
open_agb_firm

open_agb_firm is a bare metal interface for *natively* running GBA games and homebrew using the 3DS's built-in GBA hardware.

open_agb_firm is also a complete and better alternative to GBA VC injects (AGB_FIRM), allowing for:

- * Launching GBA files directly from the SD card
- * Writing save files directly to the SD card
- * Automatic save type configuration using an included database
- * User configuration, such as gamma settings
- * Button remapping
- * Border support for 1:1 scaling mode
- * And more to come!

Disclaimer

open_agb_firm is currently in alpha. While open_agb_firm is relatively stable and safe to use, some quirks that have not been fixed. See Known Issues for more information.

Additionally, we are not responsible for any damage that may occur to your system as a direct or indirect result of you using open_agb_firm.

Setup

- Download the latest release and extract it.
- Copy the `open_agb_firm.firm` file to your 3DS's SD card at `/luma/payloads` if you're using Luma3DS or elsewhere if you're using fastboot3DS.
- Copy the `3ds` folder to the root of your 3DS's SD card. Merge folders if asked.
- Launch open_agb_firm using Luma3DS by holding START while booting your 3DS or assign it to a slot if you're using fastboot3DS.
- After open_agb_firm launches, use the file browser to navigate to a `.gba` ROM to run.

Controls

A/B/L/R/START/SELECT - GBA buttons, respectively

SELECT+Y - Dump screen output to `/3ds/open_agb_firm/texture_dump.bmp` * If the screen output freezes, press HOME to fix it. This is a hard to track down bug that will be fixed.

X+UP/DOWN - Adjust screen brightness up or down by `backlightSteps` units.

X+LEFT - Turn off LCD backlight.

X+RIGHT - Turn on LCD backlight.

Hold the X button while launching a game to skip applying patches (if present)

Hold the power button to turn off the 3DS.

Configuration

Settings are stored in `/3ds/open_agb_firm/config.ini`.

General

General settings.

`u8 backlight` - Backlight brightness * Default: 64 * Possible values: * Old 3DS: 20-117 * New 3DS: 16-142 * Values ≤ 64 are recommended.

`u8 backlightSteps` - How much to adjust backlight brightness by * Default: 5

`bool directBoot` - Skip GBA BIOS intro at game startup * Default: **false**

`bool useGbaDb` - Use `gba_db.bin` to get save types * Default: **true**

Video

Video-related settings.

`u8 scaler` - Video scaler. 0 = none, 1 = bilinear, 2 = hardware. * Default: 2

`float gbaGamma` - GBA input gamma * Default: 2.2

`float lcdGamma` - Output LCD gamma * Default : 1.54

`float contrast` - Screen gain * Default: 1.0

`float brightness` - Screen lift * Default: 0.0

Audio

Audio settings.

`u8 audioOut` - Audio output. 0 = auto, 1 = speakers, 2 = headphones. * Default: 0

`s8 volume` - Audio volume. Values above 48 mean control via volume slider. Range -128 (muted) to -20 (100%). Avoid the range -19 to 48. * Default: 127

Input

Input settings. Each entry allows one or multiple buttons. Buttons are separated by a `,` without spaces.

Allowed buttons are `A B SELECT START RIGHT LEFT UP DOWN R L X Y TOUCH CP_RIGHT CP_LEFT CP_UP CP_DOWN`.

`TOUCH` reacts to all touchscreen presses. The `CP` in front is short for Circle-Pad.

Note that button mappings can cause input lag of up to 1 frame depending on when the game reads inputs. For this reason the default mapping of the Circle-Pad to D-Pad is no longer provided.

`A` - Button map for the A button. * Default: `none`

`B` - Button map for the B button. * Default: `none`

`SELECT` - Button map for the SELECT button. * Default: `none`

`START` - Button map for the START button. * Default: `none`

`RIGHT` - Button map for the RIGHT button. * Default: `none`

`LEFT` - Button map for the LEFT button. * Default: `none`

`UP` - Button map for the UP button. * Default: `none`

`DOWN` - Button map for the DOWN button. * Default: `none`

`R` - Button map for the R button. * Default: `none`

`L` - Button map for the L button. * Default: `none`

Example:

```
1 [input]
2 RIGHT=RIGHT,CP_RIGHT
3 LEFT=LEFT,CP_LEFT
4 UP=UP,CP_UP
5 DOWN=DOWN,CP_DOWN
```

Game

Game-specific settings. Only intended to be used in the per-game settings (`romName.ini` in `/3ds/open_agb_firm/saves`).

`u8 saveSlot` - Savegame slot (0-9) * Default: 0

`u8 saveType` - Override to use a specific save type, see values for `defaultSave` (0-15, 255) * Default: 255 (disabled)

Advanced

Options for advanced users. No pun intended.

`bool saveOverride` - Open save type override menu after selecting a game * Default: **false**

`u16 defaultSave` - Change save type default when save type is not in `gba_db.bin` and cannot be autodetected * Default: 14 (SRAM 256k) * Possible values: * 0, 1: EEPROM 8k * 2, 3: EEPROM 64k * 4, 6, 8: Flash 512k RTC * 5, 7, 9: Flash 512k * 10, 12: Flash 1m RTC * 11, 13: Flash 1m * 14: SRAM 256k * 15: None

Patches

`open_agb_firm` supports automatically applying IPS and UPS patches. To use a patch, rename the patch file to match the ROM file name (without the extension). * If you wanted to apply an IPS patch to `example.gba`, rename the patch file to `example.ips`

Known Issues

This section is reserved for a listing of known issues. At present only this remains: * Sleep mode is not fully implemented. * Using SELECT+Y to dump screen output to a file can freeze the screen output sometimes. * Save type autodetection may still fail for certain games using EEPROM. * Lack of settings. * No cheats and other enhancements.

If you happen to stumble over another bug, please open an issue or contact `profi200` via other platforms.

Hardware Limitations

`open_agb_firm` using the 3DS's built-in GBA hardware. Unfortunately, this comes with limitations compared to GBA emulators. This is a list of limitations we can't solve in software or are very hard to work around. * >32 MiB (>256 Mbit) games and homebrew. * Games with extra hardware built into the cartridge (except real-time clocks). Patches are required. * Proper save autodetection (can't find save type during gameplay). * GBA serial port (aka Link Cable). * >32 KiB (>256 Kbit) SRAM (homebrew games/emulators). * Reboots are required for switching between games. * No save states. Very difficult to implement because no direct hardware access. * Sound has lots of aliasing issues. No known workaround (hardware bug).

EEPROM Fixer

Most emulators output EEPROM saves differently than what `open_agb_firm` expects, making them incompatible. Fortunately, they are very easy to fix, using this tool by `exelotl`.

The tool also works vice versa, if you want to use a save generated by `open_agb_firm` with an emulator.

FAQ

Q: Why isn't `open_agb_firm` a normal 3DS app?

A: To access the 3DS's GBA hardware, `open_agb_firm` needs to run with full hardware access, which can only be provided by running as a FIRM.

Q: Is this safe to use?

A: Of course! While `open_agb_firm` does run with full hardware access, a lot of work has been put in by several people to ensure that nothing unexpected happens. Some backend code from `open_agb_firm` is actually used in `fastboot3ds`!

Q: What games work with `open_agb_firm`?

A: In theory, all of them, except those that fall within the hardware limitations.

Q: How can I increase the brightness?

A: Increase the value of the `backlight` setting in `config.ini`. See Configuration for more information.

Q: Why do the colors look off?

A: The default gamma settings are intended to make up for the washed out colors the 3DS LCD has. If they look weird to you, setting the `outGamma` setting to `2.2` might help.

Q: Why do some of my ROM hacks/homebrew games have saving issues?

A: `open_agb_firm` resorts to save autodetection when it can't find an entry for the game it's running in `gba_db.bin` (which only contains data for official games), and it's a bit wonky for games that use EEPROM or misleading SDK save strings.

Q: Why doesn't my save file from an emulator work?

A: There's a good chance that the save you're having issues with is an EEPROM save, which most emulators output differently. See EEPROM Fixer.

Q: My game doesn't save properly!

A: First, please ensure that the GBA ROM you are playing is not modified in any way, and matches its No-Intro checksums. Second, make sure you aren't using an existing `.SAV` file, because some may have issues for various reasons. Third, make sure your `gba_db.bin` is up-to-date. If everything seems

to be in order but the game still doesn't save properly, please open an issue so it can be fixed. In the meantime, the [useGbaDb](#) and [saveOverride](#) settings may be useful (see Configuration for more information).

Nightlies

If you want to test the latest changes you have 2 download options. The first is recommended.

With GitHub account

Log into your account, go to the Actions tab at the top, click on the first entry and download the file under [Artifacts](#) ([open_agb_firm_nightly](#)).

Without GitHub account

[nightly.link](#) is a thirdparty site to make builds available to everyone. I'm not affiliated with [nightly.link](#) or their developers and neither are they with GitHub. Use at your own risk.

https://nightly.link/profi200/open_agb_firm/workflows/c-cpp/master/open_agb_firm_nightly.zip

Compiling

To compile `open_agb_firm`, the following needs to be installed: * `devkitARM` * CTR Firm Builder or `firmtool`

Additionally, [p7zip](#) (or if available, [p7zip-full](#)) needs to be installed to make release builds. Also, make sure that the [dma330as](#) and [firm_builder/firmtool](#) binaries are in the PATH environment variable and accessible to the Makefile.

Clone this repository using `git clone --recurse-submodules https://github.com/profi200/open_agb_firm` and update via `git pull && git submodule update --init --recursive`.

Build `open_agb_firm` as a debug build via `make`, or as a release build via `make release`.

License

You may use this under the terms of the GNU General Public License GPL v3 or the terms of any later revisions of the GPL. Refer to the provided [LICENSE.txt](#) file for further information.

Thanks to...

- [yellows8](#)

-
- **plutoo**
 - **smea**
 - **Normmatt**
 - **WinterMute**
 - **ctrulib devs**
 - **LumaTeam**
 - **devkitPro**
 - **ChaN** (fatfs)
 - **benhoyt** (inih)
 - **fastboot3DS project**
 - **MAME**
 - **No-Intro**
 - **Wolfvak, Sono and all the other people in #GodMode9 on freenode/Discord**
 - **endrift, Extrems and all the other people in #mgba on freenode**
 - **Oleh Prypin (oprypin) for nightly.link**
 - ...everyone who contributed to **3dbrew.org**

Copyright (C) 2021 derrek, profi200, d0k3