



RetroArch iOS (v0.9.9)

Hans Kristian Arntzen, Daniel De Matteis, Jason Fetters

May 28, 2013

Contents

1 Disclaimer	2
2 Introduction	2
3 Disclaimer	3
4 On which devices can RetroArch iOS be run?	3
4.1 Jailbroken	3
4.1.1 Advantages of an iOS device being jailbroken for RetroArch	4
4.1.2 How to use data files with cores	4
4.2 Non-Jailbroken	4
4.2.1 How to obtain RetroArch iOS on a non-jailbroken device	5
4.2.2 Note on cores for non-jailbroken devices	5
4.2.2.1 Do alternatives exist for those cores?	5
4.2.2.1.1 Mednafen PSX (alternative for PCSX ReARMed)	5
4.3 How to use data files with cores	6
4.3.1 iTunes File Sharing	6
4.3.1.1 Alternatives	6
5 How to run	6
5.1 Select a data file	6
5.2 Select a core	6
5.3 App will start	6

6	Controls	9
6.1	Cocoa overlay	9
6.2	Touchscreen overlay	10
6.3	Touchscreen menu navigation	10
6.3.1	Gamepad screen	10
6.3.2	Quick Menu screen	12
6.3.3	Gameplay screen	12
6.4	Variations	12
6.4.1	Making your own custom overlays	12
6.5	Bluetooth gamepad support	14
6.5.1	Connecting a Wiimote + Classic	14
6.5.2	Connecting a PS3 pad (Sixaxis/DualShock3)	15
6.5.3	iCade pad support	16
7	Core Config Settings	17
8	RetroArch on other platforms	17
9	About Us	19
10	Troubleshooting	19
10.1	For non-jailbroken devices only	19
10.1.1	I am not a licensed developer and I want RetroArch on my non-jailbroken device. What do I need to do?	19
10.2	For jailbroken devices only	19
10.2.1	How do I pair my PS3 pad to my iOS device?	19
10.2.2	My Wiimote/PS3 pad doesn't connect to my iOS device	19
10.3	I can't start a valid data/ROM file with a core. It says 'Game failed to start'.	20
10.4	Shaders make the gameplay unbearably slow	20
10.5	The touchscreen overlays are not good for me	20
10.6	Where is the system directory?	21
11	Credits	21

1 Disclaimer

Let it be said that we pay credit where credit is due, and most of the initial leg work (and subsequent stuff) has all been done by Jason Feters, so a big thanks to him for that.

2 Introduction

RetroArch iOS is an app that has been designed to run and play:

- Games

- Emulators
- Tech demos (OpenGL and non-OpenGL)

Emulators and games that can be run on RetroArch come in the form of pluggable 'engines' which are called 'libretro cores'. The version that you just installed already has most of the full library of 'cores' preinstalled.

3 Disclaimer

RetroArch iOS is released for free and will always be free. There are no ads (push or otherwise), there is no 'spying' going on in the form of analytics or collecting stats, there is no 'paid DLC', and on and on - all the unsavory and bad aspects of this 'new generation of computing' are not to be found here. It will never be sold with a pricetag - not even disguised as a 'donationware version'. If you happen to have 'paid' for RetroArch iOS or a derivative of it, you have been scammed and you should probably demand your money back from the scam artist in question (and scam artists they are).

Just because the GPL allows people to make derivative copies of RetroArch for commercial purposes does not mean that we support it or even approve of it. If you sell RetroArch or a derivative copy of it for any commercial purpose, you are part of the problem and you need to be learnt a quick lesson in etiquette. Note to any 'entrepreneurs' out there that might be tempted by this 'easy route to makin' some money' - I honestly wouldn't bother - we will undercut you by offering this all for free and doing a better job at it to boot. That and I severely doubt you can come up with many trinkets that will persuade people to throw away their money on a derivative version when they can have it all for free to begin with - just saying - save yourself the time and the effort, because it isn't going to work out.

4 On which devices can RetroArch iOS be run?

iOS devices currently in use can be broadly divided into two camps -

- Non-jailbroken (official)
- Jailbroken

RetroArch iOS' range of functionality depends on your iOS device being jailbroken or not.

4.1 Jailbroken

A jailbroken iOS device allows you to download apps from unofficial app stores like Cydia in addition to the official Apple App Store. It also allows you to run unsigned code.

4.1.1 Advantages of an iOS device being jailbroken for RetroArch

RetroArch iOS supports several additional features when a device is jailbroken, such as:

- BTStack support. Through this it is possible to use PS3 gamepads (Six-axis/Dual Shock 3) and Nintendo Wiimote gamepads. PS3 pads being run through BTStack also do not require Blutrol.
- Some emulator libretro cores might use a JIT recompiler, which sometimes is also referred to as a 'dynamic recompiler'. Only on jailbroken devices can such libretro cores be run.
This means that the following cores as of now are only available for jailbroken devices:
 - PCSX ReARMed
 - Desmume (upcoming)

If you do try to run these cores on a non-jailbroken iOS device, you will get an error. Another project (PPSSPP) succinctly explains why this is:

'To emulate advanced systems [like the PSP] fast, the emulator needs to translate the machine code language [of the PSP] to the machine code language of your PC or mobile device at runtime. This is done with a "Just-In-Time recompiler" or JIT, also known as a Dynarec. PPSSPP has JITs for x86 and ARM.

For a JIT to function, an app needs to have the ability to generate machine code at runtime. This is allowed on Windows, Mac, Linux and Android, while it is completely disallowed on non-jailbroken iOS and on App Store Mac apps, and on Windows Phone 8.'

4.1.2 How to use data files with cores

Cores need data files in order to work. Most emulator-based cores will typically need a 'ROM' image, game-based cores will typically need some kind of 'executable'. On jailbroken devices, you can transfer these files over to the 'Documents' directory yourself.

If this directory does not yet exist, you must manually create it in your home directory. WARNING: Make sure that the folder gets created with non-root permissions - otherwise you will run into permission problems later when trying to save states, and/or certain ROMs/ZIPs will not load.

4.2 Non-Jailbroken

A non-jailbroken iOS device only allows you to download apps from the official Apple App Store. In addition to this, it also only allows you to run signed code for your own device.

4.2.1 How to obtain RetroArch iOS on a non-jailbroken device

If you are an officially licensed Apple developer, it is possible for you to compile and install RetroArch on your non-jailbroken iOS device for your own personal use.

To do this, you will need:

1. Your own iOS Apple developer account
2. Your developer account set up on your computer (your certs, etc.)
3. A Distribution provisioning profile for RetroArch (a wildcard profile is fine and suggested)

If you want to know how to compile RetroArch iOS yourself, read this: <https://github.com/Themaister/RetroArch/blob/master/ios/README.md>.

If you are not a developer, then things become harder if not impossible since you lack the means to compile and install software on your own. There might be a grassroots aftermarket solution that will spring up to help solve this vacuum or there might not - just don't bug us (the official developers) about it since we will have no part in it for obvious reasons.

4.2.2 Note on cores for non-jailbroken devices

Like previously discussed in the section 'Jailbroken', there is a small list of cores that won't work on non-jailbroken devices because non-jailbroken iOS devices don't allow you to run JIT recompilers at runtime.

The cores that don't work on non-jailbroken iOS are:

- Desmume (upcoming, not included right now)
- PCSX ReARMed

Therefore - in case you are thinking of building and installing RetroArch yourself on a non-jailbroken device, it would make sense to exclude these two cores from your install - since they won't work anyway.

4.2.2.1 Do alternatives exist for those cores?

4.2.2.1.1 Mednafen PSX (alternative for PCSX ReARMed) This is an alternative PlayStation1 emulator core which does not depend on a JIT recompiler/dynarec to function and thus should work on non-jailbroken iOS device. The problem with this core though is that it is not suited at all for low-powered ARM devices - on desktop PCs you will need at least a Core 2 Duo (preferably a Core i3) to run games at fullspeed with Mednafen PSX, so unfortunately as of now it looks like no iOS device is powerful enough to run games at fullspeed with this core. This might start to change starting with iPad 5/iPhone 6 and up though this is speculation for now.

You will notice that this core is not included with the 'jailbroken' package. This is because it makes little sense to have an alternative core that is at least 20x slower than PCSX ReARMed which was made with ARM devices in mind.

4.3 How to use data files with cores

On jailbroken devices, you can transfer these files over to your 'Documents' directory yourself. On a non-jailbroken device, you would have to go for an alternative.

4.3.1 iTunes File Sharing

iTunes File Sharing is enabled on RetroArch. You can simply drag your data files into the RetroArch app. They will be available the next time you launch the app.

4.3.1.1 Alternatives Alternatively, you can use something like iExplorer to manually copy files over. Doing this will give you the benefit of being able to use directories, since iTunes File Sharing does not support directories.

5 How to run

5.1 Select a data file

Cores need data files in order to work. Most emulator-based cores will typically need a 'ROM' image, game-based cores will typically need some kind of 'executable'.

At startup, RetroArch iOS shows you a list view. On jailbroken devices it will default to the 'Documents' directory in your Home directory. From here, select a data file.

5.2 Select a core

A 'libretro core' supports games with certain extensions. To consult which core does what, you should read the 'RetroArch Cores Manual'.

After you have selected the data file, RetroArch iOS will show you a 'filtered' list of cores which might be compatible with the data file that you just selected. It does this by looking at the data file's extension and seeing if any cores support that specific extension. Select the 'core' that you want to load this data file with.

5.3 App will start

After you have selected the right data file for the right core, the app will start.

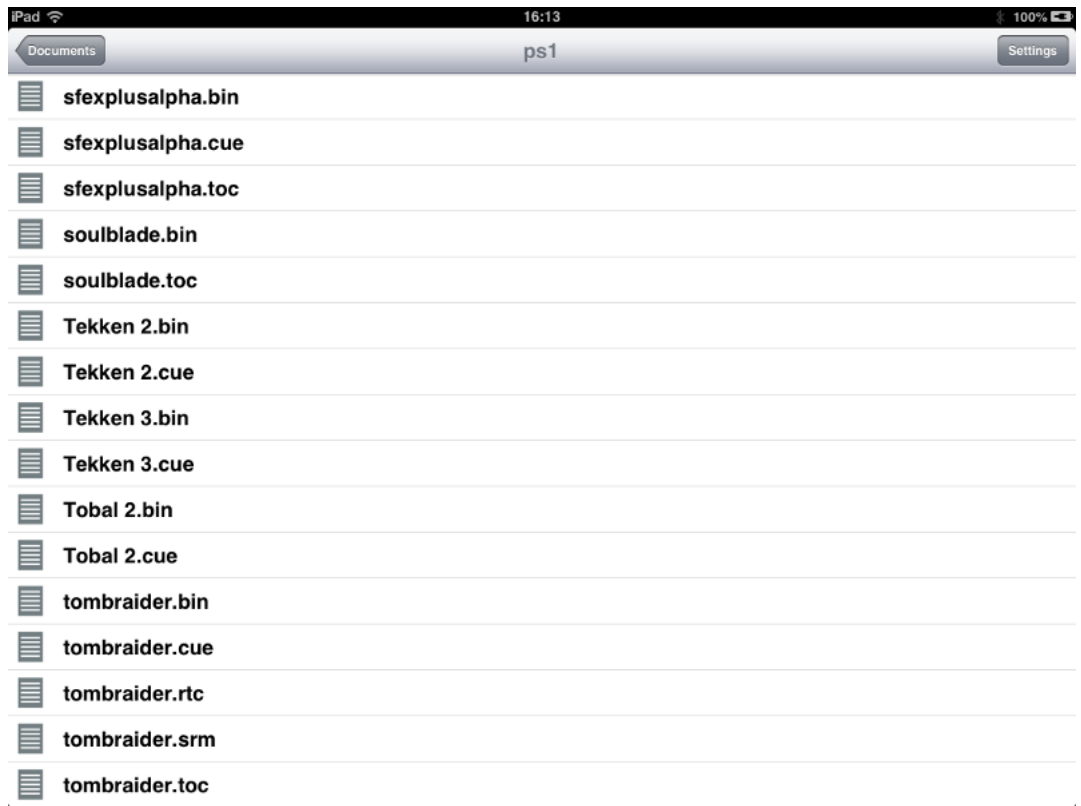


Figure 1: Select a data file from the list view.

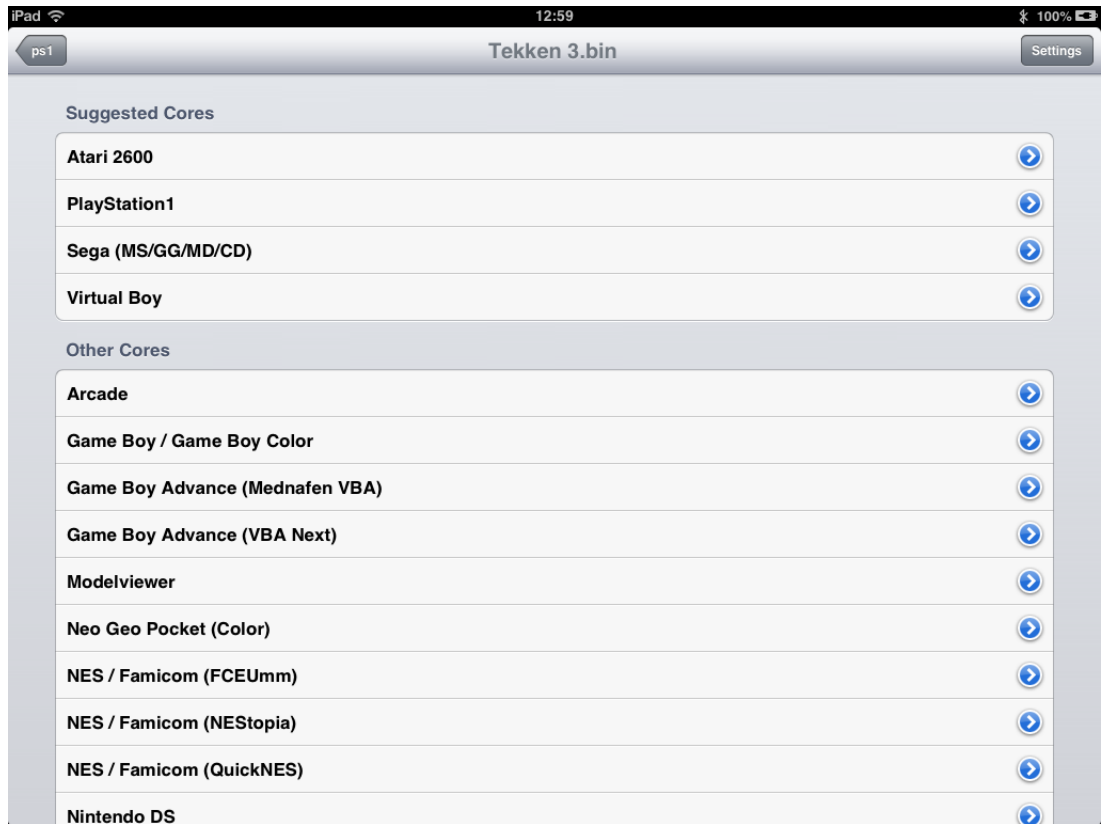


Figure 2: After selecting the data file, the app will suggest a couple of cores that are able to handle this file. Select the one which you want to use. If the one you want to use is not listed under 'Suggested Cores', then look at the cores listed under 'Other Cores'.



Figure 3: The translucent arrow icon at the top is the region of the screen that you need to press to bring up the 'Cocoa overlay'.

6 Controls

6.1 Cocoa overlay

RetroArch iOS (unlike the Android and Blackberry ports) has an additional overlay menu which we call the 'Cocoa overlay'. When you first startup RetroArch with a game, you will briefly see on the top midcenter of the screen a translucent arrow icon that will fade out after a few seconds.

By pressing this region of the screen, you will bring up the Cocoa overlay. From here, you can do a couple of actions without having to use a touchscreen input overlay.

RESUME GAME - Return back to the game and close the Cocoa menu.

TOGGLE RGUI - Bring up RGUI (the built-in menu system) and/or exit RGUI.

SYSTEM CONFIG - Go to the System Settings screen.

CORE CONFIG - Go to the Core Config screen, which has core-specific settings

SAVE STATE - Save to the currently selected state slot

LOAD STATE - Load from the currently selected state slot

RESET - Reset this core

EXIT - Quit the current game and go back to the Cocoa core/game selection screen.

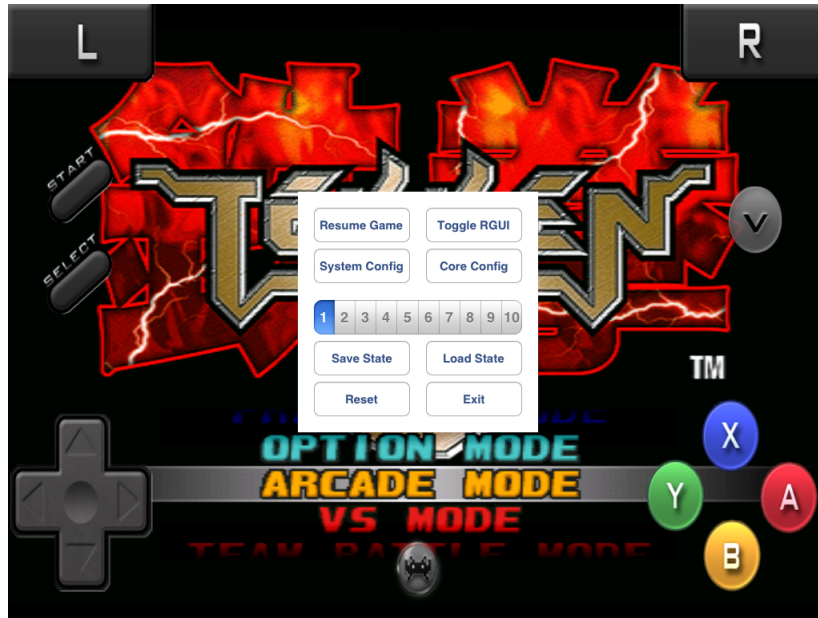


Figure 4: The translucent arrow icon at the top is the region of the screen that you need to press to bring up the 'Cocoa overlay'.

6.2 Touchscreen overlay

RetroArch iOS uses an overlay as a 'mock' gamepad to play with. The 'overlay' controls will always be bound to Player 1.

The gamepad abstraction that you see here is what we call the 'RetroPad'. When you make a libretro core and when your core supports a gamepad, it will always use as its gamepad model this pad.

The RetroPad has the same face and shoulder buttons as a Super Nintendo gamepad. In addition to this, it also has the L2/R2/L3/R3 additional buttons and the twin analog sticks from a Sony DualShock gamepad.

6.3 Touchscreen menu navigation

Each touchscreen overlay has a couple of screens that can be navigated to. To go to the next screen of the overlay, you press the 'circle' icon at the bottom.

Most of the overlays that come bundled with RetroArch Android have the same screen order.

6.3.1 Gamepad screen

You can control the game with this screen.



Figure 5: 'RetroPad overlay' screen.



Figure 6: 'Quick Menu' screen.

6.3.2 Quick Menu screen

The actions on this screen have various effects on the game currently running.

- LOAD STATE - Load a save state from the currently selected save state slot.
- SAVE STATE - Save state to the currently selected save state slot.
- STATE MINUS - Go back one save state slot.
- STATE PLUS - Go forward one state slot.
- REWIND - Rewind the game in real-time. Note - the 'Rewind' option needs to be enabled at the Settings menu or else this option won't work.
- SLOWMOTION - Press and hold this button to let the game run in slow-motion.
- RESET - Resets the game/system.
- FAST FORWARD - Fast forward the game in real-time.
- NEXT SHADER - Load the next shader in the folder (NOTE: only if shaders are enabled)
- PREVIOUS SHADER - Load the previous shader in the folder (NOTE: only if shaders are enabled)

6.3.3 Gameplay screen

This screen is useful for when you are playing with an USB or Bluetooth gamepad but you would still like to have access to the Quick Menu or Gamepad screen without outright disabling overlays. If you press the 'icon' at the bottom of this screen, you will go back to the 'Gamepad' screen'.

6.4 Variations

RetroArch iOS comes packaged with a number of different-looking overlays. Below is an image showing the different overlays:

You can select between a number of different overlays from either the Settings menu and/or RGUI.

6.4.1 Making your own custom overlays

You can make your own custom overlays for use with RetroArch Android. If you want to learn how to do this, you should read the 'Overlay Guide'.

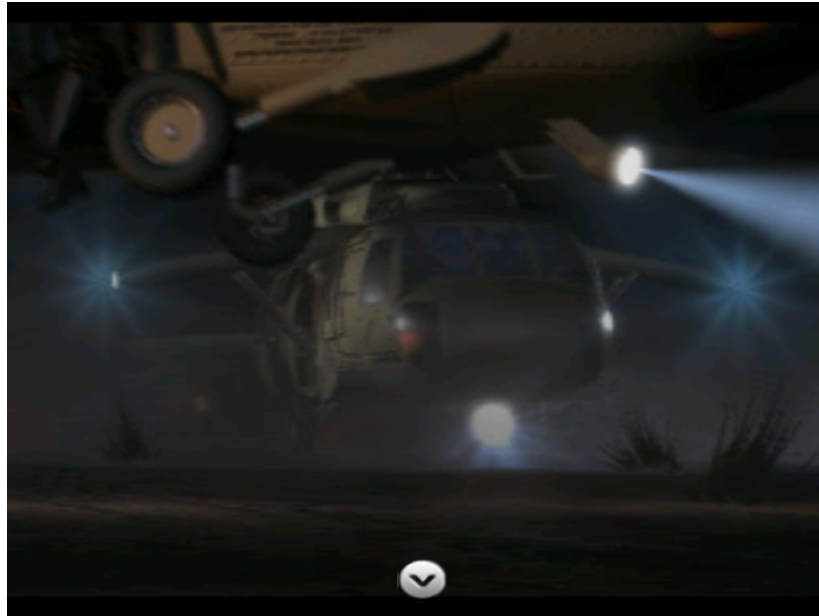


Figure 7: 'Gameplay' screen.

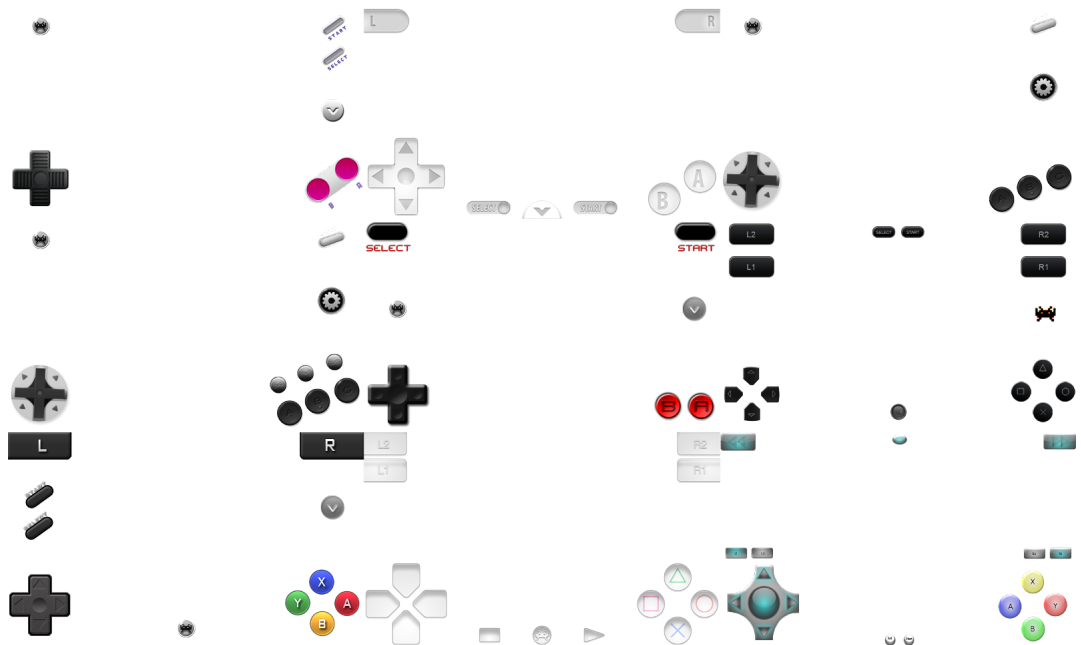


Figure 8: All the default high-resolution overlays packaged with RetroArch iOS.

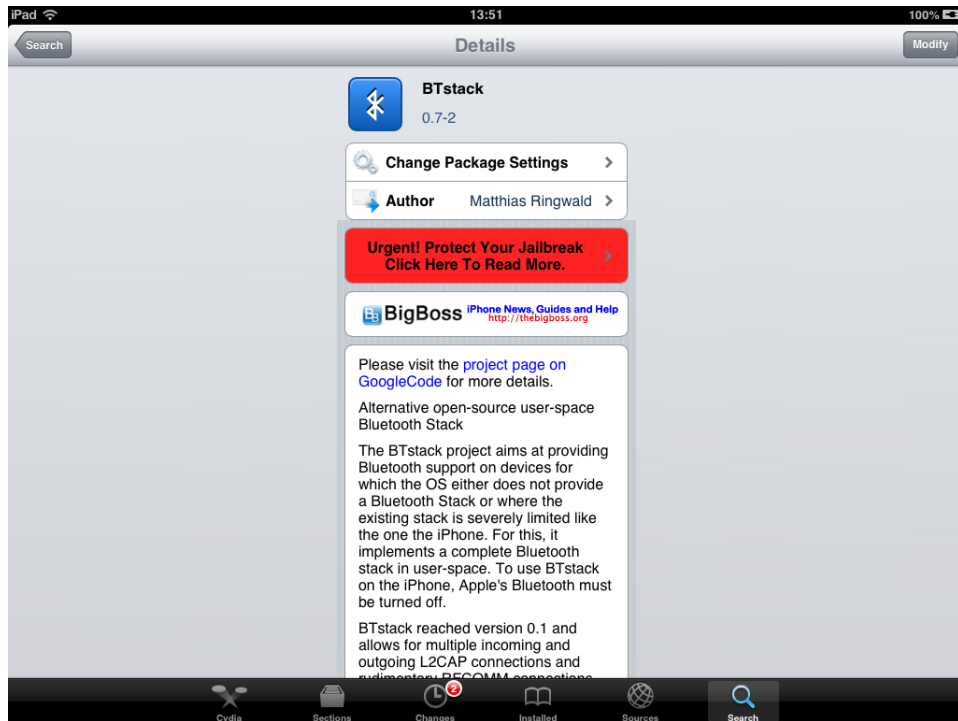


Figure 9: You need to install BTstack from Cydia first before you will be able to use PS3 and Wiimote Classic apps with RetroArch.

6.5 Bluetooth gamepad support

On jailbroken iOS, your gamepad options are increased over non-jailbroken due to the ability to use BTstack so that you can use PS3 and Wiimote Classic pads.

To do this, you will first need to make sure that you installed BTStack from Cydia Store.

6.5.1 Connecting a Wiimote + Classic

Make sure that 'Enable BTstack' is set to ON. You can do this from the initial 'screen' by clicking on the Settings icon which should take you to the 'RetroArch Settings' screen.

Press 1+2 simultaneously on the Wiimote and wait until the LED on the Wiimote is set to '1'. This means that the Wiimote + Classic is connected. From here, you can configure the controls for your Wiimote / Classic in one of two ways:

- 1 - Go to 'Core Config Settings' and configure the buttons one by one with the Wiimote + Classic.
- 2 - With the input overlay - go to the RGUI Menu, go to 'Input Options', go

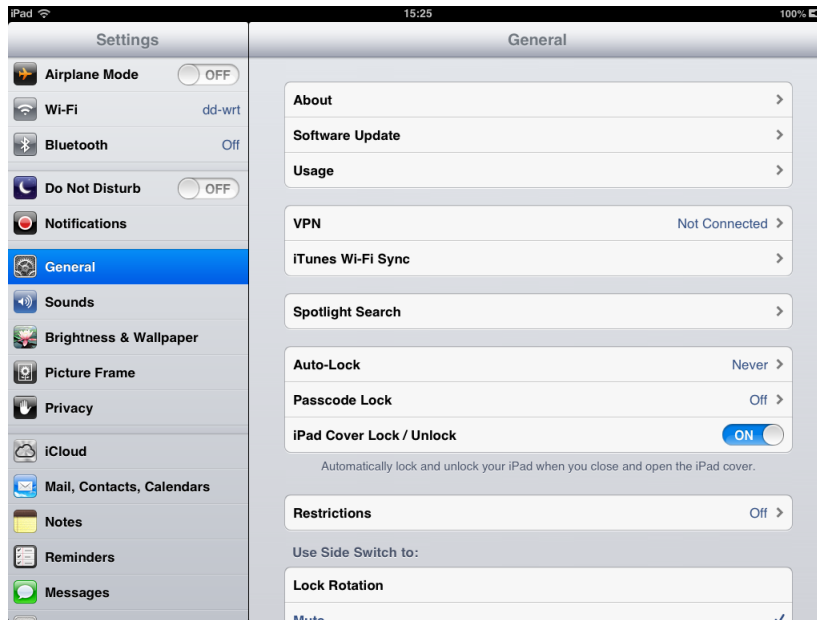


Figure 10: Select 'About' from this screen.

to 'Device' and press Left/Right until 'Device' says 'Sixaxis/DualShock3'. The keys should now be properly configured¹.

6.5.2 Connecting a PS3 pad (Sixaxis/DualShock3)

Connecting the PS3 pad to RetroArch requires a bit more work than the Wiimote.

First, you need to 'pair' the PS3 pad to the Bluetooth address of your iOS device. To pair your PS3 pad to the iOS device, you therefore first need to find out the Bluetooth MAC address of your iOS device.

You can easily find this by following the steps laid out below:

1. Go to the main menu screen of your iOS device and select 'Settings'.
1. Under the General tab, go to 'About'.
1. If you have an iOS device with Bluetooth support, it should show the 'Bluetooth' MAC address at this screen. Write it down - you will need it later.

¹Ignore the menu items for all the 'buttons' saying 'Unknown' - this stuff is currently not implemented in the current version so it reports 'Unknown' all the time.



Figure 11: You can read the Bluetooth MAC address from this screen (blacked out here for privacy reasons).

Now you need a program that can set a new Pair ID for your Bluetooth device. For the PS3 pad in particular - there is a program available for Windows/OSX called 'SixPair' which allows you to do this.

6.5.3 iCade pad support

Both jailbroken and non-jailbroken devices support the use of iCade as an input device. Go to 'RetroArch Settings', and set 'Native BT is iCade' to 'ON' in order to use this.

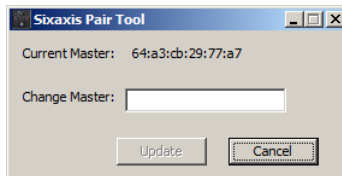


Figure 12: SixAxis Pair Tool allows you to change the current Master Pair Address of your PS3 pad. Set this to the Bluetooth address of your iOS device.



Figure 13: 'Settings' menu.

7 Core Config Settings

You can configure many aspects of RetroArch. To go to the 'Core Config' menu, bring up the Cocoa overlay (explained earlier) and press on 'Core Config'. From here, you can configure a whole range of options.

8 RetroArch on other platforms

RetroArch isn't only available for iOS. It is available on other platforms as well, including:

- PlayStation3
- Xbox 1
- Xbox 360
- Wii/Gamecube
- Raspberry Pi
- PC (Mac/Linux/Windows)

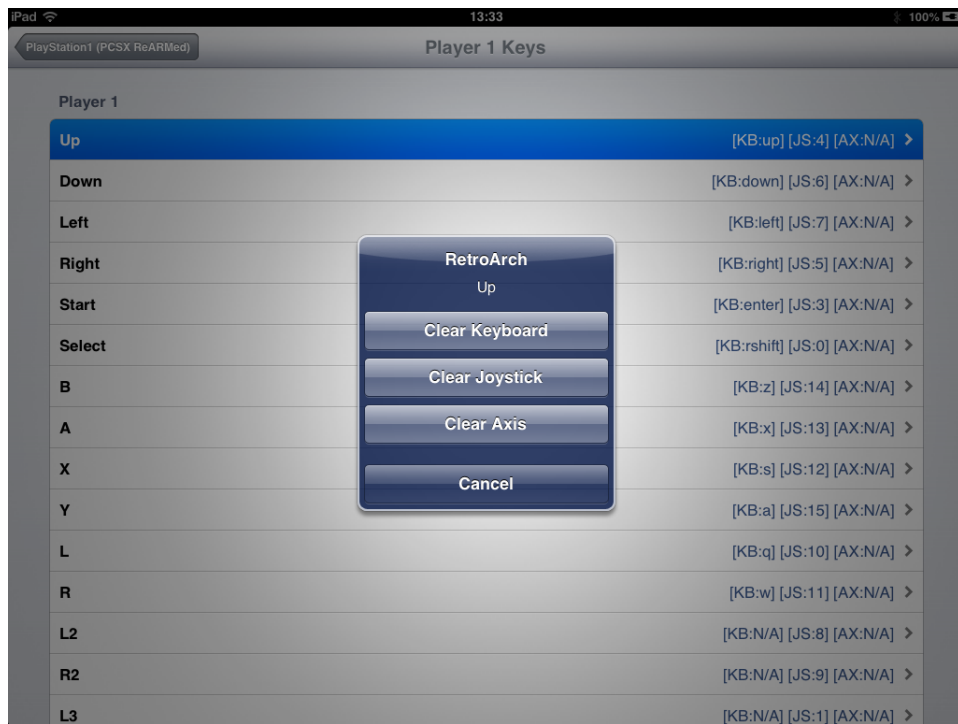


Figure 14: From the Core Config menu it is also possible to reconfigure controls on-the-fly.

- Android
- Blackberry (10/Playbook)

And it will be ported to even more platforms in the future. You might even see the libretro cores running in the official mainline version of XBMC shortly.

9 About Us

Homepage: <http://www.libretro.org>

IRC: #retroarch at freenode

Github (libretro organization): <https://github.com/libretro>

RetroArch @ Github: <https://github.com/Themaister/RetroArch>

Libretro @ Twitter: <https://twitter.com/libretro>

Libretro @ Facebook: <https://www.facebook.com/libretro.retroarch>

10 Troubleshooting

10.1 For non-jailbroken devices only

10.1.1 I am not a licensed developer and I want RetroArch on my non-jailbroken device. What do I need to do?

We can't help you, sorry. Think up a solution yourself - there might be a way - but that solution does not involve us.

10.2 For jailbroken devices only

10.2.1 How do I pair my PS3 pad to my iOS device?

Contrary to most regular iOS games, you don't really need to install Blutrol to be able to use a PS3 pad with RetroArch iOS.

10.2.2 My Wiimote/PS3 pad doesn't connect to my iOS device

Make sure that 'BTStack' is enabled first. You can enable this by pressing the 'Settings' button and setting 'Enable BTstack' to 'ON'.

You need to press the 'Pair' button on your Wiimote/PS3 pad while you are at the Cocoa GUI screen (before you select a core and a data file). Pairing your device with the iOS device will not work if you try to do

this directly ingame².

²On a Wiimote the 'Pair button' is pressing and holding 1 + 2 simultaneously. On a PS3 gamepad the 'Pair button' is the 'PS button'.

10.3 I can't start a valid data/ROM file with a core. It says 'Game failed to start'.

This could be a permissions problem to do with your Documents directory. If you created this directory as the 'root' user, it is possible that you will run into all sorts of problems later on when you try to load and/or save states in a core. Other times a certain ROM file/ZIP file will simply refuse to start and you get messages like these - 'Game failed to start'.

The best solution is if you start running into these problems, is to rename your Documents folder to some other name (like 'Documentsold'). Download a file explorer app like iFile from Cydia or something else like that. Start it up and create the 'Documents' folder in your Home directory. Make sure that under 'File Attributes -> Ownership', it says 'Owner: mobile' and 'Group: mobile' with as Access Permissions 'User: Read, Write, Execute ; Group: Read, Execute; World: Read, Execute'. Now, transfer the files from your old Documentsold folder over one by one and make sure the permissions are similar to the ones I just described above.

10.4 Shaders make the gameplay unbearably slow

This is to be expected as most shader developers target PS3/360/PC-spec GPUs, and mobile devices are lightyears removed from reaching that stage yet.

The situation might start to change over time when SoCs come out with more powerful GPUs like Tegra 4 and the next-gen ARM Malis. For now, if you find any (if not all) shaders are simply too slow on your device, just don't use them.

10.5 The touchscreen overlays are not good for me

The comfortability and usability of touchscreen overlays is very device-specific and very subject to personal taste. We allow for the following to help accommodate to your needs:

- Inside the builtin menu (RGUI->Input Options) you can toggle the opacity of an overlay.
- Inside the builtin menu (RGUI->Input Options) you can set the scaling factor of the overlay -so you can make it either bigger or smaller.

If you feel despite this that the overlay is still lacking, it is possible to either make your own overlays and add them to the existing collection or edit the existing overlays. It is not hard to do this and should only require you reading some basic documentation and having a text editor.

For more information, please read this:<https://github.com/Themaister/RetroArch/wiki/Overlay-image-configuration>.

10.6 Where is the system directory?

The system directory is a hidden directory created at `/var/mobile/Documents/.RetroArch`. Typically, libretro cores will try to load BIOS files from this directory.

PCSX ReARMed will need several BIOS files from this directory in order to be able to emulate games more accurately and bug-free.

11 Credits

RetroArch iOS

Hans-Kristian Arntzen (Themaister)

Daniel De Matteis (Squarepusher2/Twinaphex)

Jason Fetters (Meancoot)

Thanks to

Opium2k (overlay images)

Notaz (PCSX ReARMed libretro port - RetroArch ARM Linux patches)

FBA Team (for adopting libretro upstream - FBA)

Ekeeke (for adopting libretro upstream - Genesis Plus GX)

CaH4e3 (for adopting libretro upstream - FCEUmm)

Rdanbrook (for adopting libretro upstream - NESTopia Undead)

XBMC devs (for adopting libretro vis a vis RetroPlayer)

Zeromus