fmsx

COLLABORATORS			
	TITLE : fmsx		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

fmsx

1.1 "

```
                    fMSX Amiga 1.4

© 1994 by Marat Fayzullin.
© 1995 by Hans Guijt.

General information
```

1.2 "

Introduction to fMSX

fMSX stands for Fast MSX, and that’s what it is: a reasonably speedy emulation of the MSX computer system.

It was originally programmed by Marat Fayzullin as a generic MSX emulator - that means that it can be compiled on any machine, in any environment! To reach this goal he programmed fMSX in C code, resulting in a program that runs on UNIX machines, PCs, Macs, and of course Amigas.

At some point I (Hans Guijt) offered to rewrite the screen drivers for the Amiga, and Marat accepted this offer. I received the sources and went to work.

The first thing I noticed was that the emulation was exceptionally slow! This is okay for a fast UNIX workstation, but most Amiga's are low-end, and the emulation would be next to useless in such an environment.

The solution seemed easy: just rewrite the Z80 emulation in optimized 68020 assembler code (easier said than done!). The results were spectacular. A short BASIC program was selected as the benchmark:

```
10 FORT=1TO10000
20 NEXTT
```

Machine	Time
A4000/30	94 seconds (fMSX 0.3, original version)
Sun Sparc 2	86 seconds (fMSX 0.4, UNIX version)
Sun Tatung	43 seconds (fMSX 0.4, UNIX version)
a4000/30	22 seconds (fMSX 0.3, modified version)
VG-8235 MSX2	21 seconds
a4000/40	15 seconds (fMSX Amiga 0.2, standard settings)
a4000/40 (warp 4040)	9 seconds (fMSX 0.3, modified)
MSX Turbo-R/RAM mode	6 seconds

Note that this chart is not particularly valid anymore. The Z80 emulation in fMSX v0.7 was rewritten almost from scratch, causing incredible speedups in some operations (bankswitching and memory mapping) and some slowdown in others (memory reads/writes).

The end result is that BASIC programs run quite a bit slower (BASIC is fairly memory intensive), but games run faster (and that's what counts).

1.3 "

The Philips VG-8235 MSX2

This is my own MSX machine. It is from 1985, and is equipped with a single sided 3.5" diskdrive. It is an MSX2 machine, but unfortunately the VDP (video chip) is broken, so I lose half the screen once the machine starts to get warm.

I have bought a new machine! It is a Sanyo PHC-70FD, which is an MSX2+ machine with several nice features:

- 256KB RAM (expanded from the original 64KB)
- Numeric keypad
- Joystick autofire control
- Double-sided diskdrive, which is turned to the front of the machine (rather than the side as is usual on many other machines)
- Pause button
- Built-in FM Pac

As an added 'bonus', it is equipped with a Japanese keyboard!
That means that every key carries four or five different symbols, including the European glyphs.

1.4 "

The MSX Turbo-R

The newest MSX system, the Turbo-R is equipped with a very fast CPU: the R800, which is comparable to a Z80/28MHz. It can do multiplications faster than the 68030 in my a4000!

The Turbo-R has a slow ROM system, and the designers made it possible to copy ROM to RAM and use that instead. In the speed test this method was used.

1.5 "

Disclaimer

THIS SOFTWARE IS PROVIDED AS-IS. NEITHER
AUTHOR
WILL BE HELD RESPONSIBLE
FOR ANYTHING IT DOES, WHETHER RIGHT OR WRONG, GOOD OR EVIL, LAWFUL OR CHAOTIC. RUN AT YOUR OWN RISK!

1.6 "

The MSX system

The MSX originally came out in 1983, with the intention to create a low-cost low-performance all-compatible computer system (a gap that is now filled by the PC, ironic when you consider what MSX stands for):

MicroSoft eXtended

which means that it is an enhanced version of earlier Microsoft products, notably their BASIC and DOS (that’s right, MS-DOS!).

The MSX1 standard

Among other things, the MSX standard specifies that machines should have at least a Z80/3.5Mhz, 16KB RAM (although most machines had 64KB RAM), 16KB video RAM, and the MSX ROM’s. Graphics, produced by the v9918, are functional:

screenmode	resolution	colors
0	40*24 characters	2
1	32*24 characters	16 (with restrictions)
2	256*192 pixels	16 (with restrictions)
3	64*48 pixels	16

In addition, screens 1-3 can have single-colored sprites, either 8*8 or 16*16, either at normal size or enlarged.

Sound is delivered by the AY-3-8912, a three-channel sound chip also used in the Spectrum 128K and the Atari ST.

One thing that sets MSX machines apart from other comparable systems is the ability to run cartridge software - most machines can, but for MSX the system was very popular. Although I have never seen one, it should be possible to build an MSX console.

Another unique feature is the popularity in both Japan and western Europe. This resulted in a software catalogue filled with the best of both worlds, a feature that always attracted me to the system.

Finally, it must be mentioned that MSX is not the product of a single manufacturer. Instead, MSX is a minimum standard decided upon by ASCII corporation, with individual manufacturers licensing and adding to the system, much like 3D0 today.

There have been over 40 MSX manufacturers, and MSX machines were manufactured by Daewoo in Korea after Commodore went broke in 1994!

The MSX2 standard

In 1985 the MSX2 came out, which sported several large improvements. The most remarkable is the graphics, provided by the v9938; it offers all the MSX1 modes, and several others:

screenmode	resolution	colors
0+	80*26.5 characters	4
4	256*192 pixels	16 (with restrictions)
5	256*212 pixels	16
6	512*212 pixels	4
7	512*212 pixels	16
8	256*212 pixels	256

Other improvements to the graphics include an enhanced sprite system, which allowed for more sprites with more colors, a color palette of 512 colors, support for genlocking and digitizing, smooth vertical scroll, and a built-in blitter which can do almost anything the Amiga blitter can, and more.

The rest of the system had been updated as well. MSX2 machines are equipped with at least 64KB RAM (128KB and 256KB are more common configurations), 128KB video RAM, and usually a 3.5" diskdrive.

The MSX system uses exactly the same disk layout as MS-DOS. In fact, CrossDos can be used to read/write MSX disks, even single sided ones (my compliments to the authors for that), while modern PC's have lost that ability!

MSX2+, MSX Turbo-R

These are newer versions of the MSX system. New features include better graphics (the MSX2+ VDP is capable of displaying 19268 colors at once on screen), and a faster processor (the Turbo-R has an R800 CPU, comparable to a Z80/28MHz).

This document used to say they were not on the list for emulation. Today I'm not so sure - if it is remotely possible to run Illusion City on fMSX I will do it.

1.7 "

COPYRIGHT

The following notice is taken from fMSX UNIX. It applies mostly to fMSX Amiga as well.

fMSX is publicly supported freeware. Its sources are on the net, and you can modify them as long as you notify me about modifications. You can't use fMSX for commercial purposes though. If you want to market anything based on fMSX source or executables, contact me please.

The 'me' referred to here is either
Hans
(for the Amiga version) or
Marat
for any other version.

The notice is incorrect in that I do not regularly upload fMSX sources for the Amiga version to any public place. However, if you are interested, you can obtain them by mailing to me.

If your conscience does not allow you to work with software you did not pay for, neither author refuses gifts, money or postcards. I'd appreciate it very much if you were to send me email!

1.8 "

Installation

No specific installation procedure is required, just drag the directory containing fMSX to some place of your liking. You can scatter the files in the subdirectories all over your system if you want but you'll have to change some of the default pathes if you do.

Note: do not copy the MSXFonts/ directory to FONTS:! The fonts in that directory are in MSX format, not in Amiga format.

To run cartridge images from the Workbench

Many games come with pre-made icons. These icons require that fMSX: is assigned to the directory where you installed fMSX.

1.9 "

System requirements

Required are:

- Amiga OS 2.0
- 68020 processor
- 350 kilobytes chip ram
- 1000 kilobytes other ram

Recommended:

- Amiga OS 3.0
- 68030 processor, at least 25MHz
- Those 1000 kilobytes of 'other' ram had better be fast ram!

If you want to use MSX disks, you will need to have CrossDos or similar installed. CrossDos is part of the AmigaOS since v2.1. Shareware alternatives are available on Aminet.

If you want to use the shared PSG/SCC soundmode, you will need to have AHI v4 or better installed. AHI is available from Aminet.

This program was developed using:

- An a4000/30 with 14Mb RAM and 260Mb HD
- SAS/C 6.57
- GenAm 3.02
- TurboText 2.0

1.10 "

Running the emulator

From the shell

Type fMSX to start the program. It is possible to load a cartridge file by typing fMSX <ROM-NAME>, which will load the specified cartridge. A full pathname may be used when specifying the cartridge.

From workbench

Double click on the fMSX icon to start the emulator without loading any cartridges.

Loading a cartridge is accomplished by double-clicking it. Note that the included game icon requires the directory fMSX: to be assigned to the directory where the emulator resides. You will need the shell to create this assign.

Restrictions

As an MSX1 emulator fMSX Amiga is fairly complete. However, many of the MSX2 features are still missing. All screens are supported except for screen 4 and the interlace screens. Still left for later implementation are sprites for screen 6 and 8, line interrupts for all screens except screen 5, and screen positioning.

The following sort of software will run:

- Standard ROMs (16KB or 32KB)
- MegaROMs (128KB or bigger)

These are not physical cartridges, but copies that are stored on the harddisk of your Amiga. They can be obtained from several
FTP
sites.

- Disk-based games
- Disk images
- Tape images

These are also available through
FTP
, from the same sites.

1.11 "

The control window

The emulator can be controlled from a control window on the workbench screen. This window gives you the following options:

Refresh cycle

This is the number of frames skipped before a new frame is drawn. If the refresh cycle is has a low value the emulation runs smooth and slow. If the refresh cycle has a high value it runs faster but also less smooth. Try to experiment with what works best for a particular game. Usually a value of about 5 gives good results.

This value is ignored when running in
high-speed mode

.

Interrupt period

This controls the number of operations between interrupts. In interrupt driven games, setting this to a low number speeds the game up, while high numbers slow it down. If the program you run does most of its work outside the interrupt, setting it to a low number may actually slow things down.

In addition to this, the emulator waits for at least 1/50th of a second before issuing an interrupt.

CAUTION: if this number is set too low, it may crash the MSX. Try again with a larger number. A value of 1500 should always be safe.

Again, experiment for the best results. I recommend a value of 1500 when running BASIC, and 800 when running games.

Cartridge 1

Here you can specify the name of a cartridge file. It will be loaded and executed when you reset the MSX.

ROM type

This button specifies the memory mapping method that will be used for the current megaROM. If this value is not set correctly the megaROM will certainly not run! Note: a megaROM is a cartridge bigger than 32Kb.

If you do not know the correct setting for a megaROM you have no option but to try them all. However, a list of known settings is available here.

Load

This button calls up a filerequester which you can use to select a cartridge file. The cartridge file is loaded when you reset the MSX.

Lock drives

If this button has been checked the emulation can access the drive, but the Amiga is locked out. Similarly, if it is not checked, the Amiga can access the drive but the MSX cannot.

Running / Paused / Music

If you need every last cycle your machine can provide, but are unwilling to quit fMSX because you just reached level 48 of Tetris, you can pause and restart the emulation with these buttons. When it is paused it takes absolutely no CPU time.

You can pause fMSX from the MSX screen by pressing the * key on the numeric keypad.

Music mode allows you to listen to music produced by the MSX without the overhead of constant screen redraws. This mode makes fMSX a reasonably effective music player.

Reset

This button resets the MSX. It is necessary to reset the MSX for some settings to take effect.

1.12 "

The system preferences window

This window allows you to specify several system related settings for the emulation.

Device A: and B:

In these gadgets you have to specify which Amiga device or disk image is used for emulation of MSX drives A: and B:. Normally these would be PC0: and PC1:, but you can also create an MS-DOS compatible disk image or ramdisk and use it instead.

If you specify a device, it must have certain properties:
- It must have a blocksize of 512.
- It must have a low-cyl of 0.
- It must have a high-cyl of 39 or 79.
- It must have 9 blocks per track.
- It must be a real device, no assign or volume.
- It must be mounted.

It is also possible to use disk images without the help of an external device such as fmsdisk.device. To use a disk image, just type the name (including the full path) in one of the gadgets. A disk image must be either 368640 or 737280 bytes long.

It is not necessary to fill in a value in these gadgets, but if you don't you cannot use the corresponding MSX drive.

Two alternative MSX devices are provided in the archive.
RAMSX:
uses
memory as an MSX diskdrive while
FMSX0: and FMSX1:
are based on
fmsdisk.device (not included but available on Aminet). They provide permanent MSX disks stored on the Amiga harddrive.

Load

There are two buttons marked 'load', one for each MSX device. They let you pick a diskimage or device from a requester instead of typing the name into the gadget.

Save preferences

When turned on, fMSX preferences are saved when the program ends. When turned off it is necessary to save preferences by hand, using the menu option Preferences/Save as... or Preferences/Save default.

Japanese ID

When turned on, the country ID bytes in the ROM files are set to Japanese. This causes some games to act differently. Examples:

- Nemesis 1 gets a new name (Gradius).
- Nemesis 2 uses Japanese text.
- Nemesis 3 gets an animated title screen.

Plug'n'Play

When turned on fMSX will reset automatically after you select a cartridge from the filerequester or when you drop a cartridge on the fMSX window. This way you get to play it even faster ;-)

SCC channel disable

When selected, fMSX respects the SCC channel disable settings. Turning it on causes some games to sound bad while turning it off causes other games to sound bad.

As a guideline, it should be turned on for Parodius (which experiences random beeps without it), and turned off for most other SCC games. Experiment to see what sounds best.

This button has no influence on the PSG emulation, only on the SCC emulation.

MSX version

Selects what version of the MSX ROMs is used. MSX2 games require MSX2 ROMs to work. MSX1 games will run fine with MSX2 ROMs, but you may want to use MSX1 ROMs because of the shorter boottime.

Memory

With this gadget you can select the amount of memory the emulated MSX uses. Note that the amount of free BASIC memory is constant (about 23KB), no matter how big a number you put here. The biggest you will ever need is the 512KB setting. The other settings are provided to let you boast about the amount of memory your MSX can use ;-) . And remember, this is for only one memory mapper! A real MSX can use 14 for a total of 56MB!

AHI mode

This button lets you select the AHI mode fMSX uses. AHI is used in the combined PSG/SCC soundmode, and the AHI mode determines the quality of the sound in that mode.

Selecting higher frequencies increases the quality of the sound. Unfortunately higher frequencies require more CPU power as well, causing fMSX to run slower. Experiment to find a good setting.

You should not select a stereo mode, as fMSX only generates mono sound.

If you don’t have AHI installed this option is not available. AHI can be obtained from Aminet.

SCC+

Some SCC games require an updated SCC chip, called SCC+. This button enables SCC+ emulation (and disables normal SCC emulation). The only games that require SCC+ that I know of are S.D. Snatcher and The Snatcher.

Drives

With this gadget you can specify the amount of drives connected to the MSX. Note that two drives need more MSX memory than one drive. Some games will not run if you have two drives connected due to memory shortage.

This setting only takes effect when you reset the MSX.

Soundmode

With this gadget you can select between various soundmodes. These soundmodes are:

Off: fMSX is not producing any sound, either because that is what you wanted or because some other application is currently using the Amiga’s sound hardware.

PSG: this is the native soundchip of the MSX. It has three tone channels and one noise channel. PSG emulation is virtually perfect: it will even emulate sampled sound (usually produced through volume modulation of the noise channel). PSG stands for Programmable Sound Generator.

SCC: this is a sound chip used in some Konami games. Instead of blockwaves, like the PSG, the SCC uses 32-byte samples which sounds much better. It has 5 channels, of which only 4 are emulated in this mode.

Games that support SCC sound can easily be recognized by the following features:

- The game is produced by Konami.
- It is a ROM of 128KB or bigger (only exception to this: S.D. Snatcher).
- PSG sound for this game consists of nothing more than beeps and clicks.

Examples of games that support SCC sound are:

MSX1 games
Salamander
Nemesis 2
Nemesis 3
King’s Valley 2
F1 Spirit
Parodius

MSX2 games
Space Manbow
Metal Gear 2
Quarth
Gryzor
King’s Valley 2 MSX2
The Snatcher (actually SCC+)
S.D. Snatcher (actually SCC+)

PSG/SCC: in this mode fMSX emulates both the 4 PSG channels and the 5 SCC channels, allowing you to hear the (often spectacular) SCC songs the way they were supposed to be heard. If you like Konami’s musical style (and who doesn’t?) you should really install AHI and listen to the songs in this mode.

This mode requires AHI to be installed on your Amiga. AHI is available from Aminet.

1.13 "

The video preferences window

This window allows you to specify several video related settings for the emulation.

Only refresh when active

When this option is turned on, and the MSX window is deselected, the emulator will no longer redraw the MSX screen. On most games this means that the music suddenly runs at normal speeds, thus turning fMSX into a music playing system!

This is not quite the same as music mode, which also closes the MSX screen.

Double buffering

Turns double buffering on and off. It is a good idea to keep double buffering on at all times, and only turn it off when you really cannot afford the memory.

When using OS 3.0 the OS specific double buffering routines are used. However, these routines are not available on older versions of the OS. If you are still using an older OS version you may still experience some flicker when using double buffering. This has to do with timing of the screen redraws and cannot be avoided.

CyberGfx does not emulate OS3.0 double buffering, therefore you must turn it off if you are running on a CyberGfx screen.

Hide titlebar

Turns the titlebar of the MSX screen on and off.

Respect VDP blanking bit

When selected fMSX respects the blanking signal of the MSX videochip. Some games turn blanking on for very short periods during a frame, which can cause fMSX to accidentally blank the whole display. If this happens you can turn this option off.

Load colors directly

Some graphical effects rely on quickly changing the colors several times per screen. Normally the Amiga does not allow this (colors are only changed once per screen by Intuition) but if this option is set any color changes are also directly poked into the hardware registers. This may cause visual confusion on other screens as well, but is in fact totally harmless.

Use anchor screen Requires OS 3.0

Under OS3.0 (and up) it is possible to keep screens adjacent at all times. fMSX can use this feature to keep it's own screen at a place of your choosing (as opposed to popping it to the front of the display everytime some screen attribute changes).

When you activate this feature an extra screen is opened, called the anchor screen, which never changes. When the MSX screen is closed and reopened it will appear directly in front of the anchor screen.

The screen selector

The gadgets on the right hand side of the window are used to set the correct display mode. When you select an MSX mode in the listview at the top, it's name and attributes become visible at the bottom. Screenmodes marked with a '*' are not available in this version of fMSX.

If you are using the v38 ASL library you will see a button marked 'Select'. Pressing this button brings up the ASL screenmode requester. You should choose a screenmode that fits the MSX screenmode you selected in the listview.

If you are using the v36 or v37 ASL library you will see a cycle gadget in which you can choose a screenmode. Not all screenmodes are available from the cyclegadget. Consider it one more reason to upgrade to 3.1 ;-)

Note that a different Amiga screenmode can be set for every MSX screenmode. You cannot just select a single screenmode and let fMSX do the rest.

1.14 "

The tape preferences window

In this window you can control how fMSX deals with tape images. Tape images are stored by fMSX Amiga as a collection of files, similar to the way a real MSX stores files on tape. However, files on tape are ordered by nature, which is not quite true for files on disk.

To emulate the ordering of files on a tape fMSX uses an index file. This is a simple text file which lists all the files in the right order. The index file should be in the same directory as the files it indexes. For that reason it only needs to list filenames, without path information.

Load

The load button is used to load the index file of a tape. After selecting an index file the contents of the tape are shown in the listview. It is also possible to type in a pathname directly.

You can select a name that doesn't yet exist. If you do so, a new index will be created.

Append

The biggest problem with writing to a real tape is positioning it. No matter how careful you are, the day will come when you accidentally overwrite something important.

The ’append’ option prevents this. When turned on, new files are always appended to the end of a tape. When turned off, new files will overwrite the currently selected file.

The tape contents listview

The listview shows the contents of the currently selected tape. The tape is positioned just before the highlighted item. This means that the next item loaded by the MSX (or saved, if ’Append’ is turned off) will be the selected item.

It is possible to change the name of an item. To do this, select it and type the new name in the stringgadget at the bottom of the window. It is strongly recommended that you do this for newly created files. fMSX searches for possible new filenames by simply going through a range of filenames and trying them all. The more filenames exist in this range, the longer the search takes.

Up

This button moves the currently selected file upwards in the index, ie. forwards on tape.

Down

This button moves the currently selected file downwards in the index, ie. backwards on tape.

Mark

This button marks the currently selected file for deletion. It will be deleted when you press the ’Remove’ button.

Clear

This button unmarks the currently selected file.

Remove

The Remove button removes all marked files from tape. The files are removed from both the index and your harddisk.

1.15 "

The menu bar

Project/About...

Calls up the ’about’ requester.

Project/Quit

Quits the program.

Cartridge/Open...

Brings up a filerequester where you can choose a cartridge file for use in the cartridge slot. The cartridge is loaded when you reset the MSX.

Cartridge/Save config

Saves several settings in the icon of the current cartridge. If the cartridge does not have an icon, the fmsx_rom.info icon (in the program directory) is used. If this icon does not exist either, no information is saved.

You can also edit the icon manually.

Diskdrives/Write bootblock drive A:
Diskdrives/Write bootblock drive B:

The MSX cannot boot from disks that have a PC bootblock. With these menu options you can write an MSX bootblock to one of the MSX drives.

CAUTION: some games have their own bootblocks. Overwriting them will kill the game.

Diskdrives/Create 360KB diskimage
Diskdrives/Create 720KB diskimage

With these options you can create a new diskimage. Select the size you want, pick a location and name in the filerequester, and it’s ready for use. There is no need to write a bootblock to a diskimage created in this manner.

Preferences/Open...

Opens a previously saved fMSX preference file.

{fg shine}Preferences/Save as...

Saves a preference file to a location of your choosing. You can reopen such a file later by selecting the Preferences/open menu option.

Preferences/Save default

Saves fMSX preferences. Preferences are saved to the file fmsx.prefs in the ENV: and ENVARC: directories.

Preferences/Pathes...

Opens the pathes window.

Preferences/System...

Opens the system preferences window.

Preferences/Tape...

Opens the tape preferences window.

Preferences/Video...

Opens the video preferences window.

1.16 "

Icon tooltypes

The optimal settings for a cartridge can be stored in its icon. To find out what each tooltype means, read the section about the control window .

Currently the following tooltypes are supported:

REFRESHCYCLE=value

Legal values are 1 to 10.

INTERRUPTPERIOD=value

Legal values are 0 to 5000.

ROMTYPE=value

Legal values are 0 to 5. These values mean, in order: SCC 1, Konami 1, ASCII 1, ASCII 2, SCC 2, and Konami 2.

SOUNDMODE=value

Legal values are -1 to 2, which stand for: no sound, PSG sound, SCC sound, shared PSG/SCC sound.

MSXVERSION=value

Legal values are 0 and 1, which correspond to MSX versions 1 and 2.

SCCPLUS

Legal values are 0 and 1, which disables or enables SCC+.

SCCCHANNELDISABLE

Legal values are 0 and 1, which disables or enables SCC channel disabling.

1.17 "

The pathes window

This window lets you specify pathes to several external resources.

Enable external ROMs

If turned on fMSX will load the MSX system ROM files from disk instead of using the internal ones. You will need to specify what ROM files get loaded using the appropriate string gadget. Note that internal ROMs are always used when the external ROMs are not available.

Enable Kanji ROM

When turned on, fMSX loads the Kanji ROM file specified in the stringgadget marked 'Kanji ROM'. This is a 128KB ROM that contains definitions for 4096 Kanji characters. Not many programs require this ROM, and even then you need to be able to understand Kanji before it is of any use.

If you see 16x16 blocks of solid color in places where you would expect characters, try enabling the Kanji ROM.

Enable external fonts

When activated an external MSX font is substituted for the internal ROM font. If you can read Japanese, selecting the Japanese font makes many games readable.

Currently four fonts are available:

Cyrillic This is a font containing Cyrillic (Russian) characters.
Default This is the same font that is present in the internal ROMs.
Italic This is the same as the default font, but has italicized characters.
Japanese This is a font containing Japanese characters.

Note that the fonts are in MSX format, not in Amiga format. The two font formats are not compatible with each other.

1.18 "

Using MSX diskdrives

The MSX disk system is an extension of the original MS-DOS disk system. Anything you know about MS-DOS probably applies to MSX as well. This section gives a short overview of features.

MSX filenames, like MS-DOS filenames, are very limited: they have a maximum length of eight characters, followed by a three-character extension. In addition, no lower-case characters or spaces are allowed.

The number of files on a disk is limited to 112. To make matters worse, subdirectories are not supported by MSX.

There are two wildcards: * and ?. These correspond to the Amiga wildcards #? and ?.

Examples:

```
*.*   Everything
*.BAS   All files with extension .BAS
GAME*.*   All files starting with GAME
*.B??   All files that have a B as the first part of the extension.
```

fMSX Amiga supports up to two diskdrives. The MSX calls these drives A: and B:.

There are two ways of dealing with disks on the MSX: by using
                MSX-DOS
                and from
                BASIC
                .

1.19 "

About MSX-DOS

MSX-DOS is a control program for the MSX which is based on the very first version of MS-DOS. It is required if you want to use some MSX programs, and it can be useful if you want to perform certain floppy operations. It is close enough to CP/M that some CP/M programs run on it; many others only require minor modifications to run.

MSX programs that require MSX-DOS can easily be recognized by their filename. Programs that end in .BAT are MSX-DOS script files; programs that end in .COM are MSX-DOS executables. Both types can be started by typing the filename (without the extension) after the prompt.

Getting started

To run MSX-DOS you need three things: an MSX-formatted floppy, and the programs MSXDOS.SYS and COMMAND.COM. Obtaining an MSX-formatted floppy is easy: any 720KB PC floppy will do.

Step 1: Find, buy, or format a PC floppy. Assuming you have CrossDos installed this can easily be done from your workbench.

Step 2: Open the Preferences window by clicking on the Preferences button, and check that the Drives button is set to 1 or 2. If it isn’t, change the setting and reset fMSX by clicking the Reset button in the control window.

Step 3: Convert the disk to MSX format by selecting the menu option Write bootblock (for the appropriate device).

```
Step 4: Copy MSXDOS.SYS and COMMAND.COM to this floppy.

Step 5: Reset fMSX by clicking the Reset button.

Step 6: Wait until MSX-DOS has booted. One of the first things you'll
see is the prompt, which looks like this: A>

When MSX-DOS boots it looks for a file called AUTOEXEC.BAT. This is
a script file that is executed automatically if found.

  Complete overview of commands

Each command is followed by one or more examples. The lines between
square brackets represent BASIC equivalents for the MSX-DOS examples.

BASIC
Quits MSX-DOS and returns you to the BASIC interpreter. From there you
can type CALL SYSTEM or _SYSTEM to go back to MSX-DOS. Note: this doesn't
work if you didn't originally boot from an MSX-DOS disk.

Examples:
A>BASIC     (starts BASIC)
[CALL SYSTEM] or [_SYSTEM]
A>BASIC start.bas (starts BASIC and runs the BASIC program start.bas)
[no equivalent]

COPY
Copies one or more files.

Examples:
A>COPY A:*.* B:   (copies all files from drive A: to drive B:)
[COPY "A:*.*" TO "B:"]
A>COPY MSX.TXT PRN  (copies the file msg.txt to the printer)
[COPY "MSX.TXT" TO "PRN"]

DATE
Shows the date. Optionally you can also change it.

Examples:
A>DATE      (shows the current date)
[GETDATE A$ : PRINT A$] (requires MSX2 BASIC)

DEL
Removes a file from disk.

Examples:
A>DEL *.*   (removes all files)
[KILL "*.*"]

DIR
Shows the contents of a disk.

Examples:
A>DIR *.BAS   (shows all files with extension .BAS)
[FILES "*.BAS"]
A>DIR /P    (shows all files, pausing after every page of output)
[no equivalent]
```

```
A>DIR /W    (shows all files in shortened format)
[FILES]
```

ERASE
Same as DEL.

FORMAT
Allows you to format a disk. Not supported by fMSX Amiga, you'll have to format a disk from the workbench instead. Don't forget to write an MSX bootblock to it.

MODE
Change the width of the screen.

Examples:

```
A>MODE 40   (turns on 40-column mode)
[WIDTH 40]
```

PAUSE
Puts the text Strike any key when ready... on the screen and waits until any key has been struck. It can also print an extra message.

Examples:

```
A>PAUSE     (waits until a key has been struck)
[no equivalent]
A>PAUSE Message   (prints Message on screen and waits)
[no equivalent]
```

REM
Does nothing at all. If any text follows the REM statement it is printed on screen.

Examples:

```
A>REM Message   (prints Message)
[no equivalent]
```

REN
Renames one or more files.

Examples:

```
A>REN *.LDR *.BAS (changes all .LDR extensions to .BAS extensions)
[NAME "*.LDR" AS "*.BAS"]
```

RENAME
Same as REN.

TIME
Shows the time. Optionally you can also change it.

Examples:

```
A>TIME      (shows the current time)
[GETTIME A$ : PRINT A$] (requires MSX2 BASIC)
```

TYPE
Prints the contents of a file on the screen.

Examples:

```
A>TYPE MSG.TXT    (shows the contents of MSG.TXT)
[COPY "MSG.TXT" TO "CON"]
```

VERIFY
Turns on verification mode. This command is actually ignored by MSX.

Examples:

```
A>VERIFY ON   (turns verification on)
[no equivalent]
A>VERIFY OFF    (turns verification off)
[no equivalent]
```

1.20 "

Using the drives from BASIC

Unlike MSX-DOS, BASIC enforces no filenaming conventions for executables or other kinds of files. However, some guidelines exist:

Files ending in .BAS are BASIC files. There are several ways to load them:

```
LOAD "FILENAME.BAS"   (loads the file)
LOAD "FILENAME.BAS",R   (loads the file and executes it)
RUN "FILENAME.BAS"    (loads the file and executes it)
```

If you want to save a BASIC file you can use this:

```
SAVE "FILENAME.BAS"
```

Files ending in .BIN are executable files. There is only one way to load them:

```
BLOAD "FILENAME.BIN",R    (loads the file and executes it)
```

Files ending in .BAT or .COM are files that can only be used from MSX-DOS .

If the file you want to load has another kind of extension, try experimenting with it. The MSX will tell you if you did something wrong.

There is one file that has special significance to MSX-BASIC. This file is called AUTOEXEC.BAS and is executed automatically during the boot sequence.

The section about MSX-DOS contains BASIC equivalents for some MSX-DOS statements.

1.21 "

Quitting the emulator

There are three ways to quit the emulator: select Quit from the menu, close the control window, or send an ARexx quit command to the ARexx port.

If the option 'Save preferences' is activated, quitting the emulator also causes it to write many current settings to disk. These values are automatically reloaded the next time you run the program.

1.22 configure

Configuring your MSX

A real MSX2 has a small amount of battery-backed RAM, which is used to store (among other things) information about your preferred working environment.

The following commands are used to customize the environment:

```
  SCREEN n
```

Sets the desired screenmode. n can be 0 or 1.

```
  WIDTH n
```

Selects the amount of columns of text displayed on screen. Valid values for n are 80 or less for screen 0, and 32 or less for screen 1.

```
  COLOR fore,back,border
```

Selects foreground, background, and border colors. Screen 0 does not support border colors - one of the many weird features of the MSX2 VDP. Colors range from 0 to 15.

```
  KEY [ON|OFF]
```

Turns the function key list underneath the screen on or off.

```
  SET PROMPT "prompt"
```

Changes the prompt to "prompt". The maximum length of the prompt is six characters. Tip: if you often use the "Files" command (like me), it may be useful to set it as the prompt.

```
  SET SCREEN
```

Stores the current settings in battery backed RAM. fMSX actually keeps a copy of these values on disk, in the file fmsx.prefs.

1.23 "

Using the Amiga hardware

Keyboard

Keys are directly mapped to their MSX equivalents, which means that the MSX ROMs determine what character results from a key press. The ROMs that come with this version of fMSX Amiga give you an American keymap.

Some keys that may not be obvious:

Amiga key	MSX key
F6	select
F7	stop
F8	clear/home
F9	insert
F10/delete	delete
left alt	graphics
right alt	code

The following Amiga keys do nothing at all:

- Both Amiga keys.
- The keys on the numeric keypad (except '(' and '*').
- The extra international keys found on some keyboards.

Joystick

The emulator can use two Amiga joysticks. The second joystick is activated by pressing '(' on the numeric keypad while in the emulation screen. Press '(' again to get the mouse back.

Joysticks with two buttons are very common with MSX systems, and many games take advantage of the second button. Often there is a keyboard equivalent for the second button; Konami games use 'M' and 'N' for this.

Video

The emulator should run fine on any Amiga screen, including PAL, NTSC, and any graphics card that supports workbench emulation. This has been confirmed to work with several types of graphic card.

The high-speed mode will certainly not work with a graphics card, and may in fact require a 15KHz mode to be active. Without a 30Khz monitor it's hard to be sure :-(.

Audio

Depending on which soundmode is selected, fMSX will either use the Amiga sound hardware directly or use AHI. AHI is a system for playing up to 128 channels of sound, and is available on Aminet.

Clock

MSX2 machines have a built-in battery-backed clock. The emulator substitutes the Amiga clock for this clock.

Diskdrives

MSX disk drives are emulated as Amiga devices. Read the section about

diskdrives
for more information.

1.24 "

High-speed mode

fMSX Amiga is written with an eye towards system-friendliness. This has the unfortunate drawback that it is at times impossible to tap the full potential of the Amiga. In order to show what is *really* possible with the Amiga custom chips, a new way to refresh the screen was added to fMSX Amiga: high-speed mode.

In high-speed mode the copper controls the blitter, which refreshes the screen every second frame. As far as I am aware there is no way to do this in a fully system-supported way, which means that some typical Amiga features are disabled as long as high-speed mode is active.

Disabled features

The following features are disabled during high-speed mode:

- Screen dragging.
- Screen switching.

In addition, the blitter is locked up all the time, which may cause other tasks to be put on hold as long as high-speed mode is active.

Enabled features

All other Amiga OS functions remain enabled during high-speed mode, including multitasking.

High-speed mode is activated with the HELP key. Pressing HELP a second time returns the program to its normal mode of operation.

Only screenmodes 1 and 2 support high-speed mode.

1.25 "

Bugs and restrictions

Z80 emulation
- All instructions except the illegal ones from the XDCB group are emulated.

Video emulation
- Screens 4 is not yet emulated.
- screens 6 and 8 don’t have any sprites yet.
- Line interrupts don’t work except in screen 5.

If you find any bugs, please report them to the
authors
.

1.26 "

Future plans

Shortish term
- Remove everything mentioned in the
Bugs
section.
- Expand available MSX2 features, both speed and features.
- Turn the graphics system into a library, allowing specific libraries for other display systems (such as CyberGfx).
- More speed.

Longer term
- Support the FM-PAC.
- Support MSX2+ and MSX Turbo-R

1.27 "

Where to find MSX software

MSX software is available through FTP from the following sites:

Site:	Directory:
altair.komkon.com	/pub/MSX/
ftp.saitama-u.ac.jp	/pub/msx/
ftp.funet.fi	/pub/msx/
riaph.irkutsk.su	/pub/

MSX software is also available via WWW:

Site:
http://grelb.src.gla.ac.uk:8000/~webster/msx
http://www.cs.umd.edu/users/fms/

1.28 "

Why are the icons so ugly?

There are two possible reasons.

The first is that your Workbench palette is different from mine. If
you want to see what the icons should look like, set your Workbench
to 8 colors, and use the following palette:

color #0-3: standard Commodore colors
color #4: full red
color #5: - (unused)
color #6: full blue
color #7: - (unused)

The second reason is that I am a crap artist. Feel free to draw nicer
icons, if you like. And *SEND ME A COPY*!

A new set of icons was donated by Johan Forsberg (d92-jfo@nada.kth.se).
These icons use the Magic Workbench colors and are much better than my own
poor attempt at art.

1.29 "

How do I load... ?

Diskbased software

An MSX filename consists of eight characters, followed by a dot and another
three characters. The last three characters are called the extension.
The extension generally tells you how to load a file. So, if the filename
is...

xxx.BAT or xxx.COM
Go to
MSX-DOS
and type xxx.

xxx.BAS
Try the following: in BASIC, type RUN "xxx.BAS". If it responds with an
error message it isn't a BASIC file, try loading it as an executable file.

xxx.BIN
Try the following: in BASIC, type BLOAD "xxx.BIN",R. If it responds with an
error message it isn't an executable file, try loading it as a BASIC file.

xxx.PMA
This is an MSX archive file, similar to a .LHA file on the Amiga. You
need PMEXT.COM (available from several
FTP sites
) to unpack it.
PMEXT.COM requires
MSX-DOS
.

xxx.DSK
This is a disk image. See below for more information.

any other
Try loading it as either a BASIC file or an executable file.

If none of these work, it is a file that is part of another program, in which case it can only be loaded by that program.

Tapebased software

Tapebased software consists of a group files and an index file (see the description here .

There are three ways to load files from tape:

```
  BLOAD "CAS:",R
```

This command loads the first binary file it can find from tape and executes it.

```
  LOAD "CAS:",R
  CLOAD
```

These commands load the first BASIC file they can find from tape. The MSX has two ways to store BASIC files on tape - and they are not compatible. You must use the right command for each type of file.

After loading a file with CLOAD you must type RUN to activate it.

ROM files and disk images must be loaded from the Amiga side

Only MSX files can be loaded using MSX commands. ROM files and disk images are not really MSX files (a real MSX cannot load them) and can only be loaded from the Amiga side.

To load and activate ROM files you must use the "Load" and "Reset" buttons on the control window .

Disk images can be loaded from the system preferences window . Just type the name of the disk image in one of the "Device A:" or "Device B:" gadgets.

1.30 "

When MSX software fails to run...

Note: many tricks described here require a reboot before they take effect.

There is an ARexx script in the ARexx drawer that sets fMSX to its most compatible settings, as advise here. The script is called MostCompatible.fmsx.

Try a higher interrupt period

Some games perform lots of processing between every two interrupts, and the emulated Z80 may not be able to keep up (this is especially true for graphically intensive operations). If interrupts happen too early this may cause the MSX to crash or behave erratically.

Setting the interrupt period to a higher value causes the emulated Z80 to perform at least a certain number of instructions between two interrupts. This means that the game will run slower, but there is also a better chance of it working at all.

A good example of a game that needs a higher interrupt period is Maze of Galious.

The amount of instructions executed cannot easily be determined, but is normally about five times higher than the interrupt period.

Although it is possible to enter values as high as 5000 it is not normally necessary to go above 1500. Note that you do not need to reboot for this setting to take effect.

```
  Use less drives
```

Every drive costs a bit of BASIC memory. Many games will not run when more than one drive is present, because the BASIC loader can run out of memory. Turning off the second drive (using the appropriate gadget in the system preference window) may help. Allocating more memory to the MSX will not help, this does not increase BASIC memory at all. Such are the wonders of the MSX memory architecture...

```
  Use more memory
```

Some games require an MSX device called a memory mapper to be installed. 98% of all programs will feel at home on a 256Kb mapper, and the rest will run fine with 512Kb. The amount of memory used by the memory mapper can be selected in the system preferences window.

```
  Use MSX1 ROMs
```

Some MSX1 games do not properly turn off MSX2 features, causing garbage to appear on the screen (shades of ECS vs. AGA here?). Disable MSX2 in the system preferences window and see if the problem disappears.

```
  Try hacking with BASIC
```

If all this doesn’t help there is still a whole bag of BASIC tricks left to try. Some of the following might do it - then again, it might not:

```
  POKE &hFD9F,201
```

This turns off disk interrupt processing. On a real MSX this means the drive engine never stops (this is regulated through this interrupt) but fMSX controls the drive engine on its own.

```
  CLEAR 0
```

This frees up extra BASIC memory.

MAXFILES = 0

This frees up extra BASIC memory.

Poking address -1 doesn't help

The mythical address -1, aka 65535, aka &hFFFF, is used by a real MSX for secondary slot selection. fMSX Amiga supports nor needs secondary slots, therefore pokes to this address are ignored. This is a real feature, ask almost any MSX2 owner ;-)

Get the next release

fMSX Amiga is still under development. I try to have better compatibility with each new release. If something doesn't work and isn't listed in the

bugs
section send me a report and I'll see what I can do.

1.31 "

How to obtain new versions

New versions of fMSX Amiga will be distributed in three ways: one copy will be uploaded to Aminet, another to the webpage of
Marat Fayzullin
and finally a copy will be mailed to everybody on the fMSX Amiga distribution list. If you want to be placed on this list (or removed from it) simply send mail to

h.guijt@inter.nl.net

stating what you want. If you have mailed me in the past with questions about fMSX you have automatically been placed on the list; I apologize if you did not intend to receive new versions of fMSX. If you asked to be on the list but aren't, try mailing me again. I am not a mail-demon, and I occasionally misplace mail messages.

You do not need to fear being swamped by mail; there is usually a few months between each successive version of the emulator.

1.32 "

About comp.sys.msx

For public questions about MSX or fMSX, your best bet would be the newsgroup comp.sys.msx.

I'm no longer on comp.emulations.misc. There is just too much noise in that group.

Of course, questions can be aimed at the

```
                authors
                 as well.
```

1.33 "

```
    About the authors

fMSX Amiga is being written by Hans Guijt. Until march 30 1995 I was a
computer science student at the university of Leiden, but since june 1st I
am employed by Palm Automatisering in Aalsmeer. I am not aversed against
bug reports, mail (of any kind), gifts, etc. Actually I quite like receiving
email, so write to me!

Past and future projects include: a large UNIX based system for image
normalization and recognition (Photobase), an advanced MSX disassembler
(MSXResourcer), needed to convert some games from MSX to Amiga but useful
for fMSX as well, and a level editor for the first game that is to be
converted (Konami's Metal Gear). Unfortunately the game conversion is not
getting much time lately, much to the frustration of the other people who
work on that project.

My (snailmail) address:

  Hans Guijt
  Kagersingel 30
  2172 XG Sassenheim
  The Netherlands

  Telephone: (Holland) 2522-17251
  Email: h.guijt@inter.nl.net


fMSX (generic) is being written by Marat Fayzullin. He is a computer science
student in University of Maryland in College Park, who is going to graduate
this May [any job offers, especially from Europe? :)]. Aside from fMSX, he
also wrote WBGames and Dashboard for Amiga, maintains two FAQ lists on
USENET, and a large WWW site at

  http://www.freeflight.com/fms/

He also doesn't refuse small [and not so small] monetary donations
in hard currency [preferably US dollars] sent to:

  Marat Fayzullin
  6304 Hampton Place
  Elkridge, MD 21227
  USA

  phone: (410)-379-2925
  email: fms@freeflight.com
  IRC:   RST38h
```

1.34 "

About RAMSX:

Included in this package is a mountlist for a device called RAMSX:. This device (built from ramdrive.device and crossdosfilesystem) can be used with fMSX to play disk-based games directly from RAM (it’s not as good as using a harddrive, but it’s much better than using disks!).

If you want to use RAMSX: you will need about 720Kb of free RAM and the crossdosfilesystem. You must also enter the device name RAMSX: into the appropriate drive gadget in the preferences window.

It can be rather hard to recover RAMSX after a reboot, but it can be done. Open a shell and type the following commands:

```
  assign RAMSX: dismount
  mount RAMSX:
```

and RAMSX: is back online. Thanks to Nikos Drogosis (ndrog@acropolis.net) for reporting this.

1.35 "

About FMSX0: and FMSX1:

Included in this package are mountlists for two devices called FMSX0: and FMSX1:. These devices require fmsdisk.device to be installed. fmsdisk.device is available from Aminet (directory misc/disk).

FMSX0: and FMSX1: allow you to store MSX disks on the Amiga harddisk. If you wish to easily access floppy-based MSX disks but cannot be bothered with real floppies these devices are an excellent alternative.

FMSX0: and FMSX1: were contributed by Andrew G. Robson (a.g.robson@northumbria.ac.uk). Complaints and compliments are rightly his.

1.36 "

A big thank-you to:

Peter McGavin (peterm@maths.grace.cri.nz)
Peter (who wrote an excellent Spectrum emulation) contributed many ideas for the Z80 emulation. It was his Spectrum 1.7 which convinced me that fast Z80 emulation is possible. In addition the c2p necessary for the MSX2 screenmodes were inspired by his Flick 1.5.

Jeroen Vermeulen (jtv@xs4all.nl)
Jeroen kindly offered to proof-read the accompanying documentation, and found heaps of spelling errors. He is also a willing betatester (victim?) who has found many enforcer hits and general nastiness.

Alex Wulms (a.wulms@inter.nl.net)
Alex owns an MSX Turbo-R, and is the author of Zone Terra, an excellent shoot'em-up for that system. He is also an MSX-wizard, who explained many of the MSX's internal workings to me.

Sergi Martinez and Juan Gomez (pixador@adam.es)
Juan and Sergi are the authors of AmiMSX , a program similar to fMSX. Their input so far includes a trick to speed up fMSX by 30%, information about the MSX mouse interface (to be added in a future version), and of course the constant inspiration provided by new AmiMSX features ;-)

Martin Blom (lcs@lysator.liu.se)
Martin is the author of AHI, a system for playing up to 128 channels of sound at the same time on a fast Amiga. AHI made the shared PSG/SCC mode possible, and is so much better than my own attempts at channel mixing that I can safely say it would never have happened without it.

Kenny (kenny@bgnett.no)
The otherwise unnamed Kenny posted a short ARexx tutorial on comp.sys.amiga.programmer. Thanks to his writings fMSX now has an ARexx port.

Børge Nøst (borgen@icenet.no)
For finding, fixing and reporting a major bug in high-speed mode.

Andrew G. Robson (a.g.robson@northumbria.ac.uk)
For contributing FMSX0: and FMSX1: .

Nikos Drogosis (ndrog@acropolis.net)
For explaining how RAMSX: can be remounted after a reboot.

Johan Forsberg (d92-jfo@nada.kth.se)
For donating the Magic Workbench iconset for MSX.

Mark Knibbs (markk@netcomuk.co.uk)
For reporting lots of bugs and misfeatures.

...and all the betatesters and supporters!

1.37 "

About AmiMSX

fMSX is not the only MSX emulation available for the Amiga. A similar program called AmiMSX is also available through Aminet. I shall attempt to make an honest, non-biased comparison (and I point you to the AmiMSX documentation for the other side of the story ;-)):

Advantages of fMSX

- fMSX is completely free, whereas AmiMSX needs to be registered.
- fMSX can use cartridges.
- fMSX is completely system-friendly.
- fMSX can run with the 68040 copyback cache enabled (I have heard reports that AmiMSX needs to have copyback disabled).
- fMSX has better SCC support.

Advantages of AmiMSX

- AmiMSX is quite a bit faster.
- AmiMSX kills the OS for extra speed (it returns to the workbench when paused). I do not personally consider this an advantage but I know some people do.

There are also features common to both programs:

- Both require 68020+ and OS2.0+.

If you like MSX emulation you will probably want to check out AmiMSX as well.