XRoar 0.23

Dragon and Tandy Colour Computer emulator

This manual is for XRoar (version 0.23), a Dragon and Tandy Colour Computer emulator.

Table of Contents

Introduction

XRoar is a Dragon emulator that runs on a wide variety of platforms. Due to hardware similarities, XRoar also emulates the Tandy Colour Computer (CoCo) models 1 & 2. Some features are:

- Emulates Dragon 32, Dragon 64, Tano Dragon, Tandy CoCo 1 & 2.
- Emulates DragonDOS, Delta and RSDOS disk systems.
- Raw and translated keyboard modes.
- Reads and writes virtual cassettes ('`.cas`' files).
- Reads audio files as cassette input.
- Reads and writes VDK, JVC and DMK format virtual floppy diskettes.
- Save and load snapshots.

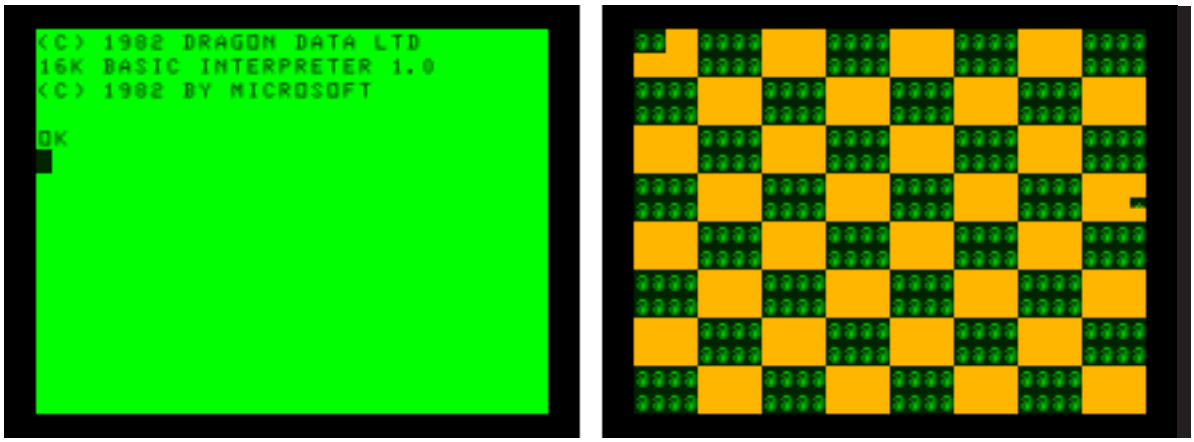
1 Getting started

To start, you will need to acquire (and maybe build) the software and install it. Pre-built binary packages are available from the XRoar home page[1] . If one is not available for your architecture, you will have to build from source. XRoar should build and run on any POSIX-like system for which SDL[2] is available.

You'll also need to get hold of firmware ROM images for the machine you wish to emulate. As Microsoft wrote the BASIC ROM, I don't feel comfortable offering them up myself, but they may well be obtainable elsewhere on the Web.

For instructions on installing from source or binary package, and where to put firmware images, see Chapter 2 [Installation], page 3.

Once you've installed XRoar, run it and an emulator screen should appear. It tries to be a Dragon 64 first, but if ROM images can't be found for that, it then tries Dragon 32 and CoCo in that order. If you just get a strange checkerboard pattern of orange and inverse '`@`' signs, it probably failed to find any ROM images - check that first.



From here you can attach tapes ('`.cas`' or '`.wav`' files) with *`Control+L`*. To load a program from tape, type `CLOADM` (machine code) or `CLOAD` (BASIC). You can hold down *`F12`* to force the emulator to run as fast as possible while loading. If the program does not start automatically when it has loaded (i.e., returns you to the "OK" prompt), you will have to type `EXEC` (machine code) or `RUN` (BASIC) to start it.

XRoar will make use of attached joysticks, but can also emulate them with the cursor keys and *`Left Alt`*. Press *`Control+J`* to cycle through three emulation modes: No joystick emulation (default), Left joystick, Right joystick.

Most of this manual documents how to use the PC builds, where a full keyboard is available. See Chapter 8 [GP32 user interface], page 17 or Chapter 9 [Nintendo DS user interface], page 18 for information about using the hand-held versions.

[1] http://www.6809.org.uk/dragon/xroar.shtml
[2] http://www.libsdl.org/

2 Installation

2.1 Mac OS X binary package

Mount the downloaded disk image and drag the XRoar application icon into your Applications directory.

Firmware ROM images can be stored either in '`~/Library/XRoar/Roms/`' or '`~/.xroar/roms/`'. See Chapter 6 [Firmware ROM image names], page 15 for filenames the emulator will search for.

XRoar can be started by double-clicking the application icon.

2.2 Nintendo DS binary package

Unpack the downloaded ZIP file and copy '`xroar.nds`' to your flash cartridge. Older flash cartridges may need you to DLDI patch the binary in order for the emulator to see files on it. The DLDI wiki[1] has more information.

Create a directory on your flash cartridge called '`/dragon/roms`' and copy any firmware ROM images into this directory. See Chapter 6 [Firmware ROM image names], page 15 for filenames the emulator will search for.

How to start the XRoar application will depend on your flash cartridge, but you probably just need to pick it from a menu (it should appear with a Dragon logo).

2.3 GP32 binary package

Unpack the downloaded ZIP file and copy '`xroar.fxe`' to your SmartMedia card (usually this would be under the '`/gpmm/`' directory).

Create a directory on the card called '`/gpmm/dragon`' to copy your firmware ROM images to. See Chapter 6 [Firmware ROM image names], page 15 for filenames the emulator will search for.

XRoar should appear in the menu when you start up your GP32.

2.4 Windows binary package

First, unpack the downloaded ZIP file. A subdirectory should be created containing the main binary, supporting DLL files and documentation.

Firmware images can be copied to this directory, or in a directory called '*USERPROFILE*`/Application Data/XRoar/roms/`'. *USERPROFILE* is usually something like '`C:/Documents and Settings/`*username*' or '`C:/Users/`*username*'. See Chapter 6 [Firmware ROM image names], page 15 for filenames the emulator will search for.

XRoar can be started by running '`xroar.exe`' either from a file browser or the command line.

[1] http://chishm.drunkencoders.com/DLDI/

2.5 Building from source code

If there is no binary package for your system, you will have to build from source. Before doing so, you should ensure you have the dependencies required to build:

- **SDL**, Simple Directmedia Layer, is used for video and audio output on most supported platforms, and is required to build for all of them. It can be obtained from the SDL home page[2] .
- **SDL_image** is an add-on to SDL that provides the ability to manipulate various graphics formats. It's used in the build process to generate font data from a supplied image file. It can be downloaded from the SDL_image project page[3] .
- **libsndfile** is optional but recommended. It allows XRoar to use audio files (such as WAVs) as a source for cassette input. It is available on the libsndfile home page[4] .
- **GTK+**, the GIMP tookit, is used to provide a file requester on Linux and Unix builds. It is available from the GTK+ home page[5] .

If you use a modern Linux or Unix distribution, it's likely that most of these packages will be installed by default, or easily available through its package management system.

The actual build process should be fairly straightforward and follows the same steps as many other software packages. Unpack the source code, change into the created source directory, run `configure` and then if everything looks good, run `make`. Example:

```
$ gzip -dc xroar-0.23.tar.gz | tar xvf -
$ cd xroar-0.23
$ ./configure
$ make
```

By default, `configure` will set things up to install to '`/usr/local`', but this can be changed by using the '`--prefix=/path/to/destination`' option.

`configure` will detect any optionally supported drivers like Sun audio, OpenGL video, etc.

Once built, run `make install` to install the binary and info documentation on your system. Firmware ROM images should be placed either in your home directory under '`.xroar/roms/`', or in '*PREFIX*`/share/xroar/roms/`', where *PREFIX* is the install prefix as above.

2.6 Cross-compilation of source code

XRoar can be built on one platform to run on another. The Nintendo DS, GP32 and Windows binary packages are all built like this.

To specify a cross-compile, use the '`--target=TARGET`' argument to `configure`. For example, to build for Windows, you might use '`./configure --target=i586-mingw32`'. `configure` will detect Windows headers and configure the build accordingly.

A patch is available for download that increases the overall speed at the expense of some accuracy. This patch is required for the emulator to be fast enough on the GP32 and

[2] http://www.libsdl.org/
[3] http://www.libsdl.org/projects/SDL_image/
[4] http://www.mega-nerd.com/libsndfile/
[5] http://www.gtk.org/

Nintendo DS. It is available from the XRoar home page, and can be applied to the unpacked source tree with the `patch` command.

3 Emulated hardware

3.1 Emulated machines

XRoar will emulate the following machines:

'`dragon32`'
: Dragon 32 (PAL).

'`dragon64`'
: Dragon 64 (PAL).

'`tano`'
: Tano Dragon (NTSC).

'`coco`'
: Tandy Colour Computer (PAL).

'`cocous`'
: Tandy Color Computer (NTSC).

XRoar will try and find a good default machine to emulate based on which ROM images you have installed (see Chapter 6 [Firmware ROM image names], page 15), but you can change it by using the '`-machine MACHINE`' command line argument. Alternatively, once started, pressing *`Control+M`* cycles through all the supported machine types.

Command-line options can be used to specify alternate ROM image files to load:

'`-bas FILENAME`'
: Specify BASIC ROM to use (CoCo only).

'`-extbas FILENAME`'
: Specify Extended BASIC ROM to use.

'`-altbas FILENAME`'
: Specify alternate BASIC ROM (Dragon 64).

'`-noextbas`'
: Disable Extended BASIC.

Dragon machines all contain a complete version of Extended BASIC; CoCos are able to run with a much reduced Color BASIC, with Extended BASIC being optional.

3.2 Video hardware

The Dragon is a UK machine, and as such generates a 50Hz PAL output. The video chip targets 60Hz NTSC displays, so extra work is required to pad this out to a 50Hz signal. This is done by stopping the video chip at two points each field and generating extra scanlines.

However a US version, the Tano Dragon, exists for use with 60Hz NTSC televisions and US CoCo machines all use NTSC. When you select which machine to emulate (see Section 3.1 [Emulated machines], page 6), XRoar picks the appropriate mode. This can be overridden with the '`-pal`' or '`-ntsc`' command line options.

On NTSC televisions, high frequency changes in luminance create cross-colour artifacts (harmonics that interfere with the embedded chroma signal). Some games use this to their advantage, creating colours in the otherwise black and white high resolution video modes. XRoar can approximate the colours generated in these modes to varying levels of detail.

The default approach is to use a 5 bit lookup table, but a faster four colour mode can be selected by running with '`-ccr simple`'.

NTSC machines have another peculiarity in that from reset, the video chip could settle into one of two states (180° out of phase with each other) that affected how the cross-colour interference was interpreted by the television. Games often prompt the user to "Press Enter if the screen is red" (for example) to identify which phase the machine started in. You can adjust which state it's in by pressing *`Control+A`*, which cycles through three artifacted colour modes: Off, Blue-red and Red-blue.

3.3 Audio hardware

The Dragon can route analogue audio from three different sources: attached cartridges, the cassette port and an internal 6-bit DAC. Additionally, a PIA line is connected to the audio output stage, so manipulating that gives a single-bit sound source. XRoar supports the DAC and the single bit audio.

Rarely, a game generates audio by toggling the analogue sound select source rapidly. XRoar will cope with this, but needs to work harder. Disable support for this with the '`-fast-sound`' command line option.

As it is connected to the output of the analogue mux, if the single bit sound output is instead configured as an input, the analogue level will be reflected fed back. While I can't claim that XRoar exactly simulates the electrical characteristics, it will emulate this as a trigger level from the DAC output derived from experimentation.

3.4 Keyboard

The layout of the Dragon's keyboard is a little different to that of modern PCs, so XRoar tries to approximate the Dragon's layout on your PC keyboard as closely as possible, so that game controls will remain in usable positions. That said, they *are* different, so some compromises need to be made: *`Escape`* is mapped to the Dragon's *`BREAK`* key and ` (grave / back-tick) maps to the Dragon's *`CLEAR`* key. There are no good nearby PC keys that directly correspond to the Dragon's cursor keys, so the PC's cursors are used for these.

If you don't use a UK keyboard, but want a close Dragon keyboard layout, you can run with the '`-keymap CODE`' command-line option, where '`CODE`' is a country code: `"uk"` (British), `"us"` (American), `"fr"` (French AZERTY) or `"de"` (German QWERTZ).

XRoar can also be put into `"translated"` keyboard mode, where characters typed on a PC keyboard are translated into the equivalent keystrokes on the Dragon. Press *`Control+Z`* to toggle this mode.

The keyboards of the Dragon and Tandy CoCo are connected in the same way, but the matrix is layed out slightly differently. When you select a machine (see Section 3.1 [Emulated machines], page 6), the appropriate matrix layout is selected for you, but you can toggle between the two layouts by pressing *`Control+K`*.

Additionally, most emulator functions are currently accessed through key combinations. See Section 4.3 [Keyboard commands], page 10 for a list.

3.5 Joysticks

XRoar supports attached joysticks, or can emulate them from the keyboard.

Joystick emulation starts off disabled, but you can cycle through three states by pressing *Control+J*: Off, Left Joystick and Right Joystick.

By default, the first real joystick found is mapped to the Dragon's left joystick port, and the second real joystick to the right port. Left and right joystick mapping can be easily swapped by pressing *Control+Shift+J*.

More fine-grained mappings can be specified with the '`-joy-left`' and '`-joy-right`' command line options. The argument to these command consists of three pairs of numbers in the format '`JOYSTICK-NUMBER,INDEX`'. The pairs map the X axis, Y axis and fire button respectively, and the joystick number is optional if previously specified. For example, '`-joy-left 0,1:0:0`' maps axes 1 and 0 on joystick 0 to the X and Y axis on the left emulated joystick respectively. Button 2 of joystick 0 is mapped to the left fire button.

Previous versions defaulted to a mapping suitable for using a PS2 adaptor. To get this old behaviour, use the command line options '`-joy-left 0,3:2:0 -joy-right 0,0:1:1`'.

3.6 Cassette images

XRoar supports three types of cassette image: '`.cas`' files, audio files such as '`.wav`' and ASCII text files containing BASIC programs ('`.bas`' or '`.asc`'). '`.cas`' files contain a binary representation of what would be loaded from tape, audio files are a recording of the tape itself, and ASCII files contain plain text that is automatically wrapped up as an ASCII BASIC file for loading.

To attach a cassette image, press *Control+L* and select it from the file requester. If instead you press *Control+Shift+L* and select a '`.cas`' file, XRoar can attempt to look into the file and determine whether the program is BASIC or machine code and then autorun it by typing the appropriate BASIC command for you (either `CLOAD` or `CLOADM:EXEC`).

To create a cassette image for writing to (with `CSAVE` or `CSAVEM` for example), press *Control+W* and enter a filename. Created files will be truncated to zero length, so be careful not to overwrite any existing files with this command.

The currently open tape files used for reading and writing are distinct. There is currently no good user interface for tape control, so if you want to rewind a tape, reattach it.

3.7 Floppy disk images

XRoar supports virtual disks by emulating three different types of disk controller cartridge (the default depending on which machine is selected):

'`dragondos`'
: DragonDOS, official DOS cartridge from Dragon Data Ltd.

'`delta`'
: Delta System, Premier Microsystems' DOS for the Dragon.

'`rsdos`'
: RSDOS, Tandy's DOS cartridge for use with the CoCo.

If you want to use the virtual disk support, you will also need a DOS firmware ROM. See Chapter 6 [Firmware ROM image names], page 15 for a list of compatible ROM images per-controller.

To override the defaults, use the following command-line options:

'`-dostype DOS`'
: Type of DOS cartridge ('`-dostype help`' for a list).

'`-dos FILENAME`'
: Specify DOS ROM (or CoCo Disk BASIC).

'`-nodos`'
: Disable DOS (ROM and hardware emulation).

Within the emulator, DOS can be enabled or disabled by pressing *Control+E*. Follow this with a hard reset (*Control+Shift+R*) for it to take effect.

Three virtual disk formats are supported (see Chapter 5 [Supported file types], page 14). Of these, DMK retains the most information about the actual layout of the floppy disk, and is the only one that XRoar will recognise as containing single-density data (as used by the Delta system).

When you attach a disk, it is read into memory, and subsequent disk operations are performed on this in-memory copy. Write protect defaults to off (so write operations on the copy will work), but write back also defaults to off, so updates will not be written to disk. To toggle write protect, press *Control+[5-8]*, where the number to press is the drive number plus 4. To toggle write back, press *Control+Shift+[5-8]*. Even with write back enabled, image files will not be updated until the disk in a virtual drive is changed, or you quit the emulator.

You can create a new blank disk in a virtual drive by pressing *Control+Shift+[1-4]*. You will be a prompted for a filename, and the extension determines which type of file will be written.

3.8 Cartridge ROM images

Some Dragon games are distributed as ROMs on a cartridge ("program pak", in CoCo terminology). These are generally mapped to a specific area in the machine's address space, and trigger an interrupt causing them to be auto-started when the machine is powered on. XRoar can load dumps of these cartridges, and will simulate the auto-start interrupt if requested.

Press *Control+L* to insert a cartridge (files should have the extension '`.rom`'). If you press *Control+Shift+L* instead, the cartridge will be automatically started (you might need to reset the machine with *Control+R* if this doesn't work straight away).

4 User interface

4.1 Video output

Under the SDL user interface, three video output modules are available, selectable with the '`-vo MODULE`' command line option:

'`sdlgl`' OpenGL accelerated video output.

'`sdlyuv`' YUV overlay accelerated video output.

'`sdlgl`' Basic unaccelerated unscalable video output.

When using OpenGL output, the '`-gl-filter`' option selects a filtering method when scaling the image. '`-gl-filter linear`' averages nearby pixels (blur), '`-gl-filter nearest`' selects nearest neighbour pixels (hard edges) and '`-gl-filter auto`' (the default) selects nearest when the image size is an exact integer multiple of the base size, otherise selects linear.

OpenGL output might not be available if you built from source without the appropriate support. Use '`-vo help`' for a list of available modules.

On slower machines, you can specify a value for frameskip with '`-fskip FRAMES`'. For every frame drawn to screen this amount of frames are then skipped before the next one is drawn, reducing the amount of work needed. The default is '`-fskip 0`', meaning no frames are skipped.

XRoar can be started full-screen by specifying '`-fs`'.

4.2 Audio output

Specific audio modules exist for OSS audio, Sun audio and Mac OS X coreaudio. If none of these are available, generic SDL audio will be used.

Use of a specific module can be forced using '`-ao MODULE`'. Use '`-ao null`' to disable audio, or '`-ao help`' for a list of available modules.

4.3 Keyboard commands

XRoar's user interface is currently based around SDL[1] . The emulator video output window is shown, and all operations are performed with keyboard combinations, usually accessed as *`Control+KEY`*:

`Control+[1-4]`
: Insert a virtual disk into drive 1-4. See Section 3.7 [Disks], page 8.

`Control+Shift+[1-4]`
: Insert a blank virtual disk (40TSS) into drive 1-4. See Section 3.7 [Disks], page 8.

`Control+[5-8]`
: Toggle write protect on disk in drive 1-4. See Section 3.7 [Disks], page 8.

[1] http://www.libsdl.org/

Control+Shift+[5-8]
: Toggle write back on disk in drive 1-4. See Section 3.7 [Disks], page 8.

Control+A
: Cycle through cross-colour video modes (hi-res only). See Section 3.2 [Video hardware], page 6.

Control+C
: Quit emulator.

Control+E
: Toggle DOS emulation on/off - reset to take effect. See Section 3.7 [Disks], page 8.

Control+F
: Toggle full screen mode.

Control+I
: Insert a cartridge. See Section 3.8 [Cartridges], page 9.

Control+Shift+I
: Insert a cartridge, no autorun. See Section 3.8 [Cartridges], page 9.

Control+J
: Cycle through joystick emulation modes (None, Left, Right). See Section 3.5 [Joysticks], page 8.

Control+Shift+J
: Swap joystick mapping (left/right). See Section 3.5 [Joysticks], page 8.

Control+K
: Toggle between Dragon and CoCo keyboard layout. See Section 3.4 [Keyboard], page 7.

Control+L
: Load a file (see below).

Control+Shift+L
: Load a file and attempt to autorun it where appropriate.

Control+M
: Cycle through emulated machine types (resets machine). See Section 3.1 [Emulated machines], page 6.

Control+R
: Soft reset emulated machine.

Control+Shift+R
: Hard reset emulated machine.

Control+S
: Save a snapshot. See Section 4.4 [Snapshots], page 12.

Control+W
: Attach a virtual cassette file for writing. See Section 3.6 [Cassettes], page 8.

Control+Z
: Enable keyboard translation mode. See Section 3.4 [Keyboard], page 7.

F12
: While held, the emulator will run at the maximum possible speed.

When using *Control+L* or *Control+Shift+L* to load a file, the action to be taken is determined by its extension. See Chapter 5 [Supported file types], page 14 for details.

XRoar still supports the use of old keyboard commands that were used to attach specific types of file. *Control+B*, *Control+H* and *Control+T* are all now synonymous with *Control+T*.

4.4 Machine snapshots

XRoar can save out a snapshot of the emulated machine state and read such snapshots back in later. To save a snapshot, press *Control+S*. When using *Control+L* to load a file, anything ending in '`.sna`' will be recognised as a snapshot.

What is included in snapshots: Selected machine architecture, complete hardware state, current keyboard map, filenames of attached disk image files.

What is *not* (yet) included: Actual disk image data (only where to find it), attached cassettes or cartridges.

4.5 Trace mode

XRoar contains a "trace mode", where it will dump a disassembly of every instruction it executes to the console. Trace mode defaults to off unless you run with '`-trace`'. Toggle trace mode on or off with *Control+V*.

4.6 Command line options

Many emulator functions can be changed using keyboard shortcuts (see Section 4.3 [Keyboard commands], page 10), but some behaviour can also be changed from the command line. Here's a summary of supported command-line options:

'`-machine MACHINE`'
: Emulated machine ('`-machine help`' for a list). See Section 3.1 [Emulated machines], page 6.

'`-bas FILENAME`'
: Specify BASIC ROM to use (CoCo only)

'`-extbas FILENAME`'
: Specify Extended BASIC ROM to use

'`-altbas FILENAME`'
: Specify alternate BASIC ROM (Dragon 64)

'`-noextbas`'
: Disable Extended BASIC

'`-dostype DOS`'
: Type of DOS cartridge ('`-dostype help`' for a list). See Section 3.7 [Disks], page 8.

‘`-dos FILENAME`’
: Specify DOS ROM (or CoCo Disk BASIC)

‘`-nodos`’
: Disable DOS (ROM and hardware emulation)

‘`-pal`’
: Emulate PAL (50Hz) video. See Section 3.2 [Video hardware], page 6.

‘`-ntsc`’
: Emulate NTSC (60Hz) video. See Section 3.2 [Video hardware], page 6.

‘`-ccr RENDERER`’
: Specify cross-colour renderer (‘`-ccr help`’ for list). See Section 3.2 [Video hardware], page 6.

‘`-ram SIZE`’
: Specify amount of RAM in kilobytes

‘`-fast-sound`’
: Faster but less accurate sound. See Section 3.3 [Audio hardware], page 7.

‘`-ui MODULE`’
: Specify user interface module (‘`-ui help`’ for a list)

‘`-vo MODULE`’
: Specify video module (‘`-vo help`’ for a list)

‘`-gl-filter FILTER`’
: Select OpenGL texture filter (auto, linear, nearest)

‘`-ao MODULE`’
: Specify audio module (‘`-ao help`’ for a list)

‘`-fskip FRAMES`’
: Specify frameskip (default: 0). See Section 4.1 [Video output], page 10.

‘`-load FILENAME`’
: Load or attach ‘`FILENAME`’. See Chapter 5 [Supported file types], page 14.

‘`-run FILE`’
: Load or attach ‘`FILENAME`’ and attempt autorun.

‘`-h, --help`’
: Display help text and exit

‘`--version`’
: Output version information and exit

‘`-keymap CODE`’
: Select host keyboard type by country code (uk, us, fr, de). See Section 3.4 [Keyboard], page 7.

‘`-joy-left [XJ,][-]XA:[YJ,][-]YA:[FJ,]FB`’
‘`-joy-right [XJ,][-]XA:[YJ,][-]YA:[FJ,]FB`’
: Map a joystick. J = joystick number, A = axis number, B = button number a - before axis signifies inverted axis. See Section 3.5 [Joysticks], page 8.

‘`-trace`’
: Start with trace mode on. See Section 4.5 [Trace mode], page 12.

If you run the Windows pre-built binary, you might find that emulator output gets written to a file called ‘`stderr.txt`’ instead of to the console.

5 Supported file types

XRoar can do useful things with a variety of file types. The type of a file is determined by its extension. Supported file extensions are:

Extension	Description	Read/write?
.dmk	Disk image file in a format defined by David Keil. They store a lot of information about the structure of a disk and support both single and double density data. All disk images are manipulated internally in (near enough) this format. See Section 3.7 [Disks], page 8.	Read & write
.jvc .dsk	Disk image file in a basic sector-by-sector format with optional header information. See Section 3.7 [Disks], page 8.	Read-only
.vdk	Another disk image file format. See Section 3.7 [Disks], page 8.	Read-only
.bin	CoCo binary file. XRoar can load these directly into memory and run them.	Read-only
.sna	XRoar snapshot. Contains a complete dump of RAM from a running emulator session along with information like which machine was being emulated, what DOS was attached, etc. See Section 4.4 [Snapshots], page 12.	Read & write
.hex	Intel hex record. An ASCII format that encodes binary data and where in memory to load it.	Read-only
.cas	Cassette file. Simple binary representation of data contained on a tape. Cannot represent silence, or some custom encodings. See Section 3.6 [Cassettes], page 8.	Read & write
.wav	Cassette audio file. XRoar can read sampled audio from original cassettes. Actually, as audio input uses libsndfile, any file with an unknown extension is passed to libsndfile to see if it recognises it as an audio file. See Section 3.6 [Cassettes], page 8.	Read-only
.rom	This represents two things: when starting, XRoar looks for them firmware ROM images with this extension. When subsequently told to load one, however, they are assumed to be dumps of cartridge ROMs. See Section 3.8 [Cartridges], page 9.	Read-only

In general, to load or attach a file, press *Control+L* and choose a file from the requester that appears. What XRoar does with it will depend on its file extension. You can also automatically attach (and optionally start) files from the command line by using the '`-load FILE`' or '`-run FILE`' options. If you '`-load`' or '`-run`' a cassette image, XRoar will automatically disable any DOS cartridge emulation for you, as this can interfere with some cassette-based games. In addition, the first non-option argument to XRoar is taken as a filename and treated as though it were the argument to the '`-run`' option.

6 Firmware ROM image names

XRoar searches for ROM images in a variety of locations. See Chapter 2 [Installation], page 3 for where your particular platform will search. The search path can be overridden by including a colon-separated list of paths in the `XROAR_ROM_PATH` environment variable.

Images are expected to have certain names. Here's a table showing the names it searches for each ROM in each system. ROM images can have either a '`.rom`' or a '`.dgn`' extension. '`.dgn`' files contain a 16 byte header, which is ignored.

Machine	ROM search order	Description
dragon32	d32	Dragon BASIC.
	dragon32	
	d32rom	
	dragon	
dragon64	d64_1	32K-mode Dragon BASIC.
tano	d64rom1	
	dragrom	
	dragon	
	d64_2	64K-mode Dragon BASIC.
	d64rom2	
coco	bas13	Color BASIC 1.3, 1.2, 1.1 or 1.0.
cocous	bas12	
	bas11	
	bas10	
	extbas11	Extended Color BASIC 1.1 or 1.0.
	extbas10	

Further, emulating a floppy drive controller cartridge requires that you have an image of the DOS ROM that would have been part of it.

Controller type	ROM search order	Description
dragondos	dplus49b	DragonDOS (using DOSplus, SuperDOS
	dplus48	or original DragonDOS ROM).
	sdose6	
	sdose5	
	sdose4	
	ddos40	
	ddos15	
	ddos10	
delta	delta	Delta System.
	deltados	
rsdos	disk11	Disk Extended Color BASIC 1.1 or 1.0.
	disk10	

7 Configuration file

All command-line options can also be used as directives in a configuration file called '`xroar.conf`'. This file is searched for in a variety of locations:

System	**Search order**
Unix	Current working directory
Mac OS X	'`~/.xroar/`'
	'`~/Library/XRoar/`'
	'*PREFIX*`/share/xroar/`'
Windows	Current working directory
	'`~/Local Settings/Application Data/XRoar/`'
	'`~/Application Data/XRoar/`'

'`~`' indicates the user's home directory. On Unix systems this is held in the `HOME` environment variable (often '`/home/`*username*'), on Windows systems it is in the `USERPROFILE` environment variable (often '`c:/Documents and Settings/`*username*' or '`c:/Users/`*username*'). *PREFIX* is the installation prefix, usually '`/usr/local`'.

Directives are listed one per line without the leading dashes of the command line option.

In a future release it should be possible to group machine or DOS configurations by some arbitrary name, allowing easy access to sets of configurations. For now though, it is only really useful for specifying a default machine, video output module, joystick mapping, etc.

8 GP32 user interface

The controller can be cycled through four modes by pressing the left shoulder button. These modes are:

Mode	**Button mappings**
Keyboard	*D-pad* selects a key, *B* presses a key, hold down *Right shoulder button* to press shift.
Cursors	*D-pad* maps to Dragon's cursor keys. *B* is shift, *A* is space, *Right shoulder button* is enter.
Right joystick	*D-pad* controls right joystick motion. *B* is fire button. *A* is space, *Right shoulder button* is enter.
Left joystick	*D-pad* controls left joystick motion, with other controls as with right joystick mode.

At any time, pressing *Select* will bring up a menu allowing you to load or save snapshots, toggle DOS emulation, insert a tape or disk, switch between Dragon & CoCo, switch keymaps and reset the machine.

9 Nintendo DS user interface

The touch screen interface is currently quite basic, but functional. Files can be loaded by selecting "Load...", the emulated machine can be changed in the "Machine configuration" menu, and snapshots can be taken with "Save snapshot".

In the "Input configuration" menu, each of the DS buttons can be mapped to an input function - a keypress, a joystick direction or an emulator configuration command. By default, the *D-pad* buttons are mapped to the right joystick, and *A* to the right firebutton. *Y* swaps joysticks for convenience, and *Start* swaps the DS screens, allowing you to use the touch screen as an analogue joystick input.

10 Curses user interface

On compatible systems (probably only Unix-based), an experimental Curses user interface may be available, specified by starting with the '`-ui curses`' option. This uses Unix terminal capabilities to render a text-only view of the video output. Keyboard commands are broadly the same as those defined for the SDL user interface.

11 Acknowledgements

The Mac OS X 'Carbon' file requester code is based on a contribution by Stuart Teasdale.

I made reference to the MAME 6809 core for clues on how the overflow bit in the condition code register was handled.

The rest is the result of reading too many datasheets.

Thanks to all the people on the Dragon Archive Forums[1] for helpful feedback and insight.

[1] http://archive.worldofdragon.org/phpBB3/