- Moved disk operations to a separate submenu of the built-in menu and added "Eject Disk" options.
- Added an easy "Load File" option to the built-in menu and the fMSX-Windows.
- Added options to redirect printer and sound logging output to the built-in menu and the fMSX-Windows.
- Added error messages to fMSX-Windows and fMSX-Symbian.
- Added frame rate display option to fMSX-Symbian.
- Added "fill screen" zoom option to fMSX-Symbian.
- Added MIDI soundtrack logging to fMSX-Symbian (saved into E:\Sounds and can be used as ringtones).
- Added configurable button and key mappings to fMSX-Symbian.
- fMSX-Symbian application menu is now split into three pages.

## New in fMSX 3.1

- fMSX-Windows and fMSX-DOS are now compiled with OpenWatcom C++ and use the new EMULib framework. Note that keyboard assignments have changed!
- Added hardware reset key ([F11]) to reboot MSX.
- Added universal screen buffer based Z80 debugger ([F10]) that works on all platforms.
- Added universal configuration menu ([F8]) that works on all platforms.
- Added ResetMSX() function to restart MSX, possibly in a new operating mode.
- Extended LoadCart() function to load or eject cartridges at runtime.
- Added Mode variable to store all configuration options.
- Replaced -vperiod/-hperiod settings with easier to understand -pal/-ntsc settings.
- Replaced -uperiod setting with a finer -skip setting that controls the percentage of skipped frames.
- Fixed Z80 NMI handling to comply with the standard.
- Fixed SRAM file saving.
- Fixed state file name generation.
- Added autofire options for joystick buttons.
- fMSX-Windows has got a new menu-based GUI, improved sound, and faster graphics updates.
- Fixed USB joystick support in fMSX-Windows.

- fMSX-MSDOS now uses 16BPP screen mode.
- Updated the documentation.

## New in fMSX 3.0

- Added transparent support for GZipped disk images as disks. You can now pass .GZ file names in `-diska/-diskb` command line options.
- Added transparent support for directories as disks (thanks to Miguel Colom). You can now pass directory names in `-diska/-diskb` command line options. Changes to such disks will not be saved for safety reasons.
- **Added DirectDraw-based full screen mode to fMSX-Windows.** Press [ALT]+[ENTER] to switch in and out of the full screen mode. Use it with caution, as DirectDraw is notoriously unstable.
- Added a full-fledged debugger to fMSX-Windows. Press [F11] to invoke the debugger.
- Fixed several bugs in MIDI implementations that made some games silent when logging soundtrack or using MIDI output in fMSX-Windows.
- Better adapted fMSX-MSDOS to 320x200 VGA screen.
- "Sync to VBlanks" option is now default in fMSX-MSDOS.

## New in fMSX 2.8

- Fixed initial envelope volume in `AY8910.c`, thanks to Koichi Nishida.
- Fixed a problem with 8kB/16kB cartridges failing to execute in the MSX1 mode.
- When guessing a MegaROM mapper type, fMSX now tries to look at a data file `CARTS.CRC` containing pairs of cartridge CRCs and mapper types.
- Added support for the third cartridge slot. You can now give up to three cartridge names on the command line and three `-rom` options.
- Cartridges are now inserted into slots 1:X (A), 2:X (B), and 3:0 (C). The `RS232.ROM` has been moved to slot 3:3.
- Added support for the 64kB Panasonic FM-PAC cartridge with 8kB SRAM (`-rom 7`).
- fMSX loads files `FMPAC.ROM`, `MSXDOS2.ROM`, and `PAINTER.ROM` automatically, as long as they are in the program directory and there is an empty cartridge slot available.
- Added support for ASCII/8kB cartridges with 8kB SRAM (Xanadu, Royal Blood). Please, [inform me](#) if this has broken support for any "normal" ASCII/8kB cartridges.
- Added support for ASCII/16kB cartridges with 2kB SRAM (Hydlide 2). Please, [inform me](#) if this has broken support for any "normal" ASCII/16kB cartridges.
- All battery-backed SRAM save files have now got `.SAV` extension. For example, `GameMaster2.rom` will produce a SRAM file named `GameMaster2.sav`.
- Added reporting of bad memory writes with `-verbose 8`.
- The *R-Type* cartridge and its variants are supported by the ASCII/16kB MegaROM mapper (`-rom 5`).
- HAL's *Hole In One Special* cartridge is supported by the GENERIC/16kB MegaROM mapper (`-rom 1`).
- Konami's *Majutsushi* (aka *Mah Jong 2* or RC765) cartridge is supported by the GENERIC/8kB MegaROM mapper (`-rom 0`), although the D/A audio circuit emulation is not implemented.
- Added fast-forwarding option (`[F9]`) and moved fixed font switch to a different key (`[CONTROL]-[F9]`).
- Changed `[F6]`, `[F9]`, `[F10]` assignments in fMSX-Windows to comply with other fMSX ports.
- Added `Options` menu to fMSX-Windows.
- fMSX-FreeBSD now compiles without changes on FreeBSD 5.x (FreeBSD 4.x users may have to change the location of the `soundcart.h`).
- Fixed sound in fMSX-FreeBSD by increasing the number of sound fragments from 8 to 64. If any of FreeBSD developers are reading this, could you explain why this is needed?

## New in fMSX 2.7

- Switched to the latest version of the OpenWatcom C/C++ compiler.
- The infamous Windows problem with spaces in directory names seems to be fixed by using a newer compiler.
- Made some arrays "const" to facilitate porting to run-from-ROM platforms (cellphones, PDAs, etc.).
- Joystick problems in fMSX-Windows are fixed.
- Decreased the default sound volume in fMSX-Unix.
- Increased the default wave synthesis frequency to 44.1kHz.
- Minor fixes to the Unix sound drivers.
- Removed MIDI volume control from Windows sound drivers.
- Added TV raster effect to fMSX-Windows.

## New in fMSX 2.6

- Added separate (and correct) emulation for the i8255 PPI chip.
- Added masking to lower bits of video table addresses.
- Added timing and other fixes to the Z80 core (courtesy of Omar Cornut).
- Fixed line coincidence and VBlank handling (thanks to Vincent van Dam).
- Save state (.STA) file format has changed.
- Fixed a bug that messed up screen table addresses when restoring state.
- Fixed a bug that left dummy memory page dirty when loading GZIPped ROMs.
- Fixed a -home option bug on Solaris (thanks to Eric Boon).
- Fixed fMSX-Unix to allow it to have an application icon under WindowMaker.

## New in fMSX 2.5

- Added emulation state saving and loading.
- Added support for GZIPped ROM files and state file.
- Changed keyboard handler in fMSX-Unix. Hopefully, the key assignments should now be more natural.
- Added sound on/off switch to fMSX-Unix and fMSX-MSDOS.
- Keyboard assignments changed everywhere.

## New in fMSX 2.4

- Small fix to the DI instruction.
- Small fix to the HALT instruction.
- Added -sync option to fMSX-MSDOS on public request.
- Fixed PHYDIO DiskROM call (4010h) to return the number of remaining sectors in register B (thanks, Maarten!).
- Fixed ASCII mappers to prevent memory corruption when switching ROM pages in disconnected slots.
- Fixed palette to span to true white.
- Added SetWave() call to the sound API.
- SCC emulation now makes use of waveforms via SetWave().
- SCC gets silenced when 0 frequency is written (thanks, Maarten!).
- Fixed SCC+ emulation.
- Vertical screen adjustment is only computed once, when refresh starts.
- Moved picture 10 pixels down when the screen is 192 pixels high.
- Removed unused SprCol variable.
- Added wave synthesis sound to fMSX-Windows.

- Fixed fMSX-MSDOS to restore palette when exiting the debugger.

## New in fMSX 2.3

- Now, when you buy fMSX-Windows, you also fMSX-MSDOS.
- Added modular FM-PAC (aka YM2413 aka OPLL) emulation.
- FM-PAC soundtrack can be logged into a MIDI file.
- Sound and MIDI logging APIs have been integrated and rewritten to accomodate MIDI instruments, dynamic instrument changes, drum usage, etc.
- Added PAL/NTSC switch on a bit in VDP[9] (thanks, Alex!).
- Added support for the overscan (Mantra demos and games run now).
- Added sprite support to SCREEN3 (no idea why it was not there).
- Added support for YJK (SCREEN12) and YAE (SCREEN10/11) screens.
- Added separate handling for IE0 and IE1 IRQs (Xevious, etc.).
- Added masking of Character Table address (SourceOfPower demo runs now).
- Fixed Z80 interrupt handling after EI (thanks, Maarten!).
- IE0 and IE1 IRQs are now reset when disabled.
- IE0 IRQ is now set when enabled.
- Palette sequencer now resets when writing to VDP[16].
- Fixed mapper #1 to support generic 16kB-paged MegaROMs (HoleInOneSpecial).
- Fixed cartridge loader to correctly load MegaROMs with "AB" signature in the last 16kB page (R-Type).
- Mappers #0 and #1 are now renamed to Generic/8kB and Generic/16kB.
- Fixed mapper #3 to have hardwired ROM at addresses 4000h-5FFFh (thanks, Sean!).
- Added mapper #6 that corresponds to the Konami GameMaster2 cartridge with SRAM (thanks again, Sean!).
- Added key click and motor relay click emulation via Drum() call.
- Major rewrite of the sound drivers and sound logging system.
- Added mouse emulation to fMSX-MSDOS.
- Added border emulation via VGA border to fMSX-MSDOS.
- Added built-in debugger to fMSX-MSDOS (F11).
- Fixed static palette in fMSX-MSDOS.
- Fixed fMSX-Unix to work in 24BPP and 32BPP modes.
- Added static palette to fMSX-Unix.
- Minor fixes to fMSX-Unix makefile.
- fMSX-Windows now runs in 16BPP HiColor mode.
- Changed Setup Panel in fMSX-Windows.
- HPeriod and VPeriod scrollbars are gone from fMSX-Windows as timing control is done automatically.
- PAL and NTSC buttons are gone from fMSX-Windows as PAL/NTSC switch is done automatically.

## New in fMSX 2.2

- Interrupt system rewritten.
- Mapper management rewritten.
- Keyboard assignments changed again. Check the documentation!
- Made a new MSDOS port of fMSX (still in development).
- Integrated modular AY8910 emulation.
- Integrated modular SCC emulation.
- Integrated SndUnix/SndWin/SndMSDOS sound drivers.
- Sound track is now logged to MIDI files.
- Added a table of I/O handler functions to Z80 core for easy expansion.
- Added ability to load flat (no mapper) 64kB ROMs.

- Added automatic MegaROM mapper guessing (-rom 6).
- Added standard PAL/NTSC timings.
- Added V9958 version flag in MSX2+ mode (VDP S1).
- Added correct emulation of HR/VR flags (VDP S2).
- Fixed border color in SCREEN8 (VDP R7).
- Fixed blinking frequency in TEXT80 (VDP R13).
- Added ADJUST register emulation (VDP R18).
- Added V9938 character scrolling to SCREEN0 and TEXT80 (VDP R23).
- Added scrolling to SCREEN1, SCREEN2, and SCREEN3 (VDP R23).
- Fixed scrolling bug in SCREEN6 and SCREEN7 (VDP R23).
- Made left/right borders 9/7 pixels in SCREEN0 and TEXT80.
- Added -home option to tell fMSX where to find system ROMs.
- Added -auto option for auto fire on pressing SPACE.
- Fixed information printed on -help.
- Fixed crash in fMSX-Windows when switching from 8kB to 16kB MegaROMs.
- Fixed trash in MSX2 logo in fMSX-Windows that appeared after reset.
- Better resizing in fMSX-Windows Setup panel.

## New in fMSX 2.1

- New **fMSX-Windows** available. Buy it **now** !
- Rewritten V9938 graphical operations support.
- Rewritten initialization sequence.
- Improved real-time clock emulation.
- Fixed a problem with color sprites (Zanac-Ex, Aleste).
- Fixed problems with memory allocation.
- Fixed mouse offset in fMSX-Unix/X.
- Extended documentation, added more questions to FAQ.
- Reduced window size in Unix/X to speed up updates.
- fMSX-Unix now supports 16bpp and 32bpp Xservers for sure.
- Added speed throttling to fMSX-Unix (see -sync).
- fMSX-Unix runs in a resizable window now.
- Fixed IPC key allocation bug in fMSX-Unix.
- Many small fixes to the code.

## New in fMSX 2.0b

- Major code rewrite!
- Line-by-line screen updates implemented
- Standard modular Z80 emulation
- Support for 16bit and 32bit X11 screens (possibly 24bit)
- Support for KANJI.ROM
- Mouse support
- Joystick support
- Multiple bug fixes
- Multiple new bugs (look for them!)

## New in fMSX 1.5

- More stable split screen (Zanac-Ex no longer blinks)
- RAM mapper fixed again (Designer+ works now)
- V9938 SRCH command fixed
- MSXDOS2.ROM support



- PAINTER.ROM support
- FMPAC.ROM support
- .PSG file format is updated
- Serial port support is not yet implemented

## Thanks to...

### **Hans Guijt [.NL]**

Working on Amiga port, Hans has done a very good job in getting fMSX run fast on an Amiga. He rewrote the CPU emulation and the screen drivers in assembler, optimized the VDP emulation, wrote sound drivers, and added many more things making fMSX-Amiga the best, if not the fastest MSX emulator for this platform. Thanks, Hans (and yes, I do remember those .ROM files too ;)).

### **Igor Sharfmesser [.KZ] and Alex Krasivsky [.RU]**

Igor and Alex have brought to life fMSX-MSDOS. The first version, including screen and keyboard drivers, was done by Igor. AdLib sound was added by Alex.

### **Teturo Tokuyama [.JP]**

Teturo has done an excellent port of fMSX to Windows. As much as I hate Windows, I must admit that Teturo's fMSX-Win32 rules. Period.

### **Murakami Reki [.JP]**

Murakami has written PC9801 port of fMSX. He is the guy whom you have to thank for the disk support in fMSX. If not for him, I would have never started implementing it :).

### **Marcel de Kogel [.NL]**

Marcel wrote an excellent port of fMSX to MSDOS. His MSDOS version of fMSX was actually the first one that supported all fMSX features correctly. Marcel also provided invaluable help debugging and improving fMSX and other emulators.

### **Ville Hallik [.EE]**

AY8910 and SCC chips emulation, written by Ville for Linux/FreeBSD /dev/dsp device, and SunOS /dev/audio, was the best fMSX sound emulation I have seen at the time. And, yes, it lets you play MSX games with sound on a Unix system! =:)

### **John Stiles [.US]**

John took over the development of the Macintosh version of fMSX and managed to speed up the common part of display drivers used in the Unix/X, MSDOS, Windows, and Macintosh versions.

### **Paul Chapman [.CA]**

Paul Chapman is the original author of fMSX-Macintosh. Although his port had bugs, it looked great and ran faster than the original MSX computer on an average PPC-based Macintosh of that time. As Paul was not able to continue developing fMSX-Macintosh, the development has been taken over by John Stiles.

### **Arnold Metselaar [.NL]**

If you happen to have a non-standard X11 display, get Arnold's X11 drivers fixed to work for any screen depth. Arnold has also added to his drivers a lot of improvements that are not in the standard fMSX distribution. Make sure to check out his work.

### **Guenter Woigk [.DE]**

Found a bug with offsets in Z80 commands using IX/IY registers.

### **Martial Benoit [.FR]**

Explained how VDP deals with sprites and sent me copies of V9938 and WD2793 databooks

(WOW! Thanks, Martial :)).

**Sean Young [.NL]**

For finally making it clear to me how GameMaster2 SRAM works and also for finding a bug in my Konami4 mapper emulation.

**Alex Wulms [.NL]**

Alex has written the most accurate emulation of the V9938 graphical operations to date. This emulation is now a part of fMSX. Also, Alex explained to me a lot of technical details of VDP operation. And he sent me a copy of the WD1793 databook :).

**Maarten ter Huurne [.NL]**

Maarten has done extensive bug-hunting on things that I would never consider to run (that is, demos :)). He also looks for glitches in games and doesn't rest until he gets to the source of a problem. Additionally, he has implemented MSX Serial I/O emulation that is slowly making its way into fMSX. And finally (as the stuff above were not enough) you can thank Maarten and Takamichi Suzukawa for the English translation of the Konami's Solid Snake cartridge!

**Miguel Colom [.ES]**

Miguel has come up with an idea and the code to use directories as if they were MSX disks.

...and to all other people who helped me with advice, information, and encouragement.

**ENJOY THE EMULATOR AND LET US KEEP MSX ALIVE :)**

---

© Copyright by [Marat Fayzullin](#) (*marat [AT] komkon <DOT> org*)