

DREaM

COLLABORATORS

	<i>TITLE :</i> DREaM		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 25, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DREaM	1
1.1	Contents	1
1.2	What is DREaM?	1
1.3	System Requirements	2
1.4	What's New?	3
1.5	Features of the emulator	4
1.6	Virtual Cassettes	5
1.7	The control windows	6
1.8	Joystick / Keyboard support	10
1.9	Sound support	11
1.10	Parallel port support	11
1.11	Emulation speed	12
1.12	Example programs	13
1.13	Conversion utilities	13
1.14	Restrictions on Distribution and Use	15
1.15	Acknowledgements	15
1.16	Contacting the Author	15
1.17	Problems	16
1.18	Development History	16
1.19	Possible Future Work	17

Chapter 1

DREaM

1.1 Contents

DREaM V1.02

Written By Sean Siford, (C) 1995-1997.

What is DREaM?

System Requirements

What's new?

Features of the emulator

Current problems

Speed of emulation

Example programs

Conversion Utilities

Notes on distribution and usage

Future Work

Acknowledgements

Contacting the Author

History

1.2 What is DREaM?

What is DREaM?

DREaM is a Dragon 32/64 Tandy CoCo emulator for the Amiga.

The emulator supports most of the hardware that the machines above do, with the most notable exceptions being a few screen modes and audio cassette loading and saving.

1.3 System Requirements

What does the emulator require?

The emulator requires a minimum of a 68020 microprocessor (sorry 68000 and 68010 owners), and V1.3 or better of the operating system.

A minimum of 450K of free ram (preferably fast ram, though about 75K needs to be chip) should be available on running the emulator.

The emulator has been tested on versions 1.3 and 2.04 and 3.0, but may well work under 1.2 also. Not having access to a machine running V3.1 of the operating system, I am unable to say whether the emulator runs under it, but there should be no reason why it should not.

The emulator has very few additional requirements:

The emulator uses Reqtools.library, which needs to be present in the Libs: directory of your boot device in order for it to run. The versions for V1.3 and V2.0 and above can be found in the Reqtools directory found in the archive.

Also, a copy of the ROMS of the machines that you wish to emulate need to be accessible to the emulator on startup.

The emulator expects the images for each ROM to be named as follows:

Dragon 32 - dragon32.rom
Tandy Coco - Tandy.ROM
Dragon 64 - dragon64_1.ROM (32k compatible)
 dragon64_2.ROM (64K Basic ROM)

The emulator expects these images to be found on the volume DRAGON:. The ROM image is not supplied with the emulator, for legal reasons - if you find it with the archive, someone else has included it with the archive - not me..

So, you will need to create an assign called DRAGON: where the ROM files can be found:

E.g. If the ROM file could be found in Work:Emulators/DREaM/

assign DRAGON: Work:Emulators/DREaM/

A ROM image can be found in a archive - 6809Roms.Zip, which is available at several FTP sites, and also the Dragon WWW home page. The files within this archive are in the .DGN format, which is used by Paul Burgin's PC-Dragon & T3 emulators for the IBM PC.

The image cannot be used by the emulator directly, but can be converted into suitable form using the ConvROM utility provided in the archive.

For details on the ConvROM utility and others supplied with the emulator, see

Conversion Utilities

Please note that not all of the ROM files are required to use the emulator, and it will allow the use of the various available architectures depending on the ROM files available.

1.4 What's New?

What's new?

Those of you who used the previous version of the emulator (V1.0B) will note that there are two versions of the emulator as last time. However, the two versions in this release do not differ in the way that the display is emulated, but instead the 6809 emulation differs slightly.

(For those of you reading this who didn't use the last version, one version of the emulator used the copper for the display emulation. This has been dropped, and now no special copperlists are used to emulate the display.)

The two versions of the emulator consist of one version where the speed can be adjusted, and one where it cannot. (These are called DREaM_Cyc and DREAM_NoCyc respectively).

The reason for producing two versions is that the speed of emulation is controlled by counting the number of cycles elapsed within a 50th of a second. This introduces some overhead which slows the emulation down. Therefore, users with less powerful machines may find that the version where the speed cannot be altered runs faster.

The emulator can now also be configured, and this configuration can now be saved. The configuration is saved as Dragon_Config in the DRAGON: volume, and is loaded when the emulator is loaded again.

The various items to be configured can be viewed using the 'Set Config' option on the main window on the workbench screen.

The type of joystick emulation, keyboard emulation (CoCo / Dragon) and update rates for each of the main modes that the emulator supports can now be controlled.

The emulator now supports an artifacing-like mode.

This attempts to imitate the red or blue fringing which is seen on televisions, and are used by some programmers to make 2-colour games look much better.

There are also a few new keys available while the emulation is running. The F4 and F5 keys can be used to flip between Dragon and CoCo keyboards, so as you don't need to go to the configuration part of the emulator to change the keyboard.

The emulator also can emulate the effect of artifacing in Mono black & white modes, which can improve the look of games written to take advantage of the effect (E.g. Mr Dig & Zaxxon). This can be selected by using the F6 (blue), F7 (red) and F8 (normal) keys.

Also, now you can flip-back to the emulator window on the workbench and pause the emulator at the same time by pressing the F10 Key.

Also, you can now load snapshots compressed with XPK. However, the emulator doesn't currently support saving compressed snapshots.

Also, the emulator supports cartridge files now, which can be loaded in the same way as snapshots. These are started by performing a hard reset of the emulator. To purge a cartridge from memory, use the 'Wipe Cartridge' option on the main control window.

Oh yes, also you can now switch between Dragon 32, 64 & Tandy Coco emulation!

1.5 Features of the emulator

Features

As previously mentioned, the emulator supports the vast majority of the Dragon's hardware including display, 6809, Peripheral Interface Adaptors (including Sound and Parallel port emulation).

The emulator also allows the loading and saving of cassette data through use of 'virtual cassettes'. Virtual cassettes are files that hold the binary content of Dragon audio cassettes, and are used by the T3, PC-Dragon and CoCo Dragon / Tandy CoCo emulators available for the PC.

DREaM can load and save these files without any conversion.

Currently there is no program in the Amiga that converts Dragon audio cassettes into the virtual cassette format, although PC-Dragon and CoCo both contain programs to perform this task. This is something I hope to include with the emulator at a later date.

The emulator also supports snapshots, which allows programs to be saved during execution of the emulator, and reloaded at a later point in time.

The snapshot format used by the emulator is not compatible with the .PAK snapshots saved by CoCo and PC-Dragon, or the .D32, .D64, .D96 snapshots saved by PC-Dragon. There are two utilities supplied with the emulator

which convert files between the .PAK/.D?? formats saved by PC-Dragon and those used by this emulator, and vica versa. See the links below for more information.

Please note that the emulator now supports loading snapshots packed with XPK, if the relevant compression library is available. However, the emulator will not saved compressed snapshots. I would ask that you do not upload compressed snapshots to anywhere, as it may be that the downloader does not have XPK, and so cannot use the snapshot.

As you may already be aware, there are actually two versions of the emulator supplied. The emulators are the same in all but two respects.

The first difference is that one version (DREaM_Cyc) allows the speed of emulation to be controlled, whereas the other (DREaM_NoCyc) does not allow this facility. You may think that it is strange that this has been done.

The reason for the two versions is that the speed is controlled by counting the number of cycles elapsed in a given time, and if this total is exceeded then the emulation is paused for a short while. However, every instruction executed needs to have the number of cycles it 'consumes' taken into account, which causes execution of the emulator to be slowed.

For those with faster machines, the cycle counting makes very little difference, but for those who have slower machines the difference can mean that the speed of execution is slower by a fair amount. For more information on this facility, see the link on

[The control windows](#)

.

The second difference concerns the generation of sound. For more information, on this see the link on

[Sound Support](#)

.

The following links contain information on:

[Virtual cassettes](#)

[The control windows](#)

[Joystick/Keyboard Support](#)

[Sound support](#)

[Parallel port support](#)

[Conversion Utilities](#)

1.6 Virtual Cassettes

The virtual cassettes

As previously mentioned, the emulator supports saving and loading of virtual cassette files, using the normal CLOAD, CLOADM, CSAVE and CSAVEM commands in basic.

When the user attempts to load a cassette when no opened cassette exists, the user will be asked to specify a cassette file to use using a requester.

This cassette file will be used for all input until the end is reached, the user resets the input cassette file, or a hard reset is issued.

The save cassette file works in a similar way. When the user first attempts to save a file (using CSAVE, CSAVEM) the user is asked for the name of a file that output is to be sent to.

All cassette output will be directed to this file until the output cassette is reset, or a hard reset occurs.

Any open cassette files are closed on a hard reset.

1.7 The control windows

The emulator display and control windows

The emulator is split into three parts, the control window, the configuration window and the main emulator display screen.

The control window (resides on the Workbench screen) is used to (amongst other things) start, stop reset the emulator.

The configuration window (obtained using the 'Set Config' button on the control window) allows various parts of the emulator to be configured.

(Please note that the emulator is paused whilst the window is displayed, and that the control window is also inactive until the configuration window is closed).

The emulator display screen is used when 'communicating' with the Dragon. When the emulator is started, the display flips to the emulator display.

The pointer should then change to a Dragon to indicate that all keypresses are re-directed to the emulated Dragon. In order to select an operation from the control window, flip back to the workbench screen and select it, or press the F10 key.

Note, it is important to remember to click on the emulator screen again in order to 'talk' to the Dragon if you manually flip to the workbench!

The Buttons on the Control Window

Load Snapshot , Save Snapshot

Allows snapshots and cartridge files to be saved and loaded.

Reset Inp Cassette , Reset Output Cassette

These buttons are used to reset the input and output virtual cassettes. These are used to allow new virtual cassettes to be selected for reading or writing.

Hard Reset , Soft Reset

Performs either a hard (power off) or a soft (reset button) reset on the emulator. Soft resets can be used to get back to BASIC without destroying the contents of the Dragon's RAM. However, some programs bypass this to the only way to get back to BASIC is to perform a hard reset.

Set ParRecord / Stop ParRecord

This button is used to start and stop the re-direction of parallel port output to a file.

Sound On / Off

This button is used to switch on or off sound output for the emulator.

Set Config

This button is used to display the configuration window which is used to configure various parts of the emulation.

Save Config

This button is used to save the current configuration to disk, so as it will be reloaded when the emulator is next started.

Pause / Run

This button is used to pause or allow the emulation to run.

Wipe Cartridge

This button is used to remove a currently loaded cartridge from memory,

and also resets the emulated machine at the same time.

Configuration Window

The configuration window allows the user to configure certain parts of the emulation's operation. The window is divided up into the following sections:

Machine:

This section allows you to select the architecture that you wish the emulator to run in. The actual architectures that can be selected depends on the ROM files available and also available RAM (D64 emulation requires more memory than D32 / Tandy Coco).

Artifacing:

This section allows the user to select whether atrifacing is emulated. Artifacing is a fringing effect which is generated when viewing monochrome modes (white on black) on televisions. These effects were used by some games programmers to improve the look of Dragon/CoCo games.

The effect is the same as PC-Dragon and CoCo emulators on the PC perform it (I like the effect anyway). This is only available in monochrome modes with a black and white palette. If you're not convinced that this looks good, look at Buzzard Bait, Mr Dig or Zaxxon!

Blue, Red or No artifacing can be selected. Also in this group the intensity of the artifacing can be selected.

The following keys can be used during the emulation to select the artifacing mode:

Key	Artifacing Mode
-----	-----------------

F6	Blue
F7	Red
F8	Normal

Display Config:

The display configuration section allows the user to configure the frame update rate and text block update rate (for text modes) for the major modes supported by the emulator.

The button at the top of the section is a cycle button, which allows the mode currently being configured to be selected. (Apologies that it doesn't look like a cycle button, hopefully the next version will fix this!).

The 'frames before display update' slider allows the number of frames before the screen is updated to be altered. A 'Frame' is a 50th of a second.

The 'text blocks per update' slider is used to define the maximum number of blocks that can be updated per screen update in text modes.

Text modes only update the display when a change occurs, so the number of text blocks to update will not have a significant impact on performance if little changes on the display.

However, if large sections of the display change in text modes (especially high-resolution ones), then this can severely reduce the performance of the emulator if many 'blocks' are updated at a time.

The values are set (by default) so as the whole screen in each mode can be updated at once. However, the user may wish to change this to reduce the overhead of screen updates.

A 'block' (as the emulator counts them) is equal to the number of bytes displayed in the mode, divided by 2. Therefore, in 32 x 16 text mode there are 256 such blocks.

Miscellaneous Options:

This section contains two sliders.

The topmost slider is used to set the number of 50ths of a second which elapse before an IRQ is generated by the emulated Dragon's hardware. On the Dragon, IRQ interrupts occur every 50th of a second.

The lower slider is used to control the speed of the emulation.

(NOTE: This is disabled on the non-cycle counting version of the emulator.)

The slider allows the maximum speed of the emulation to be set from %50 of the speed of a normal Dragon to %500, in increments of %50.

The slider actually shows the speed in terms of cycles per 50th of a second. This ranges from 9,000 up to 90,000. This is based on a 0.9 Mhz 6809 performing 18,000 cycles in a 50th of a second.

Please note that any time that the emulator is not using for the emulation is made available to other tasks - no busy waiting occurs.

Joystick / Keyboard Sections:

The joystick and keyboard sections allow the type of joystick and keyboard emulation to be altered.

The joystick option allows the emulation of digital, analogue or self-centering analogue joysticks. The keyboard section allows either the normal Dragon keyboard to be emulated, or for the keyboard emulation to emulate a Tandy Coco keyboard at the hardware level.

See

Joystick / Keyboard Support
for more information.

1.8 Joystick / Keyboard support

Keyboard / Joystick Support

The emulator allows the emulation of the Coco keyboard at hardware level in addition to the normal Dragon keyboard. This can be selected in the configuration window, but can also be selected whilst viewing the emulator's screen.

The following keys are used to select the keyboard mode:

Key	Emulation
F4	Hardware emulates the Dragon keyboard.
F5	Hardware emulates the Coco keyboard.

Both of the right and left joysticks on the Dragon are supported by the emulator, and the user also has the option to change the joystick emulation used by the emulator.

The Dragon's right joystick is used by plugging a digital joystick into the left hand port (?!) (the one not used for the mouse).

The left hand joystick is emulated using the cursor keys on the keypad, with the Help key being used as the fire button for the left joystick.

The joystick emulation methods are as follows:

Key	Emulation
F1	Digital joystick emulation, the default
F2	A non-centring analogue joystick emulation
F3	A self-centring analogue joystick

Digital Joystick Whatever direction(s) you press, the emulation assumes that the joystick is being pushed in those directions fully.

Analogue Joystick An emulated analogue joystick, the more you hold down a direction, the more the emulated joystick is moved toward the direction pressed. This occurs until the user stops pushing the joystick in that direction.

Self-Centring Analogue This is similar to the analogue emulation, but when the user stops pressing the joystick or key for a direction, the joystick automatically moves toward the centre.

The emulator does not support use of analogue joysticks in the left joystick port on the Amiga, because the gameport.device does not

support them.

Please note: If the emulator is not able to allocate the left joystick port for it's own use on startup, then both left and right joysticks are emulated using the keypad keys.

A short note about the A1200:

As you may already be aware, the A1200's keyboard is not able to determine when more than one key is pressed at a time with certain key combinations.

Unfortunately, this means that the joystick emulation for the left joystick (which uses the keyboard) is affected by this limitation. This stops a horizontal direction and a vertical direction being selected at the same time. Obviously, this isn't good, and I may solve this by moving the Dragon's cursor keys to the keypad and using the Amiga's cursor keys for the joystick emulation. However, at present this is something that will have to be lived with.

1.9 Sound support

Sound Support

The Dragon has a very basic sound output, in the form of a 6-bit DAC and a single bit sound output. Both are supported by the emulator.

The sound on the emulator is disabled by default. The Audio.device is used to allocate a channel for the sound when it is selected by the user, and the user will be informed of any problems due to lack of channels.

When sound output is switched off, no sound channels are used by the emulator.

So, you can play a soundtracker module whilst using the emulator if you wish!

The sound on the cycle counting version of the emulator and the non-cycle counting version of the emulator are produced in different ways. The non-cycle counting version uses the audio.device for sound output and does not access the sound hardware directly.

However, the cycle-counting version does, although the audio.device is allocated. My apologies for this, but I have been unable to get the audio.device to work as I need it to (and as the rom kernel reference manual documents it), and have had to resort to direct access to the audio hardware. If you don't like this, don't switch on the sound!

1.10 Parallel port support

Parallel Port/Serial Port Emulation

All parallel port output on the Dragon and Serial Port output on the Tandy can be redirected to a file by using the 'Set ParRecord' option on the control window. The user can then specify a file to which output to the parallel/serial port is redirected.

The user is advised that the option is switched off before running any games on the emulator in D32/D64 mode, as many send garbage to the parallel port as a result of generating sound.

The capture can be switched off by pressing the 'Stop ParRecord' button.

NOTE:

The D64 serial port is not supported by the emulation at this point.

1.11 Emulation speed

Speed

One question that's always asked of an emulation - how fast does it go?

On the development systems the speed is as follows for the non-cycle counting version:

A500, 68020 20MHz, Kick 2.04, 1 Mb Chip, 1 Mb Fast, Wb 2.1	- 73%
A1200, 68020 14Mhz, Kick 3.0, 2 Mb Chip, Wb 3.0	- 33%
A1200, 68030 50Mhz, Kick 3.0, 2 Mb Chip. 8 Mb Fast, Wb 3.0	- 190%

These results were obtained using the speedtest3.cas BASIC program.

These figures are not incredibly accurate, because the speed depends on the instruction mix, and display mode that is being displayed on the emulator. For example, 256 pixel 2 colour modes have a low overhead, whilst 64 pixel 4 colour modes are more CPU intensive.

I would expect a 68040 and 68060 users will very probably have an incredibly fast Dragon, unless they use the cycle-counting version!

Any other performance information for other configurations is welcomed.

NOTE:

This version is a fair amount slower than the previous version, (exactly how much will depend on your machine). The reason for this is that previously the 6809 emulation only had to trap writes to the upper 32K of memory.

However, now the emulator need to trap both reads and writes which has resulted in a performance hit :(. I've attempted to keep this as small as possible, but it can be as much as 25% in some cases.

1.12 Example programs

Example Programs

A few example virtual cassettes are supplied with the emulator in the examples directory.

All are basic programs:

modetest.cas - Shows the various Graphical screen modes available
joytest.cas - Shows the positions of the left & right joysticks
speedtest3.cas - A basic speed test using a For..Next loop
speedtest4.cas - A speed test, draws a line accross the display & times it.
semitest.cas - Shows the semigraphics modes supported.

A snapshot is also provided. This shows an example 'starfield' program written to test the emulator:

Starfield.dsn

Press the left joystick's fire button ('Help' key) to exit the starfield.

1.13 Conversion utilities

The Conversion utilities

Here follows a description of the conversion utilities provided with the emulator.

ConvROM

The ConvROM utility is supplied in order to allow the conversion of the D32ROM.DGN file into a format that can be used with the emulator.

(All that it actually does is to strip the first 16 bytes of the file and copy the next 16Kb into the specified file.)

Usage:

ConvROM <infile> <outfile>

<infile> - The path and name of the .DGN file to convert.

<outfile> - The path and name of the binary image to be output.

E.g.

```
ConvROM d32rom.dgn dragon32.rom
```

DSNtoPAK

The DSN to PAK program is unclued with the emulator to convert DSN snapshots to the .PAK format used by CoCo and PC-Dragon. The utility does not currently support the extensions introduced by PC-Dragon, nor does it support compression.

However, conversion to PAK 1.2, 1.3 and 1.4 formats are supported.

Usage:

```
DSNtoPAK [-1.2 | -1.3 | -1.4] [-V] <DSN file> <PAK file>
```

The -1.2, -1.3, -1.4 switches are used to specify the version of the PAK file to produce. The most recent (1.4) is used by default.

The -V (verbose) switch is used to obtain more detailed information about the conversion.

Please note that DSN snapshots must first be uncompressed before conversion if they have been packed with XPK.

PAKtoDSN

The PAKtoDSN program performs the reverse conversion of DSNtoPAK. The utility supports PAK version 1.2, 1.3 and 1.4 and also supports PC-Dragon extensions and compressed snapshots (using the PC-Dragon compression method).

Usage:

```
PAKtoDSN [ -D32 | -D64 | -T ] [ -V ] <PAK file> <DSN file>
```

The -D32, -D64 and -T switches are used to convert the snapshot to a specific architecture (Dragon 32, Dragon 64 and Tandy Coco respectively).

If no switch is supplied, the snapshot is analysed and the architecture which appears to be the one the snapshot uses is selected.

the -V (verbose) switch is used to obtain additional information about the conversion.

NOTE:

Although the snapshot conversion utilities have been given some testing, they are not guaranteed to perform perfect conversions, and some files may trip them up. You are therefore encouraged to keep the original files just in case the conversion doesn't work.

1.14 Restrictions on Distribution and Use

Restrictions on distribution and use

The emulator is released as freeware, though the author retains copyright on it, all are free to use it. The archive must be distributed in an unmodified form *only*.

However, it must *not* be released on any CD compilation (except the Aminet series) without the author's consent. I have given consent for it to appear on the Amiga Format Cover CD.

The author has tested the emulator as widely as possible, but please remember:

If the emulator or utilities supplied with it cause your system to crash, or causes some other damage (data loss, e.t.c.) I am not responsible!

There is no warranty and you use the program at your own risk.

By using the program you agree to these rules!

I stress that I have used the emulator alongside other programs with no problems, but that is just my system..

And finally:

If you don't like it, remember how much I charged you for it - nothing.

1.15 Acknowledgements

Acknowledgements

Thanks to Paul Burgin and Kelton Turvey for information on the Dragon and 6809 microprocessors.

Also thanks to Kelton for the icon.

0Thanks to Nico François / Magnus Holmgren for the great Reqtools.library.

Also, thanks go to Doug and Allan for having a look at the previous version of the emulator.

1.16 Contacting the Author

The Author

I've always been interested in emulators, so I decided to write an emulator for a machine which (as far as I know) hadn't been emulated

on the Amiga. I hope that you enjoy using the emulator, and think that it was worth writing - I do!

If you wish to tell me what you think of the emulator, how to improve it, or just to say hello, my current E-Mail address is:

seansi@mail.globalnet.co.uk

Also, any information on the Tandy Coco 3 would be appreciated.

I will just add that if you criticise the emulator, i'd appreciate it if it was constructive :').

1.17 Problems

Current problems with the emulator:

'Undocumented' effects of 6809 Transfer and Exchange instructions are not implemented.

Glaxxons currently 'locks up' (does not start). I know why this is, as I haven't implemented the 35ns IRQ interrupt. Maybe later!

Soft resets may cause problems with the joystick. Hard-resetting the emulator seems to cure this.

1.18 Development History

Development History of the Emulator
=====

DREaM 1.0B

First released version. (July 1996).

D32 Emulation only.

DREaM 1.01

Internal version, never released.

Added:

- GUI.
- Cycle counting.
- Artifacing.
- Configuration file.

Removed:

Copper display tricks!

(Completed around December 1997)

DREaM 1.02

Second released version.

Added:

D64 and Tandy Coco support.

Conversion Utilities.

Cartridge support.

(Completed June 1997).

1.19 Possible Future Work

Future Work

Well, I seem to have got around to some of the future work that i'd said i'd do in the last version, so here's what *might* be next:

Some work has already gone into:

Implementing the serial port on the Dragon 64.

Disk emulation.

Allow the display to be shown as a window in the Workbench / or a public screen (anyone who is willing to share a good colour re-mapping algorithm with me would be appreciated, as would any programming information on public screens!). A version is working at present on the workbench, but is restricted in the palette it uses and isn't ready for release.

And what may be happen after that...:

Real analogue joystick support by the Emulation (If you don't mind me banging the metal)..

A program to convert dragon cassettes into .cas virtual cassette files.

Perhaps a 6809 Disassembler/Debugger..

One person (Hi Doug!) has asked me to consider Coco 3 emulation. I'll certainly do that if anyone can send me some information on the Coco 3, as I know very little about it's differences as opposed to the Dragon 32.