

Unsupported Emulator Frequently Asked Questions

A spreadsheet with what emulators are supported on which platform:

<https://batocera.org/compatibility.php>

Global emulator F.A.Q.

Why isn't <X> emulator/variant/core/fork in Batocera?

Most commonly, the answer is just “No one has implemented it yet.”

But there may also be another reason; Batocera's old slogan is “Only cores that matter”. To a certain extent, superfluous emulators that are redundant (ie. already have another emulator for that system) may be intentionally not included. With that said, if there's a good reason to include multiple emulators for a single system, such as additional options, extended compatibility, major difference in performance (such as on a weaker platforms), it is a good candidate for inclusion. From early 2021, the “focus on one emulator per system” guideline has been loosened up, and Batocera currently features more emulators/cores than ever before!



The compatibility spreadsheet linked above contains information on why certain cores may not be found on certain platforms (usually, they're simply not powerful enough). Hover your cursor over the checkmark (or lack thereof) to bring up a tooltip with the explanation.

If unable to do that on your device, the [explanation list can be found in its code form here](#).

One thing that Batocera will not budge on is implementing emulators for systems that are real “current generation” consoles. Batocera is a retro-gaming focused OS, first and foremost. For more advanced users, there is nothing in Batocera that would block you from implementing your own emulator system if you know how to set it up correctly; the various community resources and the [notable files](#) page might help you get set up. Just note that emulators added manually this way are not officially supported and we cannot offer help from either our forums or Discord if you choose to do this.

But <X> emulator runs perfectly fine in Linux!

That's cool. Check the question above and also consider that Batocera is its own operating system, ie. has different libraries/compilation methods available to it from most other distributions. For instance, a lot of scripts that rely on the entirety of Raspberry Pi OS running in the background (like RetroPie does) may not work in Batocera.

But <X> emulator runs perfectly fine in Windows! Can't you add it via WINE?

Check the above reason, but even more so. Programs running through WINE tend to have even stricter library/backend requirements, not to mention the whole mess that is setting up Windows programs correctly in WINE in the first place. Cemu being perfectly WINE compatible is nothing short of a miracle.

Per-platform emulator exclusions

Why isn't Steam on my ARM-based SBC/handheld?

- Steam is a x86 based program, and thus only available on x86/x86_64 builds. [This Github post](#) explains why Steam on ARM isn't currently feasible.
 - In the meantime however, you can use [Moonlight](#) to video stream your PC games to your SBC.

Why can't I install Windows programs on my ARM-based SBC/handheld through WINE? I thought that let you run Windows programs in Linux!

- For the same reason as above, WINE is x86 and only compatible with x86/x86_64 builds.
- Even if WINE itself could be theoretically ported to other platforms, most Windows programs themselves are still x86 only/depend on x86-based libraries. The same applies to Proton.
 - In the meantime however, you can use [Moonlight](#) to video stream your PC games to your SBC.

Why can't I install Flatpaks on my ARM device?

- Most Flatpak packages are only provided for Linux on the x86/x86_64 platform. Even if Flatpak were supported on the ARM build of Batocera, not much could be installed (notably: Steam is x86 only).

Why isn't Dolphin (Wii/Gamecube) supported on my (old) SBC/handheld?

- Current versions of Dolphin only have x86_64 and arm8 JIT.
- Older versions supported arm7 (Odroid XU4 is arm7), but they are no longer maintained.
- Dolphin requires a 2Ghz CPU (Odroid C2 maximum is 1.9 Ghz).

In other words, it could be supported for specifically older handhelds, but only through an older version of Dolphin. Batocera mostly maintains the current/latest versions of most emulators (with a few exceptions). If you'd like to port over and maintain an older version of Dolphin for your particular platform, feel free to [open a pull request](#). Do note that expected performance would still be pretty low even if achieved.

Why isn't Reicast/Flycast (Dreamcast) supported on my Odroid C2?

- The Reicast/Flycast code is not ready for the aarch64 architecture yet.

Why isn't Redream supported on my RPi 3/RPi Zero 2/other ARM board?

- Redream is a closed-source emulator, we can only work with what binaries [they have provided](#).
- On that note, Redream seems to perform worse on Raspberry Pi 4 than Flycast does. It may not even be working at playable speeds on weaker boards.

Why isn't Apple II supported on certain older Batocera versions?



GSPlus, an Apple II+2GS emulator, has been included in Batocera since **v32**.

The Apple II emulator was removed from Batocera for a period of time before **v32** for the following reasons:

- The previous Apple II emulator **LinApple** required SDL v1 to work.
- SDL v2 was patched to support GPUs on recent *SBC/x86_64*, but SDL v1 never was.

In other words, LinApple had been removed as it required an upgrade to SDL v2. That never happened.

Why isn't Mesen/Mesen-S/bsnes on my ARM-based SBC/handheld?

Mesen, Mesen-S and bsnes are very accurate, but demanding emulators. Last tested in Batocera **v34**, Mesen-S and bsnes's performance was deemed too poor to include in some of Batocera's ARM builds by default. Mesen and Mesen-S have been made available on the more powerful boards.

I want <X> emulator in Batocera. I know it is possible, and want to help, what can I do?

A person of action, I like that.

- If you're a developer (or just really handy with Python scripting, it's not too difficult to work out) you can try your hand at [implementing it yourself](#). You may need to reference [certain notable files and/or folders](#) for the live install if you don't want to go all the way through with [compiling Batocera from scratch](#), however if Batocera does not already include the necessary dependencies/libraries that your emulator needs you may need to do just that.
- If you're just a user but still want to help, you can request for the inclusion of a certain emulator in the [#suggestions](#) channel on the Discord. If the emulator is being blocked from being included in Batocera for other reasons, you can *politely encourage* the developer of the emulator to get in touch with Batocera to see what needs to be changed/implemented in the

emulator itself to get it working in Batocera (most actively-maintained software have a certain procedure to follow for making feature requests like this, you should read up on their specific documentation first). Generally speaking, old/unmaintained or Windows-only emulators will be unlikely to be included in Batocera without extreme amounts of refactoring, testing and patching. Cemu has had exceptional circumstances to make it one of the more WINE-compatible emulators out there (read: it natively supports Vulkan, only needs a few easy-to-maintain Windows libraries, has a community-maintained patch to make it more WINE-friendly, etc.), most Windows-only software does not have this luxury. That being said, we've added multiple Windows-only emulators already, so it may just be a matter of time and patience.

From:

<https://wiki.batocera.org/> - **Batocera.linux - Wiki**

Permanent link:

https://wiki.batocera.org/not_supported_emulators

Last update: **2023/02/28 05:01**

