XRoar 0.28

Dragon and Tandy Colour Computer emulator

This manual is for XRoar (version 0.28), 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 (compact '`.cas`' files and audio files).
- Reads and writes VDK, JVC and DMK format virtual floppy diskettes.
- Save and load snapshots.

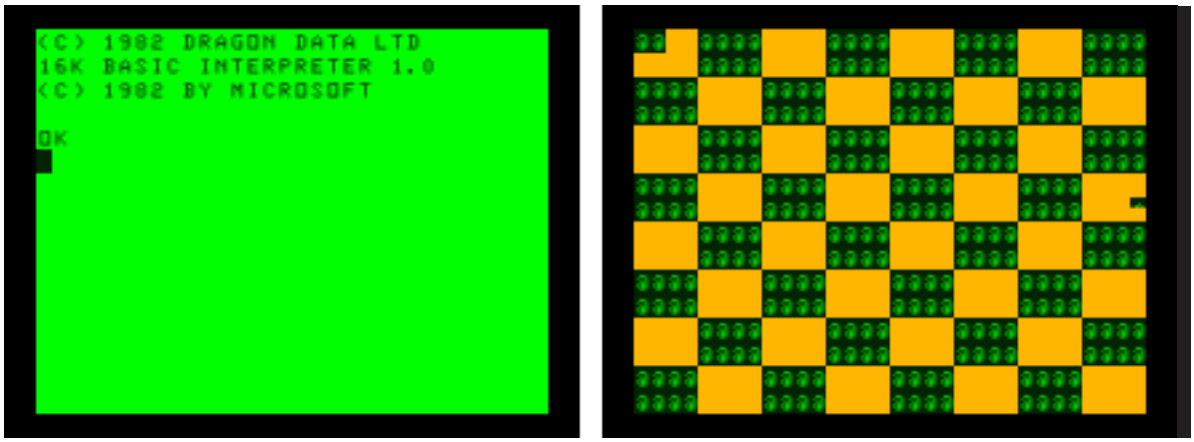
1 Getting started

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 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 ROM images, see Chapter 2 [Installation], page 3.

Once you've installed XRoar, run it and an emulator screen should appear. Depending on which ROM images are found, XRoar will attempt to emulate a Dragon 64, a Dragon 32 and a 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). 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.

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

2 Installation

2.1 Windows binary package

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

ROM images can be copied to this directory, or to a directory called '*USERPROFILE*`/Application Data/XRoar/roms/`'. *USERPROFILE* is usually something like '`C:/Documents and Settings/`*username*' or '`C:/Users/`*username*'. ROM images in these locations with standard filenames will be found automatically. See Chapter 6 [ROM lists], page 18 for information on filenames, and modifying the search lists.

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

2.2 Mac OS X binary package

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

ROM images can be copied either to '`~/Library/XRoar/Roms/`' or '`~/.xroar/roms/`'. ROM images in these locations with standard filenames will be found automatically. See Chapter 6 [ROM lists], page 18 for information on filenames, and modifying the search lists.

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

2.3 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:

- **GTK+**, the GIMP tookit, is used to provide (at least) a file requester on Linux and Unix builds. It is available from the GTK+ home page[1].
- **GtkGLExt**, an OpenGL extension to GTK+. This is optional, but if present, a full GTK+ user interface can be presented. Under Linux, this obviates the need for SDL entirely. It is available from the GTK+ OpenGL Extension project page[2].
- **SDL**, Simple Directmedia Layer, is used for video and audio output on most supported platforms. It can be obtained from the SDL home 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].

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:

[1] http://www.gtk.org/
[2] http://projects.gnome.org/gtkglext/
[3] http://www.libsdl.org/
[4] http://www.mega-nerd.com/libsndfile/

```
$ gzip -dc xroar-0.28.tar.gz | tar xvf -
$ cd xroar-0.28
$ ./configure
$ make
```

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

By default, `configure` will set up an install *PREFIX* of '`/usr/local`', but this can be changed by using the '`--prefix=PATH`' option.

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

2.4 Cross-compilation of source code

XRoar can be built on one platform to run on another. The Windows binary package is 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.

3 Emulated hardware

3.1 Emulated machines

XRoar has built-in definitions for the following machines, selectable with the '`-machine NAME`' option:

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

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

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

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

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

If no machine is specified on the command line (with '`-machine NAME`'), XRoar will try and find a good default machine to emulate based on which ROM images you have installed (see Chapter 6 [ROM lists], page 18). Alternatively, once started, pressing *`Control+M`* cycles through all the supported machine types.

Additionally extra machines can be configured, or existing ones reconfigured, with the following options:

'`-machine NAME`'
: Select existing or configure new machine ('`-machine help`' for list).

'`-machine-desc TEXT`'
: Machine description (showed in '`-machine help`').

'`-machine-arch ARCH`'
: Machine architecture. One of "dragon64", "dragon32" or "coco".

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

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

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

'`-nobas`'
: Disable BASIC.

'`-noextbas`'
: Disable Extended BASIC.

'`-noaltbas`'
: Disable alternate BASIC (Dragon 64).

'`-tv-type TYPE`'
: Set TV type ("pal" or "ntsc"). See Section 3.2 [Video hardware], page 6.

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

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.

Defining extra machines is most usefully done in the configuration file. For example:

```
machine pippin
machine-desc Dragon Pippin (prototype)
machine-arch dragon32
ram 16
```

This will define a machine named “pippin” that is basically a Dragon 32 with only 16K or RAM.

3.2 Video hardware

UK machines generate a 50Hz PAL display, whereas US machines output 60Hz NTSC. XRoar will emulate the appropriate mode depending on the machine chosen (see Section 3.1 [Emulated machines], page 5), but you can force one or the other with the ‘`-pal`’ or ‘`-ntsc`’ command line options.

Many programs employ a trick with cross-colour on NTSC televisions to generate a colour display from an otherwise black & white video mode. 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 start in one of two cross-colour states at random. Games often prompt the user to “Press Enter if the screen is red” (for example) to identify which state 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, single bit audio, and will approximate cassette audio input.

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.

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), "fr_CA" (Canadian French QWERTY) 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. Run with the '`-kbd-translate`' option to start with this enabled. 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 laid out slightly differently. When you select a machine (see Section 3.1 [Emulated machines], page 5), 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 4.6 [Keyboard commands], page 11 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. When emulating a joystick, the cursor keys control the axes and *Left Alt* maps to the fire button.

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 for reading, use the '`-load`' or '`-run`' command line options, or press *Control+L* and select it from the file requester (hold *Shift* as well to attempt to autorun).

To create a cassette image for writing (with the `CSAVE` or `CSAVEM` BASIC commands, for example), use the '`-tape-write`' command line option, or 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.

Four command line options affect how tapes are read:

The '`-tape-fast`' option accelerates tape loading by intercepting ROM calls. Disable with '`-no-tape-fast`'. On by default.

The '`-tape-pad`' option tries to make loading more reliable by intercepting ROM calls and inserting extra leader bytes where appropriate. Disable with '`-no-tape-pad`'.

The '`-tape-pad-auto`' option will, for '`.cas`' files, automatically switch on leader padding when insufficient initial leader bytes are found. Disable with '`-no-tape-pad-auto`'. On by default.

The '`-tape-rewrite`' option enables rewriting of anything read from the input tape to the output tape. This is useful for creating "well formed" cassette images.

Where available, these options can be changed on the fly in the GUI. See Section 4.4 [Tape control], page 11.

3.7 Cartridges

XRoar has built-in definitions for three cartridges, selectable with the '`-cart NAME`' option:

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

'`delta`'
: Delta System, Premier Microsystems' disk controller cartridge for the Dragon.

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

Additionally extra cartridges can be configured, or existing ones reconfigured, with the following options:

'`-cart NAME`'
: Select existing or configure new cartridge ('`-cart help`' for list). If NAME is not an existing cartridge and looks like a loadable ROM image filename ('`.rom`' or '`.dgn`' filename extension), a ROM cart is defined for that filename.

'`-cart-desc TEXT`'
: Cartridge description (showed in '`-cart help`').

'`-cart-type TYPE`'
: Cartridge type. A type of "rom" will define a ROM cartridge (e.g., for a game). The type can also be one of "dragondos", "delta" or "rsdos", which will define a cartridge with a base type according to the list above.

'`-cart-rom FILENAME`'
: ROM image to load ($C000-).

'`-cart-rom2 FILENAME`'
: Second ROM image to load ($E000-).

If no ROM is configured for a cartridge, there is a built-in list to search for each of the disk controller types. A ROM image will be required if you want to use virtual disks.

Defining extra cartridges is most usefully done in the configuration file, for example:

```
cart sdose6
cart-desc SuperDOS E6
cart-type dragondos
cart-rom sdose6
cart-rom2 dosdream
```

This will define a cartridge called "sdose6" as a DragonDOS cartridge with its ROM replaced with '`sdose6`', and an additional ROM called '`dosdream`'.

XRoar will automatically attempt to find a disk controller cartridge relevant to the current machine unless the '`-nodos`' option is specified.

Selecting a ROM image file with the '`-load`' or '`-run`' command line options, or with *`Control+L`* or *`Control+Shift+L`*, will attach a ROM cartridge.

Within the emulator, cartridges can be enabled or disabled by pressing *`Control+E`*. You will almost certainly want to follow this with a hard reset (*`Control+Shift+R`*).

3.8 Floppy disk images

If a disk controller cartridge is selected, XRoar supports virtual disks.

Three virtual disk formats are supported (see Chapter 5 [Supported file types], page 17). 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 enable defaults to on (so write operations on the copy will work), but write back defaults to off, so updates will not be written to the disk image file. To toggle write enable, 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.

Where available, these options can also be changed on the fly in the GUI. See Section 4.3 [Drive control], page 10.

Write back can be set to default to on with the '`-disk-write-back`' command line option.

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.

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.

'`sdl`' 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, ALSA, Sun audio, Mac OS X coreaudio and PulseAudio. 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.

For most audio modules, the '`-ao-rate HZ`' option can be used to specify a sample rate in Hz. The default will usually be 44100. The '`-ao-buffer-ms MS`' or '`-ao-buffer-samples N`' options may be used (where supported) to set an audio buffer size, either in milliseconds or number of samples.

4.3 Drive control

Currently only available in the GTK+ interface. Pressing *Control+D*, or selecting "Drive Control" from the "Tool" menu will open the drive control window.

This window allows you to insert and eject disk images, and toggle their write-enable and write-back states. See Section 3.8 [Disks], page 9.

4.4 Tape control

Currently only available in the GTK+ interface. Pressing *Control+T*, or selecting "Tape Control" from the "Tool" menu will open the tape control window.

This window shows the current tape image filename and position. The input tape image is scanned, and any recognised file header blocks listed, along with their position within the image. Double clicking a filename will seek to that point in the tape image.

Certain tape options can be configured here. See Section 3.6 [Cassettes], page 7.

4.5 Printing

XRoar supports redirecting Dragon printer output to a file or pipe with the '`-lp-file`' or '`-lp-pipe`' option. Printed data will be sent to the appropriate stream. Pressing *Control+Shift+P* will flush the current stream by closing it (so if the stream is a pipe, the filter will complete). The stream will be re-opened when any new data is sent.

The pipe feature allows you to use useful print filters such as `enscript`, e.g., '`-lp-pipe ''enscript -B -N r -d printer-name''`'. This will send a job to your printer, using carriage returns as line feeds (the Dragon default), each time you press *Control+Shift+P* (or exit the emulator).

4.6 Keyboard commands

XRoar's user interface is currently based around either GTK+[1] or SDL[2]. The emulator video output window is shown, and all operations are performed with keyboard combinations, usually accessed as *Control+KEY*. Under Mac OS X, most functions will be accessible as *Command+KEY*.

Control+[1-4]
: Insert a virtual disk into drive 1-4. See Section 3.8 [Disks], page 9.

Control+Shift+[1-4]
: Insert a blank virtual disk (40TSS) into drive 1-4. See Section 3.8 [Disks], page 9.

Control+[5-8]
: Toggle write enable on disk in drive 1-4. See Section 3.8 [Disks], page 9.

Control+Shift+[5-8]
: Toggle write back on disk in drive 1-4. See Section 3.8 [Disks], page 9.

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 3.8 [Disks], page 9.

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

Control+F
: Toggle full screen mode.

Control+J
: Cycle through joystick emulation modes (None, Left, Right). See Section 3.5 [Joysticks], page 7.

Control+Shift+J
: Swap joystick mapping (left/right). See Section 3.5 [Joysticks], page 7.

Control+K
: Toggle between Dragon and CoCo keyboard layout. See Section 3.4 [Keyboard], page 6.

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 5.

Control+Shift+P
: Flush printer output. See Section 4.5 [Printing], page 11.

Control+R
: Soft reset emulated machine.

Control+Shift+R
: Hard reset emulated machine.

Control+S
: Save a snapshot. See Section 4.7 [Snapshots], page 13.

Control+T
: Open the tape control window (certain user interfaces only). See Section 4.4 [Tape control], page 11.

Control+W
: Attach a virtual cassette file for writing. See Section 3.6 [Cassettes], page 7.

Control+Z
: Enable keyboard translation mode. See Section 3.4 [Keyboard], page 6.

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 17 for details.

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

4.7 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.

4.8 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*.

4.9 Command line options

Many emulator functions can be changed using keyboard shortcuts (see Section 4.6 [Keyboard commands], page 11), but some behaviour can also be changed from the command line.

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.

4.9.1 Emulated machine options

See Section 3.1 [Emulated machines], page 5 for more information.

'`-machine NAME`'
: Select/configure machine ('`-machine help`' for a list).

'`-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

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

'`-tv-type TYPE`'
: Set TV type ("pal" or "ntsc"). See Section 3.2 [Video hardware], page 6.

'`-ram SIZE`'
: Specify amount of RAM in kilobytes

4.9.2 Emulated cartridge options

See Section 3.7 [Cartridges], page 8 for more information.

'`-cart NAME`'
: Select/configure cartridge (-cart help for list). See Section 3.7 [Cartridges], page 8.

'`-cart-desc TEXT`'
: Set cartridge description.

'`-cart-type TYPE`'
: Set cartridge type (-cart-type help for list). Cartridge type. A type of "rom" will define a ROM cartridge (e.g., for a game). The type can also be one of "dragondos", "delta" or "rsdos", which will define a cartridge with a base type according to the list above.

'`-cart-rom FILENAME`'
: The ROM image specified will be mapped to $C000-.

'`-cart-rom2 FILENAME`'
: The ROM image specified will be mapped to $E000-.

4.9.3 Options to attach files

'`-romlist NAME=LIST`'
: Define a ROM list. See Chapter 6 [ROM lists], page 18.

'`-romlist-print`'
: Print defined ROM lists.

'`-crclist NAME=LIST`'
: Define a ROM CRC list. See Chapter 6 [ROM lists], page 18.

'`-crclist-print`'
: Print defined ROM CRC lists.

'`-force-crc-match`'
: Force per-architecture CRC matches.

'`-load FILENAME`'
: Load or attach '`FILENAME`'. See Chapter 5 [Supported file types], page 17.

'`-run FILENAME`'
: Load or attach '`FILENAME`' and attempt autorun.

'`-tape-write FILENAME`'
: Open '`FILENAME`' for tape writing.

'`-lp-file FILENAME`'
: Append Dragon printer output to FILENAME.

'`-lp-pipe COMMAND`'
: Pipe Dragon printer output to COMMAND.

4.9.4 Automatic actions

‘`-type STRING`’
: Intercept ROM calls to type STRING into BASIC.

4.9.5 Emulator interface options

‘`-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)

‘`-ao-rate HZ`’
: Set audio sample rate (if supported by module)

‘`-ao-buffer-ms MS`’
: Set audio buffer size in ms (if supported)

‘`-ao-buffer-samples N`’
: Set audio buffer size in samples (if supported)

‘`-volume VOLUME`’
: Specify audio volume (0 - 100)

‘`-fast-sound`’
: Faster but less accurate sound. See Section 3.3 [Audio hardware], page 6.

‘`-fs`’ Start emulator full-screen if possible.

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

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

‘`-keymap CODE`’
: Select host keyboard type by country code (uk, us, fr, de). See Section 3.4 [Keyboard], page 6.

‘`-kbd-translate`’
: Enable keyboard translation mode. See Section 3.4 [Keyboard], page 6.

‘`-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 7.

‘`-tape-fast`’
: Enable fast tape loading. See Section 3.6 [Cassettes], page 7.

‘`-tape-pad`’
: Enable tape leader padding. See Section 3.6 [Cassettes], page 7.

‘`-tape-pad-auto`’
: Detect need for leader padding automatically. See Section 3.6 [Cassettes], page 7.

‘`-tape-rewrite`’
: Enable tape rewriting. See Section 3.6 [Cassettes], page 7.

‘`-disk-write-back`’
: Default to enabling write-back for disk images. See Section 3.8 [Disks], page 9.

‘`-trace`’ Start with trace mode on. See Section 4.8 [Trace mode], page 13.

4.9.6 Other options

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

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

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 3.8 [Disks], page 9.	Read & write
.jvc .dsk	Disk image file in a basic sector-by-sector format with optional header information. See Section 3.8 [Disks], page 9.	Read-only
.vdk	Another disk image file format. See Section 3.8 [Disks], page 9.	Read-only
.bin	Binary file (DragonDOS or CoCo). XRoar can load these directly into memory and optionally autorun 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 4.7 [Snapshots], page 13.	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 7.	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 7.	Read-only
.rom	This represents two things: when starting, XRoar looks for ROM images with this extension. When subsequently told to load one, however, they are assumed to be dumps of cartridge ROMs. See Section 3.7 [Cartridges], page 8.	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 ROM lists

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. There are a set of built-in lists which can be modified using the '`-romlist`' option. List elements are comma-separated, and an element prefixed with an '`@`' character indicates a nested list. As an example, to append the ROM image "d64extended" to the "d64_1" list, use '`-romlist d64_1=@d64_1,d64extended`'.

ROM images are searched for with either a '`.rom`' or a '`.dgn`' extension. '`.dgn`' files contain a 16 byte header, which is ignored.

View the currently defined ROM lists with the '`-romlist-print`' option, but here are the canonical names for BASIC ROM images:

Machine	**ROM image name**	**Description**
dragon32	d32	Dragon 32 BASIC.
dragon64	d64_1	32K-mode Dragon 64 BASIC.
tano	d64_2	64K-mode Dragon 64 BASIC.
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	

Emulating a floppy drive controller cartridge requires that you have an image of the DOS ROM that would have been part of it. The same ROM lists are consulted, but here are the canonical names for DOS ROM images:

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

A CRC32 value is calculated (and reported) for BASIC ROMs loaded - this is used to determine whether certain breakpoints can be used (e.g., for fast tape loading). The lists of CRCs matched can be defined in a similar way to ROM lists using the '`-crclist`' option - you might do this if you have a modified version of a BASIC ROM that maintains compatible entry points with an original. View the current list with '`-crclist-print`'.

Sometimes it may be useful to force CRC matching so that breakpoints apply (e.g., you are modifying a ROM image and don't wish to have to add its CRC to the match list each time you modify it). The '`-force-crc-match`' option forces the CRCs to be as if an original ROM image were loaded.

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.

8 Acknowledgements

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

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

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