

**ASp**

Ian Greenway

**COLLABORATORS**

	<i>TITLE :</i> ASp		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Ian Greenway	August 9, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ASp</b>	<b>1</b>
1.1	ASp Guide . . . . .	1
1.2	Introduction . . . . .	1
1.3	What's a Spectrum? . . . . .	2
1.4	ASp System Requirements . . . . .	2
1.5	ASp Installation . . . . .	3
1.6	Which ROM files to use? . . . . .	3
1.7	ASp Quick Start Guide . . . . .	4
1.8	ASp Feature Guide . . . . .	4
1.9	What features are coming up? . . . . .	4
1.10	Detailed Usage Guide for ASp. . . . .	5
1.11	Basic Operation Overview . . . . .	5
1.12	The RUN GUI function . . . . .	6
1.13	The STOP GUI function . . . . .	6
1.14	The RESET GUI function . . . . .	7
1.15	The LOAD GUI function . . . . .	7
1.16	The RELOAD GUI function . . . . .	7
1.17	The SAVE GUI function . . . . .	7
1.18	The INSERT GUI function . . . . .	7
1.19	The EJECT GUI function . . . . .	7
1.20	The REWIND GUI function . . . . .	8
1.21	How to Quit ASp . . . . .	8
1.22	Keyboard layout information . . . . .	8
1.23	Types of Joystick available in ASp . . . . .	9
1.24	Kempston Joystick operation . . . . .	9
1.25	Smart Kempston Joystick operation . . . . .	9
1.26	Cursor Joystick operation . . . . .	9
1.27	Sinclair 1 Joystick operation . . . . .	9
1.28	Sinclair 2 Joystick operation . . . . .	10
1.29	Custom Keys Joystick operation . . . . .	10

---

---

1.30	Loading Snapshots	10
1.31	Saving Snapshots	10
1.32	Using TAPe files	11
1.33	Description of .Z80 files	11
1.34	Description of .SNA files	11
1.35	Description of .TAP files	12
1.36	Settings->Misc->Add AY chip to 48K model	12
1.37	Settings->Misc->Keep Kempston Interface	12
1.38	Settings->Misc->Higher Accuracy Timing	12
1.39	Settings->Misc->Delay during TAP load	13
1.40	Audio Save option	13
1.41	Emulated Machine Type Selection	13
1.42	Audio Configuration	14
1.43	Audio Channel Mode	14
1.44	Audio Sample Rate selection	14
1.45	Settings->Audio->Quiet 48K Beeps	14
1.46	Filter Control	15
1.47	Video configuration options	15
1.48	Screen Mode selection	15
1.49	Frame Skip function	15
1.50	Miscellaneous configuration options	16
1.51	ASp ToolTypes	16
1.52	ASp Command-Line arguments	16
1.53	AREXXPORTNAME Argument	17
1.54	BLANKPOINTER Argument	17
1.55	NOFKEYS Argument	17
1.56	LOADPATH Argument	17
1.57	SAVEPATH Argument	17
1.58	ROMPATH Argument	18
1.59	ROMNAME48 Argument	18
1.60	ROMNAME128 Argument	18
1.61	ROMNAME128PLUS2 Argument	18
1.62	FKEYS Argument	18
1.63	NOBLANKPOINTER Argument	19
1.64	Copyrights on Spectrum Software	19
1.65	ARexx port functions	19
1.66	Running an ARexx script from within ASp	19
1.67	ASp ARexx Commands	20
1.68	ASp ARexx important information	20

---

---

1.69 ARexx command: HELP . . . . .	21
1.70 ARexx command: GETVERSION . . . . .	21
1.71 ARexx command: RUN . . . . .	21
1.72 ARexx command: STOP . . . . .	22
1.73 ARexx command: RESET . . . . .	22
1.74 ARexx command: QUIT . . . . .	22
1.75 ARexx command: LOAD . . . . .	22
1.76 ARexx command: SAVE . . . . .	23
1.77 ARexx command: RELOAD . . . . .	23
1.78 ARexx command: INSERT . . . . .	23
1.79 ARexx command: REWIND . . . . .	24
1.80 ARexx command: POKE . . . . .	24
1.81 ARexx command: PEEK . . . . .	24
1.82 ARexx command: READRAM . . . . .	24
1.83 ARexx command: WRITERAM . . . . .	25
1.84 ARexx command: SETPAGE . . . . .	25
1.85 ARexx command: GETPAGE . . . . .	25
1.86 Demo script: LoadSCR.asprx . . . . .	26
1.87 Demo script: SaveSCR.asprx . . . . .	26
1.88 Demo script: POKEr.asprx . . . . .	26
1.89 Copyrights and Distribution . . . . .	26
1.90 Credits . . . . .	26
1.91 Version History . . . . .	27
1.92 Frequently Asked Questions . . . . .	30
1.93 Contended RAM . . . . .	33
1.94 Information about ASp's use of mmu.library . . . . .	33
1.95 What's an MMU? . . . . .	33
1.96 World Of Spectrum . . . . .	34
1.97 ASp Support Web Pages . . . . .	34
1.98 Contacting the Author . . . . .	34
1.99 Index . . . . .	34

---

# Chapter 1

## ASp

### 1.1 ASp Guide

Amiga Spectrum emulator

V0.76

(c) Ian Greenway 1999,2000

[Introduction](#)

[Requirements](#) <-- Important! Read this!

[Installation](#)

[Quick start](#)

[Full usage](#)

[Copyright and distribution](#)

[Credits](#)

[Contact](#)

[History](#)

[Features](#)

[To Do](#)

[FAQ's](#)

[Index](#)

### 1.2 Introduction

ASp is a [Sinclair Spectrum](#) emulator for Amiga computers.

It provides a comprehensive and accurate emulation of the 48K and 128K Spectrum models with fully configurable stereo sound and authentic graphics. The design philosophy for ASp is to create an easy-to-use emulator which runs almost all Spectrum software and has a good selection of features.

---

## 1.3 What's a Spectrum?

No... I have difficulty in believing you don't know, surely...

Spectrum were 8-bit computers devised by Sir Clive Sinclair and were very popular during the 1980's. It is said that the 48K Spectrum is the biggest selling single model of computer ever, including PC's since they are never the same model twice. And no, PSX doesn't count! :)

Spectrum, or Speccys as they are often known, had a modest specification for the time, being mainly designed to be cheap and accessible. The CPU was a 3.5MHz Z80 (pretty much the best 8-bit CPU made), with a humble 48Kb of RAM, 16Kb of ROM and a simple tape interface. The graphics were technically 16-colour, but limited to 2 colours per 8x8 character block, with bright and darker shades of a basic 8 colours making the 16. There was a built-in BASIC interpreter which was slow but comprehensive and used an initially off-putting keyword entry system.

Later models boasted 128Kb of memory and a 32Kb or more ROM. Higher speed tape drives and disk drives were available as accessories, and a plethora of interfaces and add-ons for joysticks, printers, modems and other ingenious devices were invented.

The rivalry between the Spectrum and the Commodore 64 was the driving force for the explosion in the home computer gaming market during the mid 80's, providing a rich arena for the more powerful 16-bit machines like Amiga and Atari to enter. This next generation of computers hailed the end of the commercial viability of the 8-bit machines, although Amstrad, now the owners of the Sinclair Spectrum technology, made a couple of attempts to re-release the machines.

Despite even the next wave of computers, 32-bit, to arrive on the scene, there was still a massive following of nostalgia for the old Spectrum. These new machines offered the opportunity to create full-speed emulation of the old favourites and many emulators have been written. Amstrad took the unheard-of step of declaring the Spectrum technology free for emulator use, a generous move.

Games and computer technology has now moved on. The generation who fondly remembers the 8-bit days have been replaced with the new voices of those who think even 32-bit is out of date. Despite that, you can still see the roots of most modern game concepts in those deceptively clunky Speccy games, which you can still play for "five minutes" and discover an hour has unaccountably passed...

## 1.4 ASp System Requirements

In order to run ASp you absolutely NEED the following:

- 68020 or better.
- Appx. 1Mb free RAM.
- Workbench/Kickstart 3.0+
- mmu.library V42.
- A working MMU.
- The Spectrum 48, 128 and Plus 2 ROM files.

To get useful performance, it is recommended that you have:

- 33MHz 68040 or better.
- A couple of megs of free RAM.
- AGA users install BlazeWCP by Rick Pratt.

The [mmu.library](#) is part of the MuLib distribution by Thomas Richter. The entire package is extremely comprehensive, but the only necessity for ASp to run is for mmu.library to be in your LIBS: directory. MuLib is available from Aminet, or a special small distribution archive is also available from the [ASp Homepage](#).

The Spectrum ROM files are legally redistributable and can be obtained from various resources, including the ASp homepage. Or [read here](#) for exactly which ROM images are required.

Note that the ROM file(s) remain Copyright Amstrad Plc. who have kindly granted permission for use with emulation software.

---

## 1.5 ASp Installation

Installing ASp is very straightforward. There is no install-script!

Copy the ASp program directory with all contents and subdirectories to anywhere which takes your fancy.

If you have the distribution archive which contains the ROM files, then it is already set up to run "as-is". Just de-archive into the desired place.

If not, either:

Copy the required Spectrum ROM files ( [details here](#) ) into that same directory,

Or:

Change the ASp icon to include a tooltype 'ROMPATH="`<directory>`"' where `<directory>` is the location you have stored the ROM files. Additionally, you may specify the exact path and/or name of each of the files required using the [tooltypes](#) .

Note: If the ROM files are placed in the same directory as the ASp program file, be sure to remove or disable the ROMPATH tooltype, or set it to read "PROGDIR:", which is the default.

## 1.6 Which ROM files to use?

ASp requires ROM image files of the Spectrum 48K, 128K and 128K+2 computers.

These files are freely available from the [ASp web pages](#) , [World Of Spectrum](#) , and other public resources.

Unlike ROM images for most other systems, this IS legal! Amstrad Plc. have kindly granted permission for free distribution of the Spectrum ROM images for which they hold copyright, for use with emulation software. Amstrad Plc. retain copyright while allowing redistribution.

You can use any Spectrum ROM files with ASp, including those which have been "bugfixed" or otherwise modified.

ASp will look for the following files in order, using the first ones it finds:

The Spectrum 48K ROM:

The 16Kb file from the "ROMNAME48" argument, if present.

Or: "Spectrum.rom" (16Kb)

Or: "48.ROM" (16Kb)

The Spectrum 128K ROM:

The 32Kb file from the "ROMNAME128" argument, if present.

Or: "Spec128.rom" (32Kb)

Or: "ZX128\_0.ROM" (16Kb) and "ZX128\_1.ROM" (16Kb)

The Spectrum 128+2 ROM:

The 32Kb file from the "ROMNAME128PLUS2" argument, if present.

Or: "Plus2.rom" (32Kb)

Or: "ZXP2\_0.ROM" (16Kb) and "ZXP2\_1.ROM" (16Kb)

These files are, by default, placed in the same directory as the ASp executable. You can use the ROMPATH argument to specify a different default path to use for the ROM files. A fully qualified path and file definition for any ROMNAMExxx argument will override the ROMPATH argument for that file only.

The ROMPATH and/or ROMNAMExxx arguments can be supplied via [Tooltypes](#) or [command line arguments](#) , or a mixture of both.

ASp must find suitable files for each of the three ROM images before it will successfully initialise.

ASp will load and use any files of the correct name and length. Use of files which are not real Spectrum ROM images should not cause any harm, but will very likely prevent any useful functionality.

---

## 1.7 ASp Quick Start Guide

A lot of people will be very familiar with the principles of emulators, both in general and specifically Spectrum ones. This is a short introduction intended to get you up and running. If you need more explanations of features, terminology or usage please read the full detailed usage section, or look up individual features in the index.

Please do not submit bug reports or feature requests without first reading the relevant detailed section.

Read the [Requirements](#) and [Installation](#) sections, copy all the files as required and double-click the ASp icon to begin.

You should see a small ASp GUI window containing several buttons appear on your default public screen.

The emulation is controlled by the RUN and STOP buttons. You can also press the Escape key to Stop the emulator. "Load" will load a .SNA or .Z80 snapshot, or "Insert" will define a ".TAP" file to use.

You can press RESET at any time to reset the emulated Spectrum.

The Amiga keyboard behaves exactly as would a Spectrum keyboard, when ASp is running. To access the extended Spectrum BASIC commands, use Shift as "Caps Shift" and Alt as "Symbol Shift". Several shortcuts are available, including the cursor keys, backspace, and some symbol keys.

Be aware that some Amiga keyboard controllers have problems with multiple simultaneous key presses. This is a limitation of the keyboard controller, not of ASp. This is only really a problem for games. If the game has no joystick option then it is suggested you investigate the [Custom-Keys joystick](#) .

A standard digital joystick plugged into the normal Amiga GamePort will behave as a Spectrum joystick of the type set in the Settings menu. Default is Kempston.

A range of audio and video options are available which will affect the accuracy of the emulation in exchange for potential speed improvements.

To quit ASp, close its window or select the Quit menu item.

## 1.8 ASp Feature Guide

System compliant, fully multitasking.

Font-sensitive GadTools GUI.

Small, easy-to-use, authentic emulation of a 48K/128K/+2 Sinclair Spectrum.

Comprehensive, speed-regulated Z80-CPU core which includes all undocumented instructions and most undocumented flag behaviour.

Genuine appearance Spectrum display with full colour and support for BRIGHT, FLASH and rainbow effects.

Stable, configurable stereo sound (48K beeper and 128K AY-3-8912 chip).

Support for various joystick types.

Loads .SNA, .Z80 and .TAP files.

Saves .SNA and .Z80 files.

ARexx port for remote control and scripted operation.

## 1.9 What features are coming up?

Just because ASp doesn't currently have a feature does not mean it has never occurred to me! :) Indeed, I have had all sorts of plans for ASp, some of which I've rejected on the grounds of time, information, practicality, popularity, etc.

If you have an idea for a feature you'd like to see, [let me know](#) .

Here are some things I'd like to implement at some time (no particular order)...

---

A status line.

Window on Workbench display.

"No Audio" option.

Full rainbow borders.

Native-only (faster) display.

More ARexx commands.

Load via XFD.library.

Load/Save Settings function.

Auto cheat-finder/poker.

IFF screen grab.

TZX file loading.

TAP file saving.

Dynamic JIT compiling emulation core.

More snapshot formats.

Disk format support.

## 1.10 Detailed Usage Guide for ASp.

Please select the following links for information relating to that feature.

[Basic Operation](#)

[Keyboard usage](#)

[Joystick usage](#)

[Loading Snapshots](#)

[Saving Snapshots](#)

[Using TAP files](#)

[Emulated Machine Type selection](#)

[Video configuration options](#)

[Audio configuration options](#)

[Miscellaneous configuration](#)

[Command-Line arguments](#)

[ToolTypes](#)

[ARexx port](#)

## 1.11 Basic Operation Overview

ASp is designed to be very easy to use. The most common operations are placed in the GUI to be quickly and easily accessible.

Main Emulation Control

On the left hand side of the ASp window are two large buttons, **RUN** and **STOP**. These are the primary control of the actual emulation.

---

When you press RUN, a display will open depicting the current Spectrum display, and the emulated Spectrum will run. Whatever job the Spectrum was last doing will be continued from exactly where it was stopped.

When first run, or directly after a reset, the emulated Spectrum will behave as if you had just applied power to it. If you RUN directly after loading a snapshot file, the information in that file tells the Spectrum where to continue running the loaded program at the point it was saved.

STOP can be pressed to cause the emulation of the Spectrum to pause, closing the Spectrum display. You can resume the emulation by pressing RUN.

When the emulation is running, pressing the Escape key has exactly the same effect as pressing the STOP button on the GUI.

The **Reset** button on the GUI provides a way of quickly clearing whatever is in the Spectrum's memory and re-booting into the normal Spectrum intro screen. You can press Reset when the emulation is running for instant effect, or when the emulation is stopped to force a clean start when you next Run.

**RUN** , **STOP** and **Reset** are also available as items in the "Emulation" menu.

### Snapshots

The next column of buttons provide the ability to **Load** and **Save** the instantaneous state of the emulated Spectrum. Saving to a snapshot file allows you to revert to that exact position at a later date by Loading it back in.

ASp will load snapshot files of type **.Z80** (any version) and **.SNA** (48K type), which are amongst the most popular snapshot formats found.

You can save the state of the emulator at any time into a **.Z80** (version 2) or **.SNA** file for later retrieval.

The **Reload** button is a shortcut which instantly reverts the emulator back to the state it was in directly after the last successful snapshot Load. It is the same as using the Load function to load the same file as last time, with the exception that the file on disk is not re-scanned, the information is retrieved from an internal buffer. This is only of concern if you modify the snapshot file.

**Load** , **Reload** and **Save** are also available as items in the Project menu.

### Tape Images

The right hand column of buttons provide support for loading from **.TAP** files. Unlike snapshots, these are representations of the data on a real Spectrum cassette tape and do not affect the current status of the emulator. To access the tape file you must **Insert** a **.TAP** file and make the Spectrum do something which would cause a real Spectrum to read from a real tape, such as selecting "Tape Loader" from the 128K Spectrum intro menu, or typing LOAD "" at the 48K Spectrum command line.

ASp can currently only read tape files. It cannot write to them. The only supported format is **.TAP**

For convenience, you can also **Rewind** and **Eject** TAP files as if they were real cassette tapes.

Tape control is also available as items in the Emulation->Tape submenu.

### Quitting ASp

ASp can be **quit** by closing its window or selecting the Quit menu item.

## 1.12 The RUN GUI function

Pressing RUN on the GUI or selecting the Emulation->Run menu item will cause the emulation display to open and the emulation will start.

The emulation carries on from wherever it was when it was Stopped.

## 1.13 The STOP GUI function

Pressing STOP on the GUI or selecting the Emulation->Stop menu item will cause the emulation to pause and the emulation display will close. The Escape key will also perform this function.

The emulation will be resumed from the point it Stopped when the Run function is used.

---

## 1.14 The RESET GUI function

Pressing RESET on the GUI or selecting the Emulation->Reset menu item will cause the emulated Spectrum to undergo a power-on Reset.

The program and display data, loaded snapshot, and other current machine status is lost.

## 1.15 The LOAD GUI function

Pressing LOAD on the GUI or selecting the Project->Load menu item will display a file requester for you to select a snapshot file to load.

You can select any SNA or Z80 file to load. The emulation will immediately change its behaviour to conform to the content of the snapshot image. All previous data and status is lost.

If you Cancel the requester or the file failed to successfully load, the original status of the emulation is unaffected.

## 1.16 The RELOAD GUI function

Pressing RELOAD on the GUI or selecting the Project->ReLoad menu item will cause the emulation to immediately revert to the state it was in directly after the last successful snapshot Load.

This is effectively the same as using the "Load" function to load the same file as before, except the information is retrieved from an internal buffer, not by re-scanning the file.

## 1.17 The SAVE GUI function

Pressing SAVE on the GUI or selecting the Project->Save menu item will display a file requester for you to specify the name of a snapshot to save.

The entire status of the emulation will be saved into the file, which can be Loaded at any later time to revert to the exact current status.

By specifying a file extension of ".z80" or ".sna", ASp will save the data in that format. You are strongly advised to use the Z80 format.

## 1.18 The INSERT GUI function

Pressing INSERT on the GUI or selecting the Emulation->Tape->Insert menu item will display a file requester to allow you to select a TAP file for use.

The file, which represents a Spectrum cassette tape, will be Inserted into the virtual "tape deck" attached to the emulated Spectrum.

## 1.19 The EJECT GUI function

Pressing EJECT on the GUI or selecting the Emulation->Tape->Eject menu item will remove the Inserted TAP file from the "tape deck".

This function is for convenience and completeness only. There is no need to Eject any Inserted TAPes before Inserting a new one, nor before a Quit.

---

## 1.20 The REWIND GUI function

Pressing REWIND on the GUI or selecting the Emulation->Tape->Rewind menu item will set the "current position" of the TAPE file in the "tape deck" to the beginning, just as if you were to rewind a real tape in the cassette player.

This function is essentially the same as re-Inserting the same TAP file as is already Inserted.

## 1.21 How to Quit ASp

To exit from ASp, you can select the Project->Quit menu item or simply close the GUI window.

ASp will not Quit if the emulation is currently Running, unless the ARexx "QUIT FORCE" command is issued.

If ARexx scripts have been invoked by ASp using the Project->Run ARexx... menu item which have not yet completed, ASp will report a status of "Waiting..." until all scripts have terminated, after which it will exit.

## 1.22 Keyboard layout information

When ASp is running, the Amiga keyboard behaves like a Spectrum keyboard. There are several things to be aware of about this arrangement.

Letter and number keys work as expected, as do Space and Enter.

Spectrum has two shift keys, Caps-Shift and Symbol-Shift. In ASp, Caps-Shift is activated by pressing either of the normal Shift keys, Symbol-Shift is available by using either of the Alt keys. To access the Spectrum's "extended shift" mode, press Shift and Alt simultaneously, just like on a real Spectrum where you'd press Caps-Shift and Symbol-Shift.

The symbols printed on the Amiga keyboard above the number keys cannot be accessed directly by pressing shift-1,2,3 etc. The Spectrum assigns other functions to shifted number keys, the symbols are available elsewhere on the Spectrum keymap using Symbol-Shift.

Some symbols and other functions usually accessed by two or more keystrokes on a real Spectrum are available as single-key shortcuts from the Amiga keyboard. The comma (,), full stop (.), slash (/), semicolon (;), hash (#), minus (-), and equals (=) are from the corresponding keys near the "return" key. Open and close brackets, "(" and ")", asterisk (\*), and plus (+) are in addition available from the numeric keypad.

Backspace is a shortcut for Caps-Shift-0 (delete) and the cursor keys operate as Caps-Shift-5,6,7,8 for Spectrum cursor action.

Due to certain limitations, the keyboard layout corresponds to a UK Amiga keymap, it is not affected by locale settings. If this really causes people major problems there may be a way round this, albeit restricted.

Some F-Keys perform GUI shortcuts, but can be disabled via [ToolTypes](#) or [command line arguments](#) .

F1 is Load,

F3 is Save,

F6 is ReLoad,

F9 is Reset.

You can use the F-Key shortcuts at any time.

In some instances, usually games, you may need to press several keys simultaneously. Some Amiga keyboard controllers (notably the A1200) do not recognise certain combinations of multiple keypresses. This is a limitation of the keyboard controller, not of ASp.

To circumvent this problem, define your keys such that they do not share the same rows of the keyboard. Try, for example, 3,E,K,L,Space for "Up, Down, Left, Right, Fire" respectively. Alternatively, investigate the use of the [Custom-Keys Joystick](#) mode.

## 1.23 Types of Joystick available in ASp

ASp offers a choice of Spectrum joystick types. Simply select which type of joystick you want fitted to your emulated Spectrum from the Settings->Joystick menu. The normal Amiga joystick will then appear to the Spectrum to be that type of joystick. You can change the joystick type at any time.

Types of joystick available:

**Kempston (normal)** (Default)

**Kempston (smart)**

**Cursor**

**Sinclair 1**

**Sinclair 2**

**Custom Keys**

Also see **Keep Kempston Interface** menu item.

## 1.24 Kempston Joystick operation

Activated by selecting the "Settings->Joystick->Kempston (normal)" menu item.

Kempston is by far the most popular joystick interface type used by the Spectrum. Almost every program will work with this type of stick.

ASp will emulate a straightforward Kempston interface which is read by IN'ing from any port with address lines A5,A6 and A7 logic low.

See also: **Smart Kempston** joystick type.

## 1.25 Smart Kempston Joystick operation

Activated by selecting the "Settings->Joystick->Kempston (smart)" menu item.

Behaves as the **Normal Kempston** joystick type with one additional feature:

Some games (seems to be especially Spanish ones) will use an unspecified joystick type which superficially appears to be Kempston, but the normal Kempston decode will not work with these games. Should such a game be running, identified by accessing ports with just A5 low, ASp will switch from treating this as a "normal" Kempston and instead respond with the Up and Down directions swapped.

It is advised that you use the normal Kempston unless you need this feature.

## 1.26 Cursor Joystick operation

Activated by selecting the "Settings->Joystick->Cursor" menu item.

This makes the Amiga joystick appear to the Spectrum to be a Cursor joystick. A Cursor interface emulated the keys 5,6,7,8,and 0 being pressed, corresponding to the Spectrum's (unshifted) cursor key locations.

## 1.27 Sinclair 1 Joystick operation

Activated by selecting the "Settings->Joystick->Sinclair1" menu item.

The Amiga joystick will appear to behave as Stick 1 (Right) of the Sinclair Interface 2. This emulates keypresses 6,7,8,9,0.

---

## 1.28 Sinclair 2 Joystick operation

Activated by selecting the "Settings->Joystick->Sinclair2" menu item.

The Amiga joystick will appear to behave as Stick 2 (Left) of the Sinclair Interface 2. This emulates keypresses 1,2,3,4,5.

## 1.29 Custom Keys Joystick operation

Activated by selecting the "Settings->Joystick->Custom Keys" menu item.

This setting allows you to define an arbitrary set of five Spectrum keys, corresponding to Up, Down, Left, Right and Fire, which will appear to the Spectrum to be pressed whenever the Amiga joystick is moved.

Therefore, any program which requires control via the Spectrum keyboard can instead be controlled using the joystick. The emulated program is unaware of the substitution, and the keyboard continues to work normally.

When you select this option you will be presented with a small window which will allow you to Accept the current key bindings or Change them. If you choose to change them, click "Change" and then press five keys in turn which correspond to the required program control directions. The window will update to show your new selections. After entering all five keys, you can again Accept or Change the new key-bindings.

When defining the key-bindings, if you press an Amiga key which corresponds to multiple Spectrum keypresses (for example, comma "," is Symbol-shift-n) then the keypress will only be logged as the first of the two keys. (In the example, pressing comma would be logged as Symbol-Shift).

There used to be a form of programmable joystick for the Spectrum which behaved in a manner similar to this, emulating keypresses. Unfortunately there was a requirement to load a short program from tape to configure the joystick before you loaded the program you wanted to run. Luckily, with ASp, you can reconfigure the key bindings at any time. :-)

## 1.30 Loading Snapshots

A snapshot is, as the name suggests, a description of the instantaneous state of the emulated Spectrum. The purpose of such files is to be able to record the entire contents of the emulated Spectrum and what it was doing at the time such that it can be re-loaded at a later date.

You can use this to save your work, a picture, your progress in a game, etc. The most common usage, however, is to exchange ready-to-run images of programs with other users. Please read [this text](#) about the legalities of that action.

To load a snapshot, press the GUI Load button, or select Load from the Project menu.

You will be presented with a file requester for you to select the file you wish to load. By default, only those files with name ending ".sna" or ".z80" are displayed. If you have a valid snapshot file which does not have such a name, modify or clear the "pattern" field of the requester to see it.

Either double-click on the desired file name, or select it and press Load in the requester. ASp will analyse the file to determine its format and attempt to load it. If it is unsuccessful, an error message will be displayed.

In the event of an error (due to a corrupt or invalid snapshot) the previous state of the emulation is unaffected.

ASp will currently load **.Z80** or **.SNA** snapshot files. Other formats may be added in the future.

Any time after a successful Load you can press Reload, or select the Project->Reload menu item to return to the state the emulator was in immediately after loading the last snapshot.

## 1.31 Saving Snapshots

To save a snapshot, press the GUI Save button, or select Save from the Project menu.

---

You can perform a save at any time to store the instantaneous state of the emulation into a disk file for later retrieval or to load into other emulators.

When you select Save, you will be presented with a file requester. Type in the name of the file you wish the information to be saved into. Take care as you can easily overwrite an existing file!. Then press the Save button in the requester.

You can specify any name you want, but you are advised to choose names which have ".z80" or ".sna" appended to them for better identification. If you specify a file extension like this, ASp will save the file in that format without the need to ask. Otherwise ASp will display another requester asking which file format, **.Z80** or **.SNA**, you want to save.

You are highly recommended to always use the **.Z80** format. It generates smaller filesizes than the **.SNA** format and can store information about the 128K modes, the **.SNA** cannot.

If you try to save to a **.SNA** file when the emulator is in 128K mode, a warning requester will appear informing you that the **.SNA** format is not suitable. You can ignore this if you choose, but you are unlikely to be able to successfully retrieve your saved work when you re-load it unless you really know what you are doing!

## 1.32 Using TAPe files

A tape file describes the data stored on a Spectrum cassette tape. ASp uses the **.TAP** format.

To select a Tape file for use, click the Insert GUI button or select Insert from the Emulation->Tape submenu.

When the file requester appears, double-click the **.TAP** file you wish to use, or press Insert in the requester. The selected Tape image is inserted in the emulated tape deck and "rewound" to the start.

The next time the emulation attempts to load from tape the Spectrum will be fooled into thinking it is reading information from a real tape. You can Rewind the tape to the start, Eject it, or Insert a new one whenever you want.

Should ASp find that the emulation is attempting to read from tape when you have not yet Inserted a TAP file, or the TAP file has run to the end, you will see the Insert Tape file requester displayed automatically, as if you had pressed Insert.

There are some limitations to the way ASp handles TAP files:

ASp can only read, not write to TAP files.

Only programs which use the Spectrum's ROM tape loading routines can load from TAP files. Some TAP files are available which use custom loader routines in the programs themselves. These will not work with ASp.

## 1.33 Description of .Z80 files

The ".Z80" snapshot file format was devised by Gerton Lunter for use with his emulator for PC's called, funnily enough, "Z80"...

It is a comprehensive format which contains information about all the features of the 48K and 128K Spectrums and additional hardware. File size is reduced significantly in many cases by using a simple run-length compression algorithm.

These files are extremely popular for snapshots on all platforms, and it is recommended that ASp users try to use this format for any saves made. This is the only format supported by ASp that can fully describe the 128K modes correctly.

There are three basic versions of **.Z80** files, some of which have optional selective compression. ASp should load any of them.

ASp saves version 2 **.Z80** files. Most emulators with limited support for **.Z80** files will deal with this version. ASp saves them with optional selective compression enabled.

## 1.34 Description of .SNA files

The ".SNA" (also ".snapshot" or ".Mirage") file is a simple description of the state of the Spectrum and a full RAM dump based on the Mirage Microdrive save format.

---

SNA files are always 49179 bytes in length and contain only enough information to describe the 48K Spectrum status. This makes them unsuitable for use as a means of snapshotting the 128K Spectrum. There is an extended SNA128 format which does contain additional information, but is rare and ASp does not support it.

On the Amiga, it is popular to find SNA files which have been compressed using PowerPacker. ASp does not support direct loading of compressed files, although it is compatible with IO wedges that patch the DOS Read() command, or you could use a transparent disk packer, like XFH.

### 1.35 Description of .TAP files

There are two types of .TAP file for the Spectrum. The one ASp supports is the more popular type used by the "Z80" emulator by Gerton Lunter. The other type is for an emulator called "Warajevo" by Samir Ribic. ASp does not support those.

A TAP file is a literal translation of the data on a Spectrum cassette tape, in the order it appears on the tape. All of the header and checksumming information used by the Spectrum is also there.

The Spectrum normally loads information from tape using a routine in its ROM. ASp hooks into that routine to override the normal emulation to force it to read the TAP file. This has the consequence that the TAP file data is only available to those programs loading through the Spectrum's ROM loading routines.

There are many programs which use their own custom tape loading routines and indeed their own private tape data formats. These tapes are normally saved for emulator use in another tape format called TZX, which ASp does not yet support. Some TAP files, however, contain data which is designed to be loaded via such a custom routine and so therefore will not load into ASp.

### 1.36 Settings->Misc->Add AY chip to 48K model

This menu item permits a slight "cheat" in terms of available hardware.

The AY sound chip was only found in 128K Spectrum models. However, several programs written for 48K machines were coded in such a way as to utilise this chip if it were present (extra sound, better title music, etc.), in case it was loaded into a 128K Spectrum.

ASp can, by activating this menu item, make the AY chip emulation respond even when in 48K mode. This enables the user to, for example, load a (48K-only) SNA file which is capable of playing 128K music.

In order for the user to hear the 48K-only music in a program which has either 128K or 48K music, the user must ensure this setting is inactive and that the emulation is running in 48K mode.

### 1.37 Settings->Misc->Keep Kempston Interface

This setting is a compatibility control. It is active by default.

Some snapshots have been saved at a point where the program has already decided that a **Kempston Joystick** interface is fitted. If you have selected a different joystick type for use or are using keys, so the Kempston Interface is not present, some programs can behave oddly.

This option causes a Kempston Joystick Interface to be plugged in at all times, regardless of the current joystick selection. Of course, if the current selected joystick type is not Kempston, the interface will not respond to joystick movement.

### 1.38 Settings->Misc->Higher Accuracy Timing

This menu option causes ASp to consider more detail about the timing of the emulation. This may be helpful to achieve accurate rainbow graphic effects, but there is a degree of overhead in its use. If you experience difficulties with running ASp at full speed, try disabling this option.

---

When this option is enabled, ASp considers the effect that writes to **contended RAM** locations have on the time taken for a given CPU instruction. Although the calculation is not exact it is quite accurate, and has shown to be more than adequate for typical graphics effects synchronisation.

This will only cause ASp to consider writes to contended RAM. Reads are also subject to timing issues in a real Spectrum, so the result of using this option will not be perfect timing. Several programs' rainbow effects will not appear correctly because of this, although this option almost always at least improves the situation.

### 1.39 Settings->Misc->Delay during TAP load

This setting is a simple user-preference.

When loading from a tape (TAP file), ASp normally "flash-loads" each header and data section. If, for example, a loading picture was displayed you will often barely get time to see it.

Activating this option will cause ASp to insert a brief delay after loading each header or data block.

### 1.40 Audio Save option

Menu item "Emulation->Tape->Audio Save"

This is a simple addition which sources the 48K beeper sound from the information sent to the MIC socket of the Spectrum/tape-deck as opposed to the normal EAR socket.

The result is that most program's attempts to SAVE to tape will result in an audible tape noise in ASp. In principle, this can be used to route into a real Spectrum for Loading. The normal 48K audio noises will often no longer work in this mode, so you should not activate this option unless you need to.

Note that it is vital that ASp is running at full speed without interruption of any kind to ensure the sound is clean and stable enough to be acceptable to the destination Spectrum. In marginal cases it may be helpful to enable maximum **FrameSkip** and disable **Higher Accuracy Timing** in an attempt to squeeze extra speed from the emulation. It is, however, advised that you keep to a higher **sample rate** and leave the **Audio Filter** ON to ensure clean enough sound.

This facility has not been fully tested. Users' reports of the degree of success in the use of this option would be appreciated.

### 1.41 Emulated Machine Type Selection

The Settings->Machine sub-menu provides the user with the ability to switch between the different supported Spectrum models.

Simply use the menu to select which of the available models you want ASp to emulate. Currently available are:

Sinclair 48K: The standard 48K Spectrum, Issue 3.

Sinclair 128K: The first 128K Spectrum, with AY sound chip.

Amstrad 128K+2: The revised 128K Spectrum released shortly after

Amstrad acquired Sinclair's copyrights.

In many situations, changing the model type in this menu will cause the emulated machine to "crash", so the standard procedure is for ASp to perform an automatic power-on **Reset** when changing the selection. The Reset on machine change option controls this behaviour. Disabling this option will allow you to change machine type without affecting the running program. Only certain programs at certain points will safely accept a "live" machine change.

When you Load a snapshot, ASp will automatically select either 48K or 128K mode dependent on the information contained in the snapshot file. This completely overrides any previous user selection.

---

## 1.42 Audio Configuration

There are several options to configure ASp's audio output to your preferences.

**Channel Mode selection**

**Sample Rate selection**

**Quiet 48K beeps**

**Filter Off**

These options affect the eventual audio output, not the behaviour of the underlying emulation. All these options can be changed "on the fly".

## 1.43 Audio Channel Mode

ASp allocates the Amiga sound hardware using audio.device. Of the available sound channels (normally 4) ASp requires either 1 or 2. If ASp is unable to allocate the audio hardware it needs, it will fail to Run the emulation and will display an error requester.

The channel mode is set from the Settings->Audio submenu.

In "Mono" mode, ASp tries to obtain any one available audio channel. All of the emulated sounds, 48K and AY chip, are mixed into one sound and played through this one channel. Mono mode is slightly less CPU load than Stereo modes.

In Stereo mode, ASp tries to obtain a stereo pair (a left and a right) of audio channels. The 48K sounds are mixed in as "stereo centre", whilst the AY chip sounds are positioned according to the AY sound channel used:

"Stereo ABC" plays AY channel A left, B centre and C right.

"Stereo ACB" plays AY channel A left, C centre and B right.

None of the standard issue Spectrum 128K models had stereo sound output, but some of the "clones" did, so a lot of AY-chip music software positions the sounds in a stereo sound stage. The most typical arrangement is the "Stereo ACB", which is the default.

## 1.44 Audio Sample Rate selection

There is a range of three sample rates available for the ASp audio output, selectable from the Settings->Audio submenu.

The sample rate affects the quality of the sound. Higher sample rates produce better quality, cleaner sound. Lower sample rates take less CPU time to compute, helping speed on slower Amiga configurations.

Many Spectrum games titles modulate the 48K internal speaker with high frequencies to improve the quality of the sound output. These high frequencies will interact with ASp's sample rate to produce unpleasant Aliasing Distortion. This is particularly noticeable with lower sample rates.

Removing this distortion requires extra processing time, but this may become an option in a future version.

## 1.45 Settings->Audio->Quiet 48K Beeps

A real 128K Spectrum mixed the 48K sound with the 128K sound such that the 48K sound was as loud as all AY sound channels put together. By default, ASp follows this behaviour.

Activating the Quiet 48K beeps option will cause ASp to mix the 48K sound only as loud as one AY channel, effectively increasing the volume of the AY chip sounds and decreasing the volume of the 48K sounds.

## 1.46 Filter Control

Normally, ASp ignores the Amiga's built-in 5kHz audio filter. If you had it enabled, ASp sound will be filtered, if not, ASp sound will not be filtered.

Activating ASp's "Filter Off" option causes ASp to try to switch off the filter every time you Run the emulation. ASp will return the filter to its previous state when it Stops.

Forcing the filter off gives more high-frequency content to ASp's sound, but can sound harsh and any Aliasing Distortion is more readily audible.

It is advised that the Amiga audio filter be On if you try to use the [Audio Save](#) facility.

## 1.47 Video configuration options

These menu functions allow you to configure some aspects of ASp's video display.

[Screen Mode selection](#)

[FrameSkip setting](#)

These options can be changed on-the-fly.

## 1.48 Screen Mode selection

Settings->Display->Screen Mode...

This function will display a standard ASL Screenmode requester. You can select any of the displayed modes for use with ASp. The available listed modes are dependent on your system configuration.

When ASp next opens the emulation display, it will try to open a screen of this mode. If for any reason the screen is unable to open, ASp will display an error.

Regardless of mode chosen, the opened screen will be the same pixel dimensions. This means that if you pick a high-resolution screen mode you will get a very small emulation display. ASp will not scale the image to fit the screen size.

Nominally, ASp requires a display of around 320 pixels wide with a 1:1 aspect ratio (square pixels).

The default mode is obtained by a call to BestModeID() for a full-screen display.

## 1.49 Frame Skip function

ASp has to run the Spectrum's Z80 CPU emulation through every instruction, but even though a real Spectrum would update the video display every frame, there is no reason why an emulation has to do that.

The FrameSkip setting allows the entire beam-synchronisation (rainbow fx) calculations and display rendering update to be skipped for some frames. Since display-related computation can be as much as 50% of the CPU load, this can allow considerable extra processing power to be available for the Z80 CPU emulation, causing it to run faster. The disadvantage of using this function is the display will only be updated periodically, meaning smooth scrolling effects are no longer smooth.

The FrameSkip figure is the number of 50Hz Spectrum video frames which will occur without any updates. Hence, "FrameSkip 0" means NO frames will be skipped, all frames will be rendered. "FrameSkip 1" means one frame will be allowed to pass without rendering, meaning every other frame is rendered, etc.

The adaptive option, which is ON by default, will cause this behaviour to change slightly.

The selected FrameSkip figure becomes the maximum permitted FrameSkip, but ASp analyses how fast the emulation is running and will try to use a lower FrameSkip figure if it feels able to. In other words, you should obtain the best display update rate possible at any one time, subject to a worst-case limit set by your FrameSkip selection.

In most cases this is a more desirable behaviour, but with some software this can result in an inconsistent and distracting update frequency. Therefore the facility is provided as an option which is disabled by deselecting the Settings->Display->Adaptive menu item, causing ASp to use the fixed FrameSkip behaviour described above.

## 1.50 Miscellaneous configuration options

The following settings relate to miscellaneous changes to the behaviour of the emulation.

Add AY chip to 48K model

Keep Kempston Interface

Higher Accuracy Timing

Delay During TAP Load

Audio Save facility

## 1.51 ASp ToolTypes

A number of icon ToolTypes may be used with ASp. For the most part these are identical in name and usage to the Command-Line Arguments.

Note the behaviour of ASp on startup is to scan the icon file for ToolTypes, even if started from the CLI Shell. The CLI arguments are then able to override the tooltype settings.

Add any of the following ToolTypes to ASp's icon to use the feature:

AREXXPORTNAME

BLANKPOINTER

NOFKEYS

LOADPATH

ROMPATH

ROMNAME48

ROMNAME128

ROMNAME128PLUS2

SAVEPATH

## 1.52 ASp Command-Line arguments

On startup, ASp finds and scans its icon, if available, for tooltype settings, regardless of whether it was started from CLI or Workbench. If it was started from the CLI, you can offer arguments to override the tooltype settings.

The CLI arguments are mostly identical in name and usage to the tooltypes. The argument template is available by typing "ASp ?" in a CLI Shell, and currently has the following options:

AREXXPORTNAME/K

LOADPATH/K

SAVEPATH/K

ROMPATH/K

ROMNAME48/K

ROMNAME128/K

ROMNAME128PLUS2/K

BLANKPOINTER/S

FKEYS/S

NOBLANKPOINTER/S

NOFKEYS/S

### 1.53 AREXEXPORTNAME Argument

This is available as a tooltype or CLI argument.

Format: AREXEXPORTNAME=<name of arexx port>

The name of the ARExx port can be specified as any valid port name up to 32 characters in length. It must be a unique name otherwise ASp will display an error.

### 1.54 BLANKPOINTER Argument

This is available as a tooltype or CLI argument.

Format: BLANKPOINTER

The presence of this tooltype or argument will cause ASp to make the mouse pointer invisible when the emulation display is clicked on or when first opening the display.

When supplied as an icon tooltype, this can be overridden by the **NOBLANKPOINTER** CLI argument.

### 1.55 NOFKEYS Argument

This is available as a tooltype or CLI argument.

Format: NOFKEYS

The presence of this tooltype or arguments will disable the shortcuts available from the Function Keys. ASp will ignore any F-Key activity.

When supplied as an icon tooltype, this can be overridden by the **FKEYS** CLI argument.

### 1.56 LOADPATH Argument

This is available as a tooltype or CLI argument.

Format: LOADPATH=<initial path for requester>

This tooltype or argument allows specification of which directory is initially displayed when the user first opens the Load or Insert file requesters.

If this argument is not present, it defaults to "PROGDIR:".

### 1.57 SAVEPATH Argument

This is available as a tooltype or CLI argument.

Format: SAVEPATH=<initial path for requester>

This tooltype or argument allows specification of which directory is initially displayed when the user first opens the Save file requester.

If this argument is not present, it defaults to "RAM:".

---

## 1.58 ROMPATH Argument

This is available as a tootype or CLI argument.

Format: ROMPATH=<directory containing ROM files>

This tootype or argument allows specification of a directory in which the Spectrum ROM image files can be found.

If not specified, this defaults to "PROGDIR:"

Read the section on [which ROMs to use](#) for more information.

## 1.59 ROMNAME48 Argument

This is available as a tootype or CLI argument.

Format: ROMNAME48=<filename of ROM image>

This tootype or argument allows specification of the filename of the Spectrum 48K ROM image file. The file is looked for in the directory optionally specified by the [ROMPATH](#) argument, but if you provide a fully-qualified path and file name here, this will override it for this file.

Read the section on [which ROMs to use](#) for more information.

## 1.60 ROMNAME128 Argument

This is available as a tootype or CLI argument.

Format: ROMNAME128=<filename of ROM image>

This tootype or argument allows specification of the filename of the Spectrum 128K ROM image file. The file is looked for in the directory optionally specified by the [ROMPATH](#) argument, but if you provide a fully-qualified path and file name here, this will override it for this file.

Read the section on [which ROMs to use](#) for more information.

## 1.61 ROMNAME128PLUS2 Argument

This is available as a tootype or CLI argument.

Format: ROMNAME128PLUS2=<filename of ROM image>

This tootype or argument allows specification of the filename of the Spectrum 128K+2 ROM image file. The file is looked for in the directory optionally specified by the [ROMPATH](#) argument, but if you provide a fully-qualified path and file name here, this will override it for this file.

Read the section on [which ROMs to use](#) for more information.

## 1.62 FKEYS Argument

This is a Command Line argument only.

Format: "FKEYS"

The presence of this argument will force ASp to enable the Function Key shortcuts, even if the [NOFKEYS](#) tootype is present.

By default, the F-key functionality is active, so there is no need to specify this argument unless you need to override the tootype.

---

## 1.63 NOBLANKPOINTER Argument

This is a Command Line argument only.

Format: "NOBLANKPOINTER"

The presence of this argument will force ASp to disable the pointer-blanking facility, even if the **BLANKPOINTER** tooltype is present.

By default, the pointer-blanking functionality is disabled, so there is no need to specify this argument unless you need to override the tooltype.

## 1.64 Copyrights on Spectrum Software

It is a popular misconception that exchanging or downloading images of commercial software for use with emulators does not constitute software piracy.

Unfortunately, it IS piracy and IS illegal!

All is not lost, however! A few software companies and authors have made public statements that specific titles for which they own the copyrights may be made available for use. In addition, the maintainers of **World Of Spectrum** website have taken a lot of time and effort to contact copyright holders to try to encourage them to allow free distribution. They have had a lot of positive responses, with many individuals giving them permission to retain copies for public download. In all cases, however, the original author retains copyright, the software is simply being made free to the public.

Please respect the generous gestures made by those individuals by not abusing the privilege of being able to use their software for free. Do not claim it as your own, do not distribute hacked or broken copies as originals. Always acknowledge their copyrights.

Please Note: The author of ASp is in no way responsible for the user's choice of program images used in conjunction with ASp. The use of illegal images of Spectrum software is entirely at the user's risk and discretion.

The World Of Spectrum website contains links to detailed information concerning the many viewpoints and legalities with this issue.

## 1.65 ARexx port functions

ASp creates an ARexx port for communication with other programs. By default, the port will be named "ASP\_0" or, if that name is unavailable, "ASP\_1", "ASP\_2" and so on until it finds a unique name. Alternatively, the user can specify the name of the ARexx port to use by the **AREXXPORTNAME** tooltype or CLI argument.

ASp makes two uses of the ARexx port. **Executing commands** and **running scripts** .

## 1.66 Running an ARexx script from within ASp

To run an ARexx script, select the Project->Run ARexx... menu item. Select the name of the script you wish to run in the requester.

The ARexx program which is run from this function will address the ARexx port of the ASp copy it was called from by default, without having to determine its name.

You can use this feature as a kind of macro functionality, or to expand the number of features or functions ASp has.

Some demo scripts are supplied in the Rexx subdirectory. These are just very simple examples of what you could do.

**LoadSCR.asprx**

**SaveSCR.asprx**

**POKEr.asprx**

---

## 1.67 ASp ARexx Commands

ASp supports a selection of ARexx commands to enable macro functionality, or allow other programs and external scripts to control its features.

**Important general information** regarding ASp's command set.

Commands are likely to get added with subsequent versions, therefore the following list shows minimum version numbers of ASp which support the command. Commands new or modified in this version have the version number in bold.

V0.76 **HELP**

V0.76 **GETVERSION**

V0.75 **RUN**

V0.75 **STOP**

V0.75 **RESET**

V0.75 **QUIT**

V0.75 **LOAD**

V0.76 **SAVE**

V0.75 **RELOAD**

V0.76 **INSERT**

V0.76 **REWIND**

V0.76 **PEEK**

V0.75 **POKE**

V0.76 **READRAM**

V0.76 **WRITERAM**

V0.76 **GETPAGE**

V0.75 **SETPAGE**

## 1.68 ASp ARexx important information

ASp is an ARexx Command Host. The commands for which you expect a result will put the result in the "RESULT" ARexx special variable if, and only if, the ARexx "OPTIONS RESULTS" feature is switched on. You cannot treat the commands as functions. ie:

```
V=PEEK 16384
```

```
SAY 'Location 16384 contains 'V
```

```
is INCORRECT!
```

```
Whereas,
```

```
OPTIONS RESULTS
```

```
PEEK 16384
```

```
SAY 'Location 16384 contains 'RESULT
```

```
is correct.
```

```
*****
```

It is important to realise that the ARexx commands are Asynchronous to the GUI. ie: If you STOP the emulation, the user could hit the RUN button on the GUI to start it again before you realise. Similarly, commands like POKE or PEEK will still execute while the emulation is running and therefore their meaning or usefulness may be limited if you do not first STOP the emulation.

It is up to the ARexx programmer to ensure his code is aware of the current state of the emulation when required and executes suitable interlock procedures to ensure that data passed to and from the emulation is meaningful.

## 1.69 ARexx command: HELP

Syntax: HELP

Arguments: None.

Result: Command summary.

Return code: OK

Function: Returns a short command summary text.

Unlike most other commands, this command will write the result to the command source's STDOUT if the ARexx OPTIONS RESULTS feature is not active.

## 1.70 ARexx command: GETVERSION

Syntax: GETVERSION

Arguments: None.

Result: Version number.

Return code: OK

Function: Returns a 4-digit decimal number which is the ASp executable version number multiplied by 1000. ie: V0.76 will return "0760".

In-between versions will increment the last digit.

Should be used to check the version of ASp is adequate to run the script, eg:

Options Results

GETVERSION

If RESULT>0760 Then Do

...

## 1.71 ARexx command: RUN

Syntax: RUN

Arguments: None.

Result: None.

Return code: OK if emulation started ok.

WARN if emulation was already running.

FAIL if emulation could not be started.

Function: Attempts to open the emulation display and start the emulation as if the user had pressed the RUN button on the GUI.

## 1.72 ARexx command: STOP

Syntax: STOP

Arguments: None.

Result: None.

Return code: OK if emulation stopped ok.

WARN if emulation was not running.

Function: Stops the emulation and shuts down the emulation display, as if the user had pressed the STOP button on the GUI.

## 1.73 ARexx command: RESET

Syntax: RESET

Arguments: None.

Result: None.

Return code: OK

Function: Resets the emulation as if the user had pressed the RESET button on the GUI.

## 1.74 ARexx command: QUIT

Syntax: QUIT [FORCE]

Arguments: FORCE: Shut down emulation and don't ask user. (V0.76+)

Result: None.

Return code: OK if the Quit Requester was successfully displayed.

FAIL if the args were invalid, or:

if no "FORCE" argument, the emulation was running.

Function: Acts as if the user had selected the Project->Quit menu item.

Displays the "Quit? Yes/No" requester, which the user may cancel.

If the emulation was running, this command fails.

New for V0.76:

Use of FORCE argument will always cause shutdown even if running, and will not ask the user.

## 1.75 ARexx command: LOAD

Syntax: LOAD [<filename>]

Arguments: <filename>: Optional snapshot filename.

Result: None.

Return code: OK if file loaded ok.

ERROR if user cancelled filerequester.

FAIL if could not load file for any reason.

Function: Attempts to Load the snapshot file specified by <filename>. If no file was specified, displays the Load Snapshot requester.

## 1.76 ARexx command: SAVE

Syntax: SAVE [<filename>]

Arguments: <filename>: Optional snapshot filename.

Result: None.

Return code: OK

Function: Saves the state of the emulation to the specified file.

The format used is determined by the file extension. If ambiguous, the Z80 format is used.

If <filename> is not supplied, displays the Save Snapshot requester.

## 1.77 ARexx command: RELOAD

Syntax: RELOAD

Arguments: None.

Result: None.

Return code: OK if last file reloaded ok.

ERROR if no ReLoad buffer available. (ie: Previous load failed.)

Function: Recovers the state of the emulation after the last successful Load.

## 1.78 ARexx command: INSERT

Syntax: INSERT [<filename>]

Arguments: <filename>: Optional TAP file name.

Result: None.

Return code: OK if file inserted ok.

FAIL if unable to read file.

Function: Inserts the TAP file specified into the virtual tape deck. If no file specified, the Insert TAP requester is displayed.

---

## 1.79 ARexx command: REWIND

Syntax: REWIND

Arguments: None.

Result: None.

Return code: OK if tape rewound ok.

WARN if no tape currently inserted.

Function: Rewinds the currently selected TAP file to the beginning, as if the user had pressed the REWIND GUI button.

## 1.80 ARexx command: POKE

Syntax: POKE <address> <value>

Arguments: <address>: The address (0-65535) to write to.

<value>: The value (0-255) to write to that address.

Result: None.

Return code: OK if value successfully written.

FAIL if <address> or <value> out of range.

Function: Writes the byte <value> to memory location <address>.

ROM locations cannot be changed by this command.

## 1.81 ARexx command: PEEK

Syntax: PEEK <address>

Arguments: <address>: The address (0-65535) to read from.

Result: Decimal value (0-255) of the contents of the memory location.

Return code: OK if value successfully read.

FAIL if <address> out of range.

Function: Reads a byte value (0-255) from memory location <address>.

## 1.82 ARexx command: READRAM

Syntax: READRAM <address> <length>

Arguments: <address>: Address (0-65535) of first location to read from.

<length>: Number of bytes to read. (1-16384)

Result: A data stream (string) containing byte values of each sequential location from <address> for <length> bytes.

Return code: OK if data read ok.

FAIL if <address> or <length> out of range.

Function: Reads a string of <length> bytes from <address> memory location onwards. <address> + <length> must not total more than 65536.

ie: Address wrap-around not supported.

---

### 1.83 ARexx command: WRITERAM

Syntax: WRITERAM <address> <data>

Arguments: <address>: The first address (0-65535) to write to.

<data>: A string of byte values to write to the location.

Result: None.

Return code: OK if values written ok.

FAIL if <address> out of range.

Function: Writes each byte value from <data> sequentially into memory locations starting at <address>. It is not possible to overwrite ROM with this command.

Address wrap-around is supported.

### 1.84 ARexx command: SETPAGE

Syntax: SETPAGE <value>

Arguments: <value>: Value (0-255) to send to port \$7FFD.

Result: None.

Return code: OK if output ok.

FAIL if <value> out of range.

Function: Allows control of 128K paging features by outputting <value> to port \$7FFD. The bits of this port have the following functions:

Bits: 0-2: RAM page number to map at RAM \$C000-\$FFFF.

3: 0=Normal screen, 1=Shadow screen.

4: ROM select: 0=128K Basic, 1=48K System.

5: If set, further output will be ignored until Reset.

6-7: Unused. Should always be set to 0.

The running emulation will honour Bit 5, but this command is effective regardless of the state of Bit 5.

### 1.85 ARexx command: GETPAGE

Syntax: GETPAGE

Arguments: None:

Result: Value last output to port \$7FFD

Return code: OK

Function: Gets the value last output to port \$7FFD which controls 128K paging features. The bits of this port have the following meaning:

Bits: 0-2: RAM page number to map at RAM \$C000-\$FFFF.

3: 0=Normal screen, 1=Shadow screen.

4: ROM select: 0=128K Basic, 1=48K System.

5: If set, further output will be ignored until Reset.

6-7: Unused.

## 1.86 Demo script: LoadSCR.asprx

This is a very simple script which takes a .SCR file and writes the data into the display RAM region of the emulated Spectrum. As soon as you RUN the emulation you will see the Loaded .SCR image, but it will be lost as soon as the running program writes to the screen. ie: You should only use this when the Spectrum display will remain static.

## 1.87 Demo script: SaveSCR.asprx

This demo script reads the RAM contents of the current Spectrum display and saves the data to a .SCR file selected by the user. Since a .SCR file is a simple RAM dump of the display region, this is a very simple procedure!

## 1.88 Demo script: POKEr.asprx

This demo script allows you to enter POKE values at any time.

Running the script will open a small window in which you can type a comma-separated address and byte value to POKE into RAM. Enter a blank line to Quit.

This is a simple basis of a script to accept multiple POKEs as game cheats etc.

## 1.89 Copyrights and Distribution

ASp is programmed by and copyright Ian Greenway.

The Spectrum ROM images are copyright Amstrad Plc.

MuLib and MuTools are copyright Thor Software.

All other copyrights acknowledged.

**Important information** concerning the copyright status of software relating to Spectrum emulators.

The archive may be freely distributed in unmodified form. No files may be added, removed or modified, with one exception: The Spectrum ROM files may be removed, but may NOT be replaced with modified or alternative files.

ASp is Freeware. It may not be distributed in a form which requires payment beyond typical media costs, nor can it be used to promote commercial software. I don't get any money from it, I don't see why anyone else should!

Disclaimer:

All parts of the ASp package are provided without warranty of any kind. The entire responsibility for use of this package is with the user. No responsibility is accepted by the author or any other individual or group for any consequence of the use or performance of the package. No representation is made for fitness for purpose, reliability, accuracy or any other aspect of performance. If this is in contravention of any local laws, or is otherwise unacceptable, do not use this package.

In other words, ASp is provided in the spirit of "Copyrighted Freeware". All features are provided in good faith, but at the end of the day it is free and you get what you pay for. If it makes your harddrive blow up or starts World War Three, don't go looking at me! :)

## 1.90 Credits

A lot of people have been very helpful during the creation of this program.

Thanks and acknowledgments to them all (strictly alphabetical):

---

Stephen "FBlit" Brookes

Mac Buster

Lee Cook

David Cripps

Martijn "void" van der Heide

BKJ/Vulture

Luca "Hexaae" Longone

Thomas "Thor" Richter

Mario Sarno

Gerard Sweeney

Everyone on the Spectrum Emulator mailing list.

..and a load of other ppl who I've forgotten to mention.. ;-)

Information from:

A Z80 Workshop Manual - E.A. Parr

MC68030 User's Manual - Motorola

The "TechInfo.doc" from the "Z80" emulator - Gerton Lunter.

The Spectrum Emulator FAQ - Phil Kendall.

Various anonymous documents on the internet.

And my trusty rubber-keyed Spectrum 48K, of course!

---

ASp NewIcon image by Lee Cook.

The excellent mmu.library and MuTools by Thomas Richter.

---

ASp has been written from scratch in 680x0 asm using HiSoft's Devpac 3.18.

Some of the data structures have been created in TurboCalc2 and processed using programs I've compiled with VBCC.

Total size of source and data files required to build ASp: Appx 1Mb.

## 1.91 Version History

Most recent changes first.

V0.76: Add: Rewrite display rendering to support RTG.

Add: Beam synchronised graphics update.

Add: 128K+2 mode.

Add: CLI Args.

Add: More ARexx commands, including those which return values.

Add: Ability to run ARexx scripts.

Add: Definable ARexx port name.

Add: Higher accuracy timing.

Add: "Smart Kempston" stick option.

---

Add: More flexible ROM loader.

Add: Audio Filter control.

Add: ZipWindow gadget.

Add: Very simple Audio-Save.

Fix: NOFKEYS option now OFF when no icon.

Fix: Minor bug in Load routines.

Fix: Sinclair 2 Joystick now correct.

Fix: Loading without STOP/RUN now uses correct CPU timing.

Fix: Z80 Load/Save routines with uncompressed data.

Fix: Memory leak in GUI code.

Fix: Window refresh was broken.

Fix: EI no longer allows INT directly afterwards.

Chg: Make ARexx port name visible.

Chg: Rearrange emu control to avoid potential (rare) load problem.

Chg: 48K mode now uses "real" 48K ROM, not 128K ROM#1

Chg: Small mods to GUI font-sensitive scaling code.

V0.75b: Fixed MCP Enforcer hits with BLANKPOINTER. (Luca Longone)

Fixed lockouts in the GUI.

Fixed menu command keys for Rewind, Reload and Run.

Fixed T-times for single-shifted opcodes which broke in V0.75. :(

Fixed potential to lose an audio event after the end of a frame.

Added "Keep Kempston interface" menu item.

Move Reset from F10 to F9. (Lee Cook)

Add "NOFKEYS" tooltip.

Add a LendMenus() to make it easier to access the menus when dragging down the system screen.

Add correct T usage for successive DD or FD opcodes.

Add interrupt is only valid for 32T.

V0.75: Change Z80 core to obtain 15-20% speed increase on 68030.

Added simple ARexx port.

Add more joystick types.

Add user-defined "joy-keys" window.

Change Kempston decode for compatibility. (Artur Chlebek)

Add BLANKPOINTER tooltip.

Fix multiple menu selections. (Luca Longone)

Added variable audio sample rates.

Added timing changes for 128K/48K modes.

Add Quiet 48K option, default changed to Loud.

Fix block IN and OUT flags and timings. Fixes Gryzor music.

Corrected timings for ZX interrupts.

---

---

Fixed loading of Z80 files of 49179 bytes.  
Optimised AY sound build again.  
Changed version string to correct format. (Luca Longone)  
V0.74: Much faster 128K bank switching.  
Add support for V42 mmu.library.  
Re-introduce 68030 speedups based on ASp V0.71b.  
Z80 loader will load the initial part of a SLT.  
F2 (ReLoad) changed to F6. (Grr!)  
Better error report on init failure.  
V0.73dβ: Fix: Menus again! Forgot a MENUTOGGLE flag.  
V0.73cβ: Internal development version.  
V0.73bβ: Fix: Fixed NewLook menus rendering as black-on-black.  
Fix: If AGA wasn't available, the OCS display was broken!  
V0.73B: Fix: 128K paging.  
Fix: Shadow screen swapping.  
Fix: Glaring major bug in TAP loader.  
Fix: Z80 loader bugs.  
Add: Z80 saver.  
Add: Auto-detect and option for SNA or Z80 save.  
Add: 48K-128K mode switch.  
Add: Menus.  
Add: Stereo sound.  
Add: About function.  
Enhance GUI appearance.  
Speed up sound building.  
V0.72B: Prerelease 128K version.  
Add: 128K paging.  
Add: 128K file loading.  
Add: Lots of bugs. ;^)  
V0.71b: Experimental speedup. 5-15% depending.  
V0.71: Fix: Rewrote sound handling. No longer uses audio.device calls.  
Added ToolTypes: LOADPATH, SAVEPATH, ROMFILE.  
Added Reload and Rewind functions.  
Key shortcuts for Load, Save, Reload, Reset.  
System screen now \*activates\* input!  
Fixed "." on numeric pad.  
Test for AGA chipset.  
Filerequesters are a bit nicer.  
Inactive GUI buttons are ghosted.

---

Some potential deadlocks removed.

V0.70: Speeded up sound routines.

Now all audio goes through audio.device! 100% system friendly.

Allows running multiple copies at once. :)

Add emulation of INI/OUTI/IND/OUTD.

Generalised i/o port decoding.

V0.69: Internal Test Version. Added AY-3-8912 sound emulation.

V0.68: Beta Release only. Limited ".TAP" loading support.

V0.67: Now has choice of Custom AGA Display or Native System Screen.

GUI: Added STOP button and Display-Type cycle gadget.

Can now Load/Save/Reset even if emulator is running.

Now requires WB3.0 (V39) or above.

Lee Cook contributed an alternative NewIcon. Thanks!

V0.65-V0.66: Internal.

V0.64: Interim bugfix release:

Fixed a nasty enforcer hit in the .SNA loader! Oops! :-)

Added facility to load ".Z80" files, versions 1, 2 & 3, 48K type only.

V0.63: Added some error requestors.

Fiddled around with parts of the source to aid future updates.

V0.62B: Made emulation code more regular at minor cost in speed on '030.

First public release.

## 1.92 Frequently Asked Questions

Here are some issues that could or do get raised in relation to ASp..

Q: I get a "tear" or glitch in the moving graphics. I thought this emulation was supposed to be beam-synched?

A: It is! :-) However, the display update does not happen synchronous to the Amiga video beam. What you are seeing is the update rendering happening as the video beam passes that point.

Q: The screen update seems jerky. I know this software on a real Spectrum scrolls the display perfectly smoothly. What gives?

A1: You are using an Amiga ScreenMode with a vertical refresh other than 50Hz.

The Spectrum only knows about 50Hz frame speeds and so only updates its display in time with that. The interaction between the two frame rates causes some frames to be visible once, others twice, some not at all. This results in a jerky display.

A2: If you have insufficient CPU power available to ASp, or you are running other programs which periodically steal the CPU at the same time, ASp cannot

maintain full Spectrum speed. Since it must finish emulating one frame before starting on the next, the frame update rate is inconsistent and slow.

Q: The rainbow effects in program XYZ don't look like they should.

A1: Try switching on "Settings->Misc->Higher Accuracy Timing" option.

A2: Make sure the program was designed to run on the machine being emulated. The timings of 48K and 128K machines are different, and may be different to other clones, like Scorpion or Pentagon.

A3: ASp's rainbow effects are approximated. Taking account of every possible timing issue and calculating every display pixel based on it is not practical to do without using huge amounts of CPU time. The system used in ASp tries to work in most cases with minimal additional CPU load.

Q: I have a TAP file here which doesn't work with ASp. It's fine with other emulators.

A: ASp's TAP loader is limited. Only programs which load through the ROM routines will be able to use TAP files. Strictly, TAP files should not be used in other cases, but several are around. :(

Q: I have a game which is supposed to work with Kempston Joystick but it just goes nuts when I run it!

A1: Try the Smart Kempston stick. It may help in some cases.

A2: Some games have an as-yet unidentified problem with most emulators' Kempston implementations. Use an alternative control method if possible.

Q: I tried the Low Audio Sample Rate option and it sounds dreadful!

A: Yes! For music which uses high frequencies to produce more interesting 48K sound you hit problems with Aliasing Distortion. It is difficult to process the sound data to avoid this without making the system very slow.

Q: I was running a couple of copies of ASp with "Filter Off" control selected. Now, no matter what I do, I cannot get the filter to come back on!

A: There is no arbitration of access to the Audio Filter in the Amiga system. Each copy of ASp will return the filter to the state it found it, which is OK in most cases where it is the only program to control the filter.

Unfortunately there is a logical complication with this scheme when two or more copies are run at the same time. If they do not Stop in the reverse order in which they were Run, it can cause the filter to be stuck Off. :(

Q: I have written an ARexx script to control ASp, but it always complains that ASp's "Host Environment not found".

A1: By default, ASp names its ARexx port with a suffix, 0-9, to allow multiple copies to run simultaneously. Alternatively it is possible to explicitly specify a name. Make sure your script knows which port name ASp is using.

A2: Calling the script from within ASp (Project->Run ARexx...) will always be

able to address ASp's port by default, regardless of its name.

Q: XYZ program crashes, misbehaves, or otherwise goes wrong.

A1: ASp doesn't have a perfect emulation, although it's as accurate as possible.

If you can determine exactly which CPU instruction or flag is causing the problem, please let me know ASAP!

A2: Are you sure the snapshot you are using is not corrupted?

Q: I have a .Z80 snapshot file which loads into other emulators but not into ASp.

A: ASp's .Z80 loader is very fussy. Some emulators will save out files in an incorrect format. This often results in loss of data, although the program appears to work for the most part. ASp currently only loads files which adhere strictly to the file format specification. In future there may be a "tolerant loader" option added.

Q: I have saved a .Z80 file from ASp which ASp refuses to load back in!

A: Argh! :-) Please contact me with details.

Q: The sound/music in program XYZ is juddery or broken.

A: You have insufficient CPU power to emulate this program at full speed. Try cutting out some potentially unnecessary options like Higher Accuracy Timing, and increase the FrameSkip value.

Q: ASp runs fine with graphics card screenmodes, but is hopeless with AGA. What can I do to help?

A1: Install a fast C2P OS patch. I recommend BlazeWCP by Rick Pratt.

A2: On any given CPU configuration, ASp is likely to perform worse with an AGA mode than a gfx-card mode. A future version may contain a special mode to help with this problem, but it may be best to avoid high scan-rate AGA screens like DBLPAL or Productivity. You could also try to use the Adaptive FrameSkip mode.

Q: I can't seem to get ASp to load files from an XFH partition I have set up. The docs say it should work. What's wrong?

A: You need to use XFH V1.35 or better. ASp uses the AmigaDOS 2.0+ function ExamineFH() which requires the underlying filesystem to support the packet "ACTION\_EXAMINE\_FH".

Q: Will you share the source code for ASp? Will you release it publicly?

A: At the moment, I don't want to give sections of the source to third parties for use with "competing" projects. I feel our already small community would benefit more from a choice of different software rather than a small pool of commoned software.

Eventually, I think I will release the source publicly or pass it in its entirety to someone who wishes to continue with it.

Q: Why are you spending time coding one old computer to emulate an even older one? Have you nothing better to do?

A1: No.

A2: Some people seem to like it...

## 1.93 Contended RAM

Spectrum models have a "ULA" chip which concerns itself with several things, including generating the video display. In order for it to be able to do this it has to have priority in accessing the display RAM region(s).

Since the main Z80 CPU can therefore be temporarily blocked from accessing these memory areas they are known as "contended" memory regions.

Running code which resides in this memory or accessing data in it will affect the time taken by the CPU to perform the operation. The exact effect depends on when the access takes place in relation to the video beam.

Since multicolour "rainbow" effects by necessity access the display region throughout the video frame and are time-critical to achieve the desired effect, such timing issues need to be taken into account to reproduce realistic rainbow effects in an emulator.

Software which does not produce rainbow effects is unlikely to be substantially affected by the absence of this additional timing accuracy.

## 1.94 Information about ASp's use of mmu.library

The mmu.library by Thomas Richter is an Amiga shared library which allows more than one program to control the MMU without unpleasant interaction.

Historically, the Amiga OS has not offered support for arbitration of access to the MMU hardware. Software wishing to use it have been forced to poke around in proprietary MMU tables, attempting to interpret what is already there in order that it may add its own functions. In a lot of cases this would work when one program did it, but if two or more attempted it, chaos could result.

Using the mmu.library to access the MMU provides two definitive advantages. First, the coding of the MMU is abstracted behind a well-defined API, whereas writing real code for the real thing can be tricky. Secondly, a program accessing mmu.library gets all the benefits of having an MMU without having to concern itself about other running software. Programs using the MMU via the library are guaranteed not to trip over each others MMU code.

ASp uses the MMU to provide the 128K paging feature. In a 128K Spectrum, hardware exists to cause "pages" of RAM to appear in a common address space. In some cases, the same page can appear in two places at once! There are ways to achieve this without the use of an MMU, but they are slow and/or problematic.

ASp requires at least Version 42 of the mmu.library.

At present, ASp needs the mmu.library regardless of whether you wish to run 128K or 48K software. In future it may be modified to drop back to a 48K-only mode if no MMU is available.

## 1.95 What's an MMU?

The MMU is the Memory Management Unit of the CPU. 68030, 68040 and 68060 CPU's have them built in, 68020's can have a separate chip to perform the function.

The action of an MMU is to take a "logical" (ie: what the running program asks for) address of a RAM region and, transparently, convert that to a physical (ie: real) address in real RAM where the data resides. It is also capable of assigning characteristics like Write-protected to the RAM region.

An MMU is the core of a Virtual Memory system. Data currently unused by the running program(s) gets saved to disk and other data put in its place. The programs that wish to access that data do not know that it has been transferred to disk. When they try to access it, the data may get placed in a different area of RAM, but the MMU quietly converts the requests for the old area into accesses to the new area. The running program is unaware of the change.

ASp exploits these abilities by maintaining an image of the entire 128K Spectrum RAM area and then dynamically mapping selected pages into the address map used by the emulated Spectrum. This allows the CPU emulation core code to remain simple and therefore fast.

## 1.96 World Of Spectrum

The World Of Spectrum Website is the official site for all things relating to Spectrums. There are large numbers of documents about everything from game instructions to detailed workings of a Spectrum. There are box scans, adverts from old magazines, cover scans of magazines, famous names of the times and of course, a huge archive of downloadable programs and games for use with the Spectrum.

The maintainers of this site have gone to great lengths to try to get the agreement of the copyright holders for the information contained on it.

You can go there by looking at the URL:

<http://www.void.jump.org/>

## 1.97 ASp Support Web Pages

Updates, extra information, beta versions, prereleases, ROM images, rambling waffle, all are available from time to time from the ASp support pages, part of my humble website.

Take a look at:

<http://www.greew.freemove.co.uk/ASp.html>

## 1.98 Contacting the Author

ASp is written by Ian Greenway.

You can contact me by email at: [iang@ukonline.co.uk](mailto:iang@ukonline.co.uk)

Feel free to offer comments, suggestions, support, criticism, etc. If nobody tells me they're using the software, I will stop developing it. I cannot guarantee your suggestions will ever be implemented, but all suggestions are considered.

I am also happy to answer any problems or questions you have with using ASp. Please read the appropriate guide section and the FAQ's before asking, though, that's why I spent hours writing them! :)

Oh, and if you've got any money, women, fast cars, etc. you don't want, I'll take 'em off your hands! ;-))

Also, check out [my website](#) .

## 1.99 Index

A

[adaptive frameskip](#)

[add AY chip to 48K model](#) , menu item

[ARexx](#)

[commands](#)

[portname, defining](#)

[scripts](#)

[AREXXPORTNAME/K](#) argument

[ASp web pages](#)

[audio](#)

[filter control](#)

---

options

save mode

author, contact

B

BLANKPOINTER/S argument

C

channel mode , audio

command line arguments

copyrights

credits

cursor joystick

custom keys joystick

D

delay during TAP load , menu item

distribution conditions

E

Eject , GUI function

F

FAQ's

features , overview

filter off, audio

FKEYS/S argument

frameskip setting

G

GETPAGE , ARexx command

GETVERSION , ARexx command

H

HELP , ARexx command

higher accuracy timing , menu item

history, version

I

INSERT , ARexx command

Insert... , GUI function

installation

introduction

J

joysticks

cursor

custom keys

---

Kempston (normal)

Kempston (smart)

Sinclair 1

Sinclair 2

K

keyboard, usage

keep Kempston interface , menu item

Kempston (normal) joystick

Kempston (smart) joystick

L

LOAD , ARexx command

Load... , GUI function

loading snapshots

LOADPATH/K argument

M

machine type selection

MMU , description

mmu.library

N

NOBLANKPOINTER/S argument

NOFKEYS/S argument

O

P

PEEK , ARexx command

POKE , ARexx command

Q

quiet 48K beeps , menu item

QUIT , ARexx command

Quit , GUI function

R

READRAM , ARexx command

RELOAD , ARexx command

ReLoad , GUI function

requirements

RESET , ARexx command

Reset , GUI function

reset on machine change , menu item

REWIND , ARexx command

Rewind , GUI function

---

ROM images required

ROMNAME48/K argument

ROMNAME128/K argument

ROMNAME128PLUS2/K argument

ROMPATH/K argument

RUN , ARexx command

RUN , GUI function

S

sample rate, audio

SAVE , ARexx command

Save... , GUI function

saving snapshots

SAVEPATH/K argument

screenmode selection

Sinclair 1 joystick

Sinclair 2 joystick

Spectrum, description

SNA , snapshot format

snapshots

legalities

loading

saving

STOP , ARexx command

STOP , GUI function

T

TAP files

using

file format

tooltypes

U

usage guide

V

version history

video configuration options

W

World Of Spectrum website

WRITERAM , ARexx command

X

Y

Z

Z80 , snapshot format