fmsx

COLLABORATORS

	TITLE : fmsx	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

fmsx

1.1 "

fMSX Amiga 0.7

General information

Installation and usage

Technical 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.

I've heard rumors that the Colecovision is in fact an MSX console with a slightly different way of interfacing the cartridges. If this is true it should be possible to use Colecovision software with fMSX! If you know more about this, don't hesitate to write to
me
.

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 "

Notes from the author

Hi everybody!

It’s been a while, but here it is: a new fMSX Amiga update.

Let’s get straight down to the most important thing: new features.

The reason this version took so long is that I have completely rewritten the Z80 emulation. Someone asked why I did this - well that’s because I am a natural masochist... No that’s completely wrong, actually it was because I wanted faster megaROMs.

I am happy to announce that this has in fact been accomplished.

With the new Z80 emulation...

+ MegaROMs are much faster,
+ as is booting the emulation from disk, or with MSX2 ROMs.
+ It uses approximately 100KB less memory,
+ and it can use a memory mapper,
- however, everything else is just a little bit slower,
- and I have not been able to track down the bug that causes graphic corruption in almost every Konami game.

Another new thing is support for the SCC sound system. To activate this you must do three things:

- Set the ROMType gadget to ’SCC’
- Set the soundmode gadget to ’SCC’
- Load a game that supports SCC sound (Salamander, Nemesis 2 & 3, King’s Valley 2)

General guidelines for recognizing SCC games: they are exclusively produced by Konami, they are 128KB or bigger, and they sound like sh*t using normal PSG emulation.

The other Big New Thing is in the experimental stage, but may still be fun to look at: screens 5 and 6 are partially emulated. Turn on MSX2 in the preference window and you can look at the intro sequences of Space Manbow and Metal Gear 2 (disk version only for now), or watch the attract mode of Firebird.

Actually playing games in these modes is not possible, because I haven’t implemented sprites yet. Also, the MSX2 blitter is only partially done. It doesn’t do transparency yet, or logic modes, and some commands are missing as well.

Despite all this I still wanted to release this version as it is, especially because this is the first version that is compatible with the v36 OS.

My thanks go to those people on the net who assisted in debugging the v36 compatibility - it took a while to get right, but now it works!

Happy MSX’ing,

Hans Guijt

1.9 "

Installation

No specific installation procedure is required, just drag the directory containing fMSX to some place of your liking.

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.

1.10 "

System requirements

Required are:

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

Recommended:

- 68030 processor, at least 25MHz
- Those 600 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.

This program was developed using:

- An a4000/30 with 6Mb RAM and 260Mb HD
- SAS/C 6.51
- GenAm 3.02
- TurboText 2.0

1.11 "

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, most of the MSX2 features are still missing! Currently supported are screens 5 and 6 (partially), vertical scroll, some blitter commands. Not supported is the rest: line interrupts, screen positioning, screens 7 and 8, MSX2 sprites.

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

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

The following sort of software will not run:

- MSX tape

Actually, if someone else does a good MSX tape -> Amiga disk conversion program I’m willing to give it a try. Write to me for further discussion if you think this is a nice project!

1.12 "

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 sets the number of times the emulation causes an interrupt before the screen is redrawn. If the refresh cycle if very low the emulation runs smooth and slow. If the refresh cycle is high it runs faster but also less smooth. Try to experiment with what works best for a particular game. Usually a value of about 3 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!

If you do not know the correct setting for a MegaROM you have no option but to try them all.

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

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.

Reset

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

Preferences

This button opens the
preferences window
.

1.13 "

The preferences window

This window allows you to specify several more settings for the emulation.
Preferences are saved when the emulation is quitted.

Device A: and B:

In these gadgets you have to specify which Amiga device 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 hardfile or ramdisk
and use it instead.

The device you specify must have certain properties, among which are:

- It must have a blocksize of 512.
- It must have a low-cyl of 0.
- It must have a high-cyl of 79.
- It must have 9 blocks per track.
- It must be a real device, no assign or volume.
- It must be mounted.

It is not necessary to fill in a value in these gadgets, but if you
don't you cannot use the corresponding MSX drive. Also note that these
settings take effect only when you reset the MSX.

An example of a RAM-based device with correct settings is provided in
the archive.

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.

Videomode

With this gadget you can set the video mode of the MSX screen. There are
four options:

- PAL: always force the screenmode into PAL. This means (among other
things) that MSX interrupts are issued 50 times per second.

- NTSC: always force the screenmode into NTSC. Although I have a PAL
machine I use NTSC mode for fMSX, because the screen is bigger and
interrupts are issued faster (causing the music to run faster as well!).

- Amiga: use the same screenmode as the workbench of your Amiga uses.
This option is meant for those people with 30KHz monitors.

- MSX: an MSX2 has PAL/NTSC selection just like the Amiga. This option causes the emulation to choose the same video mode as the MSX.

This setting takes effect when the MSX screenmode changes.

Soundmode

With this gadget you can select between PSG and SCC sound. SCC is a sound chip used in some Konami games. It provides far better sound then the PSG.

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 consists of nothing more than beeps and clicks.

Examples of games that support SCC sound are:

Salamander
Nemesis 2
Nemesis 3
King’s Valley 2
F1 Spirit (not that it works with fMSX Amiga or anything...)

And MSX2 games:

Space Manbow
Metal Gear 2
Quarth
Gryzor
King’s Valley 2 MSX2

Note that SCC emulation is incomplete; it should emulate 5 SCC channels and the entire PSG chip, rather than just 4 SCC channels. I’m working on a channel mixing procedure that will play all channels from both chips at the same time.

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 of the buggers!

MSX2 enabled

Enables / disables MSX2 features.

Allow external ROMs

If turned on fMSX will first look on disk for ROM files, otherwise it will use the internal ROMs. Note that internal ROMs are always used when no external ROMs are available.

Quick inactive

This cryptically named option does the following: when it 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!

1.14 "

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.

Write bootblock

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.

1.15 "

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. Default is 3.

```
  INTERRUPTPERIOD=value
```

Legal values are 0 to 5000. Default is 1500.

```
  ROMTYPE=value
```

Legal values are 0 to 3. Default is 0.

1.16 "

Using different ROMs

fMSX has built-in ROMs, but it is possible to use different ones. To use different MSX1 ROMs: place a file containing BIOS and BASIC in the same directory as the program itself. The file must be named MSX.ROM.

Using MSX2 ROMs works the same, but requires two files: MSX2.ROM (BIOS and BASIC), and MSX2EXT.ROM (the SUBROM).

You must also turn on external ROMs in the preferences window.

1.17 "

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
.

1.18 "

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]
```

1.19 "

```
                    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.
```

1.20 "

```
    Quitting the emulator

There are two ways to quit the emulator: select Quit from the menu, or
close the control window.

Quitting the emulator also causes it to write many current settings to
disk. These values are automatically reloaded the next time you run the
emulation.
```

1.21 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 SCREEN
```

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

1.22 "

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.
- The extra international keys found on some keyboards.

The HELP key switches between system-friendly and
high-speed mode
.

Joystick

The emulator can use one Amiga joystick.

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 works with at least the Retina 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 :-(.

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.

1.23 "

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 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.

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

Not all screenmodes support highspeed mode.

1.24 "

Bugs and restrictions

Z80 emulation
- No undocumented Z80 instructions have been implemented so far, apart from SLL.
- I suspect there is an error in the shift/rotate section. I haven't been able to find it, so far.

Video emulation
- Screens 4, 7 & 8 are not yet emulated.
- There are no sprites in MSX2 screenmodes.
- Blitter emulation is incomplete.
- Line interrupts don't work.

Sound emulation
- SCC emulation is only 4 channels rather than 5 + PSG.

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

1.25 "

Future plans

Shortish term
- Remove everything mentioned in the
Bugs
section.

- Expand available MSX2 features.

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

1.26 "

Where to find MSX software

MSX software is available through FTP from the following sites:

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

ROMs and megaROMs are stored as big archives packed with TAR and GZIP (these are UNIX utilities that are also available for the Amiga; copies can be obtained from Aminet). Diskgames should be in separate directories.

MSX software is also available via WWW:

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

1.27 "

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.

1.28 "

How do I load... ?

A 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.

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.

1.29 "

When MSX software fails to run...

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

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 examples of a game that needs a higher interrupt period is Maze of Galious.

The amount of instructions skipped cannot easily be determined, but is normally about five times as high as 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 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 preferences window.

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 ;-)

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 preference 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.

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.

1.30 "

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 weeks between each successive version of the emulator.

1.31 "

About comp.sys.msx

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

I'm also on comp.emulations.misc.

Of course, questions can be aimed at the
authors
as well.

1.32 "

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.cs.umd.edu/users/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@wam.umd.edu
IRC: RST38h

1.33 "

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.

I feel that this mountlist can be greatly improved (for example, the device must be formatted before the icon appears on the workbench, and it cannot be recovered after a reset), but I fail to see how this is done. If you can do better than this (or if you have made similar mountlists based on other devices) please contact me.

1.34 "

A big thank-you to:

Peter McGavin
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.

Jeroen Vermeulen
Jeroen kindly offered to proof-read the accompanying documentation, and found heaps of spelling errors.

Alex Wulms
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.

...and all the betatesters who endured lots of successive versions with weird numbers like v0.6b and v0.6c (not necessarily in that order), while I attempted to add v36 compatibility to fMSX.

1.35 "

History

0.7: (28-10-95)

- The MSX will now recognize if the Amiga makes a change to a disk.
- Found and corrected a memory-corruption bug.
- Resident programs will be flushed when Reset is pressed.
- Z80 emulation completely changed. MegaROMs are much faster, the rest is a bit slower.
- Memory mapper support added.
- Limited support for the SCC sound system added.
- Handling of cartridge filename is much better. It now appears in the requester and the path is remembered between sessions.
- Removed several minor errors from the drive allocation code.
- Improved support for screen 0 (faster, 80 columns).
- Removed the check that cartridge files are really cartridges. This is the software equivalent of an SCC switch.
- Rewrote critical memory management routines in assembly.
- Implemented many VDP operations.
- Partial support for screens 5 and 6 added.
- Now compatible with the v36 OS.
- Includes MSX2 ROMs.
- Several new options: freeze screen while window is inactive, use MSX2 ROMs, use internal ROMs only.

0.6: (23-08-95)

- Drive support added!
- A two-line enhancement to the VRAM-write scheduler causes a great deal of speedup in some games, for instance Tank Battalion.
- The user interface was enhanced and offers a lot more functionality now.

```
- The Z80 emulation is smaller and faster than before.
- Preferences are now automatically saved when quitting fMSX.
- No longer hangs when the right mousebutton is pressed in highspeed
  mode.
- MSX2 ROMs didn’t work in v0.5. Fixed.
- Fixed a bug in the MSX2 palette code.
- Highspeed mode is automatically turned off if the fMSX screen is
  deactivated. This means that it should no longer hang if for some
  reason a window on another screen becomes activated.
- Added requesters for most errors and warnings.
- Removed the ability to use two cartridges at once. This will return
  in a future version.

  0.5: (17-07-95)

- There was a bug in 0.4 that caused the emulation to run a lot
  slower than necessary.
- A simple change to the bankswitching code caused some speedup,
  especially notable when starting with MSX2 ROMs.
- If one of the libraries could not be opened the general shut-down
  routines would still call functions from that library.
- No longer hangs when it cannot allocate sound channels.
- Rewrote some of the documentation.
- Added high-speed video refresh mode for screens 1 and 2.
- Rewrote the PSG emulation. As a consequence, the Dunkshot music
  no longer suffers from random beeps.
- Added noise to the PSG emulation.
- Sound output is a lot softer now, which allows for far better white
  noise emulation.
- The joystick now supports two firebuttons.

  0.4: (02-04-95)

- Joined the ROMs together to one file.
- The ROMs are now internal, but external ROMs are still possible.
- It no longer closes and reopens the screen if the new screen has
  the same size and depth as the old one.
- The GUI has been made font sensitive.
- Started work on the hardware hitting screen drivers.
- Changed DoOut, DoIn, and VDPOut to conform to fMSX UNIX 0.7.
- Added VDP operations directly from fMSX UNIX 0.7.
- Will load MSX2 ROMs if available. MSX2 BASIC runs, but starting
  takes a long time.
- Screen 6 implemented (crudely).
- Fixed the problem with the joystick mechanism. Protector and
  Volguard work fine now.

  0.3: (08-03-95)

- Some important routines were recoded in assembler, giving a nice
  speed increase.
- Turned off the low-pass filter for more authentic sound.
- Transparent sprites are made invisible.
- Task priority is lowered to -1.
- Uses a greater volume range (0-63 instead of 0-60).
- Keyboard handling is a bit smoother on slow machines.
- Keyboard mapping has changed.
```

- Removed the bug that caused the humming sound in Eggerland Mystery, Dunkshot, and other Hall Laboratories games.
- Rewrote the video drivers for greater speed.
- Sprites are now clipped against the top border.
- Interrupts are limited to at most 50 per second.

0.2: (20-02-95)

- Now supports loading two cartridges (for use with the Gamemaster).
- A GUI was added. It’s topaz-8 sensitive for now.
- The help key is no longer used to quit the emulation, so insert works now.
- The interrupt mechanism was somewhat enhanced.
- It is now possible to set the number of interrupts per screenredraw.
- Added the ability to set the number of instructions per interrupt.
- Added joystick support for one joystick with one button.
- Refresh cycle and interrupt period are read from the cartridge icon.
- Added a ’save configuration’ option.
- The emulator now detects if no sound channels could be allocated and runs without sound if so.

0.1: (13-02-95)

- Screens 0..2 should work with ECS and graphics cards now.
- Sprites are clipped horizontally.
- Sprites support shift-left bit.
- Screens are no longer specified as being PAL.
- Screen 3 works fine now.
- Sprites no longer flicker.
- Sound is much better now, though some bugs remain (specifically: no noise generation, and no effects on channels 1 and 2).
- Vertical sprite position was off by 1 pixel.
- Removed a glaring bug that prevented 32K cartridges from loading. Software compatibility jumped to 85%.
- Added the ability to load a cartridge when starting from the workbench.
- Video emulation now respects the Disable Screen bit.
- Sometimes sprite number 0 wasn’t displayed.
- Enlarged sprites were broken.

0.0: (09-02-95)

- First release. This is only for internal review, but people who really want it can get it as well.