RetroArch End-User Guide

January 1, 2014

Welcome to RetroArch!

RetroArch does things differently from other programs. In this guide, you will learn what RetroArch is, how it works and the various things you can do with it as an end-user.

This guide is aimed at the end-user. It does not intend to be a comprehensive reference for anything and everything to do with the libretro project and/or RetroArch.

Contents

Part I

What is RetroArch?

RetroArch is a cross platform architecture that aims to become a 'one-stop shop' for emulators, videogames, multimedia, augmented reality and other areas of interest. It manages to do all of these things by implementing a specification known as the libretro API (Application Programming Interface).

Libretro is a powerful interface that allows you to port applications to the spec and be able to run them on any libretro-compatible client in existence.

RetroArch is the 'official' libretro client. Right now it is available on many platforms and it aims to deliver the most optimal performance possible on a given host platform. You will generally find that RetroArch will be first in implementing new features and/or additions that get added to the libretro interface.

RetroArch is most well known for an entire suite of emulators that have been ported to the libretro specification and are therefore able to be run in RetroArch. Therefore, it has been compared in the media to other multi-system emulators, such as OpenEmu, and/or MESS. Note that we don't particularly approve of this attempt to pigeonhole RetroArch - we don't think of libretro and/or RetroArch as being limited to emulators, or even games for that matter.

Over the next few months the distinction between RetroArch and these types of programs will start becoming more and more apparent.

Part II

What platforms can I use RetroArch on?

Being on every system anywhere is the most important goal of RetroArch. Therefore, it is available for many platforms - for free. The program (except for minor cosmetic differences per version) should behave more or less exactly the same from one platform to another.

The list of available devices and/or operating systems include:

- PC (Microsoft Windows XP/Vista/7/8/8.1)
- Mac (OSX Snow Leopard [10.6.8] up to Mavericks [10.9])
- Linux
- Android (version 2.3 and higher)
- iOS (version 6.0 and higher [*])
- Blackberry Playbook
- Blackberry 10
- PlayStation3 [*]
- Nintendo Wii [*]
- Nintendo GameCube [*]
- Microsoft Xbox [*]
- Microsoft Xbox 360 [*]
- Browser / Javascript (Emscripten)

Some of the systems listed above (the ones marked [*]) might require a jailbreak or a modification of some sort so that RetroArch is able to run on it. This is something you will have to figure out on your own and for which we provide no support or help at all.

For the latest versions, go to the official homepage - `http://www.libretro.com`, and click on the platform you want to download a copy of RetroArch for.

In the future, RetroArch will be ported to even more systems. The aim for RetroArch is to have a program that will be able to run on as many platforms as possible, and bringing along with it all the libretro cores that RetroArch (as a libretro client) is able to run.

Part III

How does RetroArch work?

RetroArch is a plugin-driven application. Every program that you can run in RetroArch is a plugin that gets loaded by RetroArch on-the-fly. We refer to these plugin cores from this point on as 'libretro cores'.

A libretro core is an app implementing the libretro specification packaged as a plugin. The 'libretro core' defines the 'behavior' of what RetroArch will do, since RetroArch by itself does nothing. So, a core can turn RetroArch into a videogame emulator, a game, a movie player, etc. It's up to what the developer wants the libretro core to do.

Through this model, apps implementing the libretro specification form part of a larger ecosystem that every libretro-compatible client will be able to tap into. Libretro's aim is to make convergence possible - so that you can run an app on one system, then run it on another system, or play in a browser and then continue and pick up where you left off by launching the same app from your mediaplayer.

RetroArch can be thought of as the 'Proof of Concept' that demonstrates that this kind of convergence is possible. Other apps (media players, multi-system emus, CAD applications) are encouraged and allowed to implement the specification themselves.

The user can do many things with RetroArch, from playing games to watching movies (and other activities in the near future). The user is in control of what he wants to turn RetroArch into, for what purposes it will be used and what content will be run in it. There is no digital rights management or restrictions imposed upon the user.

A brief summary of all available 'libretro cores' so far will be provided at the end of this guide.

Part IV
How do you control RetroArch?

RetroArch (and libretro as a result) has been designed around one common input device - something we call the ’RetroPad’. You use this input device for controlling all aspects of RetroArch - from the game to the RetroArch system menus.

The RetroPad is a traditional videogame controller of the Nintendo/Sony mould. It has a D-pad, a Start/Select button, four face buttons, up to 4 shoulder buttons/triggers, and (optional) two analog sticks.

Unlike other apps, RetroArch is designed to be controlled with a gamepad as its primary input. We want to deliver a user experience that is as close to a traditional videogame console as possible, and to that end, we believe having to jockey between a keyboard, a mouse, and/or a gamepad/mouse is a bad user experience - when you could all do it with your gamepad anyway.

The RetroPad is an ’abstraction’ - when you run RetroArch, your own gamepad (or keyboard/touch overlay) will have a 1-to-1 mapping to this ’RetroPad’ abstraction. You map your input device of choice to all the buttons and triggers that the RetroPad supports, and from there on you can control RetroArch’s built-in menu system[1].

The mobile ports (because of touch being the primary input device) are controlled using a graphical overlay of the RetroPad that gets pasted on top of the screen. This overlay can be interacted with through touch, and it’s possible to switch between different ’pages’ of the overlay - and to switch overlays on-the-fly.

[1] On tablets and phones, you can interact with the upper-layer system menus through touch

Part V

How do I go to the built-in UI?

RetroArch has a built-in UI (User Interface) that functions nearly the same across every platform. It is called RGUI (short for Retro GUI). Most of RetroArch's features can be changed on-the-fly from this.

Just like everything else in RetroArch, RGUI is controlled with the RetroPad. Like the name suggests, it is a no-frills User Interface that is not big on eye-candy. On the plus side, it is very scaleable and works well even at resolutions as low as 320x240.

The user has two ways to bring up this menu:

- Button bind - most RetroArch versions allow you to bind 'Toggle Menu' to a key/button on your input device. By pressing this button/key, you can toggle the built-in UI on or off.

- Overlay - An overlay usually contains a button with a 'space invaders' icon in it. Touching this button will toggle the built-in UI on or off.

Part VI

How do I load content in RetroArch?

A version of RetroArch typically has three ways to load content. You can select these options from the built-in UI.

Load Content (History): You can select from a list of previously loaded content here. All the content you have ran in RetroArch will be added to this list.

Load Content (Detect Core): You can select a file from the filebrowser. It will try to detect if a core has been installed that supports the extension of the file you just selected. If it finds only one core that supports this file extension, it will autostart that core with the chosen content immediately. If it finds multiple cores that supports this file extension,it will let you choose from a list of supported cores.

Load Content: You can select a file from the filebrowser, and it will be started with the currently selected 'core'. In order to change the core currently being selected, you have to select another one from 'Load Core'.

Part VII
Listing of available cores

The number of plugin cores that get added to the libretro ecosystem keeps increasing. Covering all of these would fall outside the scope of this guide, so we're only going to cover a few ones in more detail.

We try to make an effort to have a core run on as many systems as possible. Unfortunately, there will always be instances where a core is not available for a specific platform.

1 bsnes

This is a Super Nintendo emulator based on the open-source emulator bsnes. It is an accuracy-focused emulator and as such can run almost all games for the SNES with cycle accuracy, unlike other SNES emulators. However, this comes at the cost of performance.

1.1 How to run

bSNES does not work on its own, but runs SNES ROM images. These image files need to have the following extensions

- sfc

Select one of these. If the image is verified to be OK, it will start up the game.

1.2 How do all these different versions compare?

There are several versions of bsnes out there.

1.2.1 bsnes Performance v0.92

This is a fairly recent version of bsnes' performance core - the same one used in Higan. Some games - like Mega Man X2 and X3 - are broken with this version even if you have the BIOS files placed in your System directory. Some of the games requiring high accuracy won't run correctly either. To make up for it, it has much better performance than the balanced or accuracy cores.

1.2.2 bsnes Balanced v0.92

This is a fairly recent version of bsnes' performance core - the same one used in Higan. It should play games like Mega Man X2/X3 with no problems. It is much slower than bsnes performance but still a lot faster than bsnes accuracy.

1.2.3 bsnes Accuracy v0.92

This is a fairly recent version of bsnes' accuracy core - the same one used in Higan. This should be capable of playing all SNES games accurately at the cost of performance.

1.2.4 bsnes Performance C++98 v0.85

Later versions of bsnes are written in cutting-edge C++11 and cannot be used on certain outdated compilers. This port of bsnes to C++98 was written for those compilers in mind. It is an older version of bsnes compared to the one in Higan but there should only be minor differences between this and bsnes performance v0.92.

1.3 Libretro port-specific additions

1.3.1 Cartridge folders not enforced

Cartridge folders are not forced on you, and you are still able to load SFC ROMs without having to go the Purify route.

1.3.2 More pragmatic static syncing

Higan's way of static synchronization is for all intents and purposes like RetroArch's, except ours is superior from an end-user point of view. It does not require of the user that he has to dial a figurative knob for hours on end to get the 'perfect refresh rate' of his monitor as a reference clock for the emulator - instead, it only requires that the app refresh rate you use is close enough to the refresh rate of your screen. From there on, dynamic rate control picks up the slack for whenever audio is running late, and combined with threaded video this allows highly demanding and sync-heavy cores like bsnes to run tolerably even on a very high-latency OS like Android.

1.4 Known issues

1.4.1 Mega Man X2/X3 don't run with bsnes performance core

This is a known issue and it's unlikely byuu is going to fix it for the performance core. You will have to use either Balanced or Accuracy core.

2 Dinothawr

Dinothawr is an indie game made by one of RetroArch's lead authors. It is a simple block puzzler that is similar in gameplay to Sokoban and Kickle Kubicle. It is the first game that is designed from the ground up as a core for RetroArch. The standalone version of this game has a cutdown version of RetroArch included as an integral part of the app.

2.1 How to run

1. Go to this website - https://github.com/libretro/Dinothawr and select 'Download to ZIP'.
2. Extract the ZIP file on your device.
3. Start up RetroArch. Select 'Load Content' (either 'Load Content (Detect Core)' or plain 'Load Content' will do).
4. Go to the directory where you extracted the contents of the ZIP file to (see step 2).
5. Select 'dinothawr.game'. The game should now start up.

2.2 Controls

2.2.1 RetroPad default mapping

These are the default button mappings on the RetroPad.
B BUTTON - Push block
A BUTTON - Go back to previous menu
D-PAD - Movement

3 MAME 2003

This is a multi-system arcade emulator based on a late 2003 version of MAME (to be specific, it's version 0.78). This old ancient version has been picked because MAME has gotten drastically slower over the years, and a version dating back to 2003 would still be fast enough for running most games on previous-generation videogame consoles and mobile.

3.1 Libretro port-specific additions

3.1.1 Midway DCS speed hacks re-addition

After version 0.72, MAMEdev removed various speedhacks for games using Midway DCS which led to these games being rendered more or less unplayable on current-day hardware (by 2003 standards) back then. These speedhacks have been re-integrated into the MAME 0.78 codebase, and they are of great benefit to the runtime performance of games like Mortal Kombat 1/2/3/Ultimate, NBA Jam and other Midway games using this hardware.

3.2 Notes

- You can bring up MAME's OSD GUI by pressing the R2 button on the RetroPad.
- MAME 2014 supports the RetroKeyboard as an input device as well.

4 MAME 2010

This is a multi-system arcade emulator based on a late 2010 version of MAME (to be specific, it's version 0.139). This makes it competitive with MAME4Droid Reloaded which also targets this codebase. This halfway house between MAME 2003 and MAME 2014 is mainly intended for more powerful mobile platforms that were released during 2012 to 2013. While far slower than MAME 2003, it is still measurably faster than MAME 2014.

Compared to MAME 2003, MAME 2010 has a lot more content available. For instance, Namco System 11/12 games can be played with full sound, Capcom CPS3 support was added, Killer Instinct 1/2 are fully playable, Dynamic recompilation support for 64bit got added, and more.

4.1 Libretro port-specific additions

4.1.1 Cave SH3 re-addition

The Cave SH3 drivers were removed at a specific point in time due to a legal dispute between Cave and MAMEdev. This driver has been readded to MAME 2010.

4.2 Notes

- You can bring up MAME's OSD GUI by pressing the R2 button on the RetroPad.
- MAME 2014 supports the RetroKeyboard as an input device as well.

5 MAME 2014

This is a multi-system arcade emulator based on the latest version of MAME (as of this moment, 0.151). Far slower than MAME 2003 and measurably slower than MAME 2010, it makes up for this with increased game compatibility and feature completeness.

5.1 Libretro port-specific additions

5.1.1 Cave SH3 re-addition

The Cave SH3 drivers were removed at a specific point in time due to a legal dispute between Cave and MAMEdev. This driver has been readded to MAME 2014. It will probably be readded soon to MAME mainline as well.

5.2 Notes

- You can bring up MAME's OSD GUI by pressing the R2 button on the RetroPad.

- MAME 2014 supports the RetroKeyboard as an input device as well.

6 Mupen64 Plus

This is a Nintendo 64 emulator based on the open-source emulator Mupen64 Plus. It is a current work-in-progress that aims primarily at the PC and mobile platforms, such as Android, iOS, Blackberry, and others. It ships with three graphics plugins - of which Glide64 has been worked on the most and is the most accurate, but also the slowest. Development on the other two plugins will take more work.

6.1 How to run

Mupen64 Plus does not work on its own, but runs N64 ROM images. These image files need to have one of the following extensions:

- n64
- v64
- z64

Select one of these. If the image is verified to be OK, it will start up the game.

6.2 Controls

6.2.1 RetroPad default mapping

These are the default button mappings on the RetroPad. It is assumed that your RetroPad has two analog sticks so that the N64's analog stick and the C buttons can be mapped to them.

B BUTTON - (Normal press) N64 B button / (Press and hold with R2) C Button Down

A BUTTON - (Normal press) N64 A button / (Press and hold with R2) C Button Right

Y BUTTON - (Press and hold with R2) C Button Left

X BUTTON - (Press and hold with R2) C Button Up

L BUTTON - N64 L Trigger

R BUTTON - N64 R Trigger

L2 BUTTON - N64 Z Trigger

R2 BUTTON - (Modifier button) - press and hold this plus one of the face buttons to do C button presses

D-PAD - N64 D-Pad

SELECT - Toggle between per-game control layouts

START - N64 Start button

LEFT STICK - N64 Analog stick

RIGHT STICK - N64 C buttons mapped to stick like the Gamecube's C Stick

6.3 Libretro port-specific additions

6.3.1 Plug-and-play configuration of games

N64 emulators are traditionally dependent on numerous config/database files that the user has to manually update from time to time. These database files contain core-specific settings needed to play the game correctly (or at all). This varies from Save Type to which Glide 64 settings should be enabled to get the game to look correctly.

This libretro port tries to automate as much of that stuff as possible and integrate it into the core. No longer will you need to have a Mupen64plus.ini and an RDB database from which all this information will get pulled.

The Glide64 config file which was previously a necessity has also been totally integrated into the core - to the extent that even the microcode mappings for each game are included. So that config file is not needed any longer either.

Baking in the config files for glN64 and Rice will take some more time. These have not been included as of yet.

6.3.2 60fps framerates in nearly all games with specific core settings

There have been other N64 emulator forks that have advertised being able to run games like GoldenEye 007 at 60fps. Our approach is most like Project64's in which VI Rate gets increased from the default (1500) to 2200.

Some games are framerate throttled so they can't ever reach 60fps - however, even these games will benefit from setting VI Refresh to 2200 and Framerate to 'fullspeed'. You will notice that a lot of input latency and slowdown will be removed under these settings.

6.3.3 Targets OpenGL - both desktop and GLES

We are using the GLES2 branches of Glide64, Rice and glN64. The libretro port of Mupen64plus targets OpenGL instead of spreading itself thin between Direct3D and OpenGL. We try to make sure that the code works for both desktop GL and GLES at the same time.

We have also made a lot of optimizations/changes to these rasterizer codebases so that they perform better on our target platforms. We are not anyway near done with that though.

6.3.4 Accuracy settings

Turning on all the knobs in a demanding graphics plugin like Glide64 can lead to almost glitchless graphics, but it comes at a heavy performance cost. For this reason, 'accuracy' settings per graphics plugins are introduced as core options. If you care more about double the framerate instead of the current framebuffer image being displayed on a wall billboard in Mario Kart 64, then you can set the Accuracy slider to a less high setting, and vice versa.

6.3.5 Incorporates low-level RSP next to High-Level RSP - cxd4

Certain games will have certain sound effects missing with the default high-level RSP plugin, such as GoldenEye 007. Others will simply sound wrong, such as PilotWings 64. The low-level RSP accurately emulates these effects at a signficant performance cost.

The plugin also handles certain graphics functions that the high-level RSP is missing. For instance, Resident Evil 2 requires that you use the low-level RSP plugin in order for all the FMV movies and backgrounds to work.

6.3.6 Preconfigured 'sane' controls for various targeted games

The N64 controller was not always the best pad for certain games, and having to map them to the RetroPad clearly makes that very apparent. We have remapped the controls in various games so that you can get a more pleasant gameplay experience. If you don't like this 'sanitized' control scheme, you can switch on-the-fly between it and the default control scheme by pressing Select (a button which was absent on the N64 pad - which is convenient for us since it gives us an extra button on our RetroPad to do 'stuff' with).

A couple of examples - Killer Instinct Gold is now mapped like the SNES Killer Instinct, Wipeout 64 is mapped like the PS1 Wipeout 2097, Mortal Kombat Trilogy is mapped like Mortal Kombat on the SNES and PS1, Resident Evil 2 is mapped like the PS1 version, and so on. Since the RetroPad is a replica of the Dual Shock, remapping controls this way just makes a lot more sense. And if you don't want it, you can disable it anyway and/or remap to your heart's content.

6.4 Known resolvable issues

6.4.1 Conker BFD and Perfect Dark crash with the low-level RSP plugin

Don't use the low-level RSP plugin with these games. Instead, use the high-level RSP plugin.

6.4.2 Resident Evil 2 doesn't work correctly with the high-level RSP plugin

You need to use the low-level RSP plugin for this game to work correctly.

6.4.3 I can't use the Rice plugin

Make sure that a config file called RiceVideoLinux.ini has been placed in your 'system' directory. The libretro Mupen64plus core still depends on this file being there in order to execute the Rice plugin.

6.5 Known issues

Consider the Mupen64 core right now at an advanced alpha state. Lots of issues are known about and will be looked at. The most prominent ones we are aware of are listed below.

6.5.1 iOS dynamic recompiler bugs

Right now the dynamic recompiler for iOS is not at parity with Android. This means that certain games might either crash or freeze at certain points due to bugs still lurking in the code. One example of this is GoldenEye 007 crashing at startup. We are working hard on resolving these remaining issues so that the iOS port is at least equal to the Android port.

6.5.2 Wobbling textures with Glide64 in certain games

Ground texture wobbling seems to occur in games like Super Mario 64, Star Fox 64, F-Zero X and other games depending on how close the camera is to the ground. This seems to be an issue with Glide64 in general. We will try to investigate this issue.

6.5.3 GoldenEye 007 - Glide64 - frigate level - water surface disappearing vertices

You will be able to notice significant polygon breakup when first entering this stage on the boat. Vertices of the water surface appear black depending on the proximity of the camera to the surface. This, like 3.3.2, seems to be also an issue with Glide64 that we will have to investigate.

6.5.4 Broken framebuffer effects in glN64

This is a known issue and it will probably be trivial to hook this up.

6.5.5 Other glN64/Rice issues

Most of the hard work has gone into Glide64 at this point so the glN64/Rice side has not been actively worked on - though we intend to do so eventually.

6.5.6 Tsumi to Batsu (Sin and Punishment) crashes either at startup or at exit

Known issue and currently no fix for it. Will have to be debugged.

6.5.7 Star Wars: Shadows of the Empire sometimes boots, sometimes crashes

This is a known issue, and there is currently no fix for it.

6.5.8 Pokemon Puzzle League doesn't work

This is a known issueand there is currently no fix for it.

6.5.9 Blast Corps - Crashes after Rare logo with dynarec on ARM devices

This is a known issue. The very same issue happens with Mupen64 Plus AE too. Currently, the only way to play this game on an ARM device would be to switch the CPU core to 'Cached interpreter'. This will however make the game run way too slow (possible exception being Apple A7 CPUs and competing hardware).

6.5.10 Perfect Dark is very slow in scenes with framebuffer effects or motion blur

This is a known issue. The issue is that the only way to accelerate these scenes would be to use Hardware Framebuffer Emulation, and that has drawbacks of its own such as vertices with black textures after certain scenes have ran (such as the Spycam). For now there is no good fix for this other than doing it all on the CPU, which is CPU-intensive and slow.

6.5.11 Conker's BFD - Conker has no shadow

This is a known issue. Shadows depend on a depth texture being rendered, and the only way to render these is to use Hardware Framebuffer Emulation. While we can enable this mode, it has severe drawbacks in other areas right now that precludes its use. Therefore, currently the shadow is stubbed out.

6.5.12 PilotWings 64 - There is a very annoying long black vertice obscuring a great part of the screen

This is a known issue, and it seems to happen with glN64 too and mainline Mupen64 in general. What is supposed to happen there is that a shadow on the ground should be in place of these broken vertice spans. A fix is not yet known. There is a 'duct-tape hack' core setting that makes this 'glitch' somewhat more bearable - it makes the culled vertice appear translucent instead of solid - which means you can watch through it.

6.5.13 Legend of Zelda: Majora's Mask - subscreen framebuffer image takes about five seconds to appear

This is a known issue, and a fix is not yet known.

7 NX Engine

This is a game engine capable of running the game **Cave Story** (Japanese title: **Doukutsu Monogatari**). It is based on an open-source game engine recreation called NX Engine. An extensive code rewrite has been done for the purposes of increased portability and performance.

7.1 How to run

In order to run the game, you need to have the original datafiles from the original freeware PC version.

1. Go to this website - `https://github.com/libretro/nxengine-libretro` and select ’Download to ZIP’.
2. Extract the ZIP file on your device.
3. Start up RetroArch. Select ’Load Content’ (either ’Load Content (Detect Core)’ or plain ’Load Content’ will do).
4. Go to the directory where you extracted the contents of the ZIP file to (see step 2). Select the directory ’datafiles’.
5. Select ’Doukutsu.exe’. The game should now start up.

7.2 Controls

7.2.1 RetroPad default mapping

These are the default button mappings on the RetroPad.

B BUTTON - Jump
Y BUTTON - Shoot
X BUTTON - Map screen
L BUTTON - Weapon cycle left
R BUTTON - Weapon cycle right
D-PAD - Movement
SELECT - Options screen
START - Inventory screen

7.3 Libretro port-specific additions

7.3.1 Caches all assets - avoids disk I/O overhead

The original NXEngine source was quite inefficiently written and had very high disk I/O overhead. This all but killed performance on the game consoles where disk I/O is very slow. What happens instead now is that instead of dumping all content from the main binary to disk, it caches it in memory so that there is zero overhead from accessing assets. This way, NXEngine is playable at fullspeed on any system without any disk I/O spikes - including consoles.

7.3.2 Ability to switch between 50fps (default) and 60fps

The original game ran at 50 frames per second. For the libretro port, we have made an optional toggle so you can switch the game to 60 frames per second. You can do this by pressing the Select button to go to the Options screen, and changing 'FPS' from 50 to 60, and vice versa.

8 PCSX ReARMed

This is a Sony PlayStation 1 emulator based on the open-source emulator PCSX Reloaded. It has been specifically optimized for mobile platforms, such as Android, iOS and Pandora. It also has a software-rendered graphics renderer, called NEON GPU plugin, that is more accurate than most PlayStation 1 emulators.

8.1 How to run

PCSX ReARMed does not work on its own, but runs PlayStation1 CD images. These image files need to have one of the following extensions:

- bin
- cue
- img
- mdf
- pbp
- toc
- cbn
- m3u

Select one of these. If the image is verified to be OK,it will start up the game.

8.2 Controls

8.2.1 RetroPad default mapping

These are the default button mappings on the RetroPad.
B BUTTON - PS1 X button
A BUTTON - PS1 Circle button
Y BUTTON - PS1 Square button
X BUTTON - PS1 Triangle button
L BUTTON - PS1 L1 Button
R BUTTON - PS1 R1 Button
L2 BUTTON - PS1 L2 Button

R2 BUTTON - PS1 R2 Button
D-PAD - PS1 D-Pad
SELECT - PS1 Select button
START - PS1 Start button

8.2.2 RetroPad DualShock controls

To enable this mode, you need to go to Core Options and set 'Pad 1 Type' to 'analog'. Your RetroPad is assumed to have two analog sticks to be able to use this mode. Note that not all games might work in DualShock mode, and vice versa - so switch between them if the controls don't work.
LEFT STICK - PS1 Left Analog Stick
RIGHT STICK - PS1 Right Analog Stick
L3 - PS1 L3 Button
R3 - PS1 R3 Button

8.3 Setting up BIOS files

The compatibility of PCSX ReARMed is increased a lot by using real BIOS firmware images.

PCSX ReARMed looks for the following BIOS files:

- scph1001.bin
- scph5501.bin
- scph7001.bin

These files should be placed inside your System directory. If your system directory path does not point to anything, it will try to load the BIOS files from the same directory as the CD image.

8.4 Core options

8.4.1 Increasing graphics resolution

You can increase the internal graphics resolution by enabling the core option 'Enable NEON enhanced resolution (slow)' This will force the core to render at a resolution of 1024x512[2]. This will significantly increase the quality of the graphics, but it will also impact performance. Only enable this option if you are on a fast platform with enough CPU power - otherwise you're best off leaving this core option 'off'.

[2] This mode is deactivated on games that have high-resolution interlaced graphics modes, such as Tekken 3 and Tobal 2. Setting this option to 'ON' or 'OFF' will make no change.

8.4.2 Disabling interlacing for better performance with interlaced games

High-resolution interlaced games such as Tekken 3 and Tobal 2 would typically be displayed in interlaced mode on a real PlayStation. The emulator typically renders these games with interlacing turned off. This however will have an impact on performance. If you find that your platform for whatever reason is falling below fullspeed, it can help to set the core option 'Interlacing' to 'ON'.

8.4.3 Disabling dynarec for platforms that don't support JIT compilation (Windows Phone/RT, non-jailbroken iOS)

Normally, PCSX ReARMed starts up with the dynarec CPU core being automatically activated.

There are some platforms that don't allow for JIT compilation inside an app (which is what the dynarec CPU core relies upon). These include:

- Non-jailbroken iOS
- Windows RT
- Windows Phone

You will need to disable the core option 'Dynarec' if you want to be able to run this core on such a device. Be aware that disabling the dynarec is very CPU-intensive and might result in PCSX ReARMed not being able to run fullspeed on yur device. Disable dynarec therefore only if you absolutely must.

8.5 Known outstanding issues

8.5.1 Garbled sound samples at times

Sound emulation is not perfect. At some point we intend to backport the latest sound driver from PCSX Reloaded and offer a way to toggle between the original driver and the latest PCSXR one. The original PCSX ReARMed version was geared around the Pandora, which is limited by a 1Ghz Cortex A8 CPU - concessions had to be made. Today's mobile devices are far more powerful than this and should be capable of much more demanding sound emulation.

8.5.2 Broken geometry in Jumping Flash 1/2

This is caused by an inaccurate Geometry Transfer Engine plugin used in PCSX ReARMed. A core option will probably have to be included at some point that allows the user to switch between inaccurate/fast GTE and accurate GTE emulation.

8.5.3 Random game does not work

First of all, check that you are running with a real BIOS image. You can tell if this is the case if PCSX ReARMed does not display a warning at startup telling you that it could not find a BIOS file. The internal HLE BIOS emulation is very incomplete and might cause many games to be buggy.

9 TyrQuake

This is a Quake 1 game engine based on the open-source Quake 1 game engine TyrQuake. Unlike other Quake 1-based game engines, it focuses on accuracy to the original DOS game and portability. As such, it features no graphical enhancements and the renderer is software-based.

9.1 How to run

TyrQuake does not work on its own, but runs Quake 1 PAK data archives.

Select either pak0.pak or pak1.pak from the shareware version or the registered version. If the files are OK, it will start up the game.

9.2 How to run mission packs

It is also possible to run mission packs with TyrQuake.

Place the necessary files in subdirs hipnotic/ and rogue/. Make sure the original Quake 1 datafiles are in the root directory.

Start pak0.pak from the hipnotic/rogue directory. Instead of launching the original game, it should now start the mission pack.

9.3 Controls

9.3.1 RetroPad default mapping

These are the default button mappings on the RetroPad.

Y BUTTON - Shoot
B BUTTON - Jump
A BUTTON - Weapon cycle
X BUTTON - (Press and hold) free look
L BUTTON - Strafe left
R BUTTON - Strafe right
D-PAD - Movement
START - Bring up Options Menu

9.3.2 RetroPad Dual analog controls

To enable this mode, you need to go to Core Options and set Gamepad type to 'dual-analog'. Your RetroPad is assumed to have two analog sticks to be able to use this mode.

LEFT STICK - X minus and positive X strafes left and right, Y minus and positive Y moves forward and backwards
RIGHT STICK - Free look
R3 - Center view

9.4 Libretro port-specific additions

9.4.1 Enabling/disabling software bilinear interpolation

The original Quake software renderer had no filtering routines to speak of for textures. Textures would appear pixelated if you came close to them. Enabling this (from the Quake settings menu) will apply the kernel filtering algorithm from Unreal 1's software renderer on each texture. This might look more aesthetically pleasing to people who prefer quasi-3D hardware bilinear filtered textures to nearest textures. Enabling this might come at a negligible performance cost.

9.4.2 Enabling/disabling mipmapping

Normally, Quake's software renderer has this always enabled. Objects in the distance will use lower-quality texture assets than the ones closer to the player. Disabling this (from the Quake settings menu) will render each object with the most high-quality texture available, no matter what the distance is to the player. Enabling this will come at a big performance cost - so only enable it if your platform is powerful enough.

9.4.3 Enabling/disabling animation interpolation

By default the game updates animations of its 3D models at a very sluggish rate - 11 frames per second approximately. Enabling this option (from the Quake settings menu) allows you to enable frame interpolation so that the movements of monsters will appear much smoother. Enabling this might come at a negligible performance cost.

9.4.4 Enabling/disabling third person view

By default, Quake is played from a first person perspective. Enabling this option (from the Quake settings menu) allows you to play Quake from a third person perspective. There are two camera modes to choose from - 'Clamped' and 'Clipped'. 'Clamped' will not allow the camera to pass through walls and it will try to prevent showing you any of the empty 'clip space' that exists beyond the game world. When this 'clamping' happens, the camera will zoom in to your player. Setting it to 'Clipped' does not bound the camera to the game world and does not zoom in on the player - instead, it will simply show the empty clip space that exists beyond the game world. It is up to user preference which of the two he/she prefers.

9.4.5 Setting framerate to 50 or 60fps

The original Quake never ran at a fixed framerate, and it took quite a few years for PCs to even reach the state where they were able to run Quake at 50 or 60fps. Libretro TyrQuake assumes that you will want to run Quake at either a constant 50 or 60 frames per second. You can set the framerate to either 50fps or 60fps from the Quake settings menu.

9.5 Core options

9.5.1 Increasing graphics resolution

Because Quake is using software rendered graphics and because the original sourcecode to Quake 1 had hand-tuned assembly routines for 486/early Pentium 1 CPUs (routines which can't be used on today's PCs), it might come as a surprise that TyrQuake still demands high system requirements in order to run it above the default resolution (320x200).

By increasing the internal resolution with the core option 'Internal resolution', the graphical quality of the game can be significantly enhanced at the expense of speed. After each internal resolution change, you will need to restart the TyrQuake core in order for the changes to be applied.