

gsplus



Modern cross-platform Apple IIgs emulator and tools based on KEGS

Manual Versions:

Alpha Draft: 2016-10-27

Initial Draft: 2016-05-01

Overview & Purpose

This project is an attempt to build a better Apple IIgs emulator by modernizing the KEGS/GSport emulator family and extending it into a platform.

The first steps were a release of a new SDL2 driver. SDL or "Simple DirectMedia Layer" enables the writing of one cross-platform driver to handle input and output (video, mouse, keyboard, audio, joystick). This helps the goal of providing first-class support for all of the major platforms supported by SDL, including Mac OSX, Windows and Linux.

Part of this effort has also led to the development of an automated build and packaging pipeline using Gitlab CI. This helps our team of one continue to develop on primarily a single platform, while still ensuring that it works across many operating systems

Currently, the main supported platforms are Linux, OSX and *Windows. *Unfortunately the SDL2 version of Windows is not done as of this draft, but the Win32 API version still works well and is regularly built, though missing some features like screenshots.

Getting started

1. Download the emulator package and put it in a folder
2. Download the Apple IIgs Firmware ROMs, for either a ROM01 or ROM03 machine.
(If you have a real IIgs, there are ways to save it from your machine as well.)
3. Download some Apple IIgs software.
 - a. I recommend the excellent Apple IIgs dedicated site,
<http://www.whatisthe2gs.apple2.org.za/>
 - b. If you have a real IIgs, you can transfer your disks using ADTPro
<http://adtpro.sourceforge.net/>
4. Edit your config:
 - a. You can manually edit the config.txt file ... OR ...
 - b. While the emulator is running... hit F4 to enter config menu.
5. Boot the Apple IIgs by running the emulator
 - a. Windows: `gsplus.exe`
 - b. Mac/Ubuntu: `./gsplus`

Emulator Hot-Keys

There are several keys used to interact with the emulator while it is running. Here's a quick list of the main hot-keys and what they do.

- F4** - Configuration menu
- F5** - Config menu
- F6** - Toggle System Speed (1, 2.8, 8, ∞ MHz)
- F7** - Toggle Fast Disk Emulation
- F9** - Invert Paddles
- F10** - Toggle a2vid_palette (?)
- F11** - Toggle Fullscreen Mode
- F12** - RESET Key (so to hit "Ctrl-Reset" on an Apple II, you hit "Ctrl-F12")

- SHIFT+F5** - Screenshot

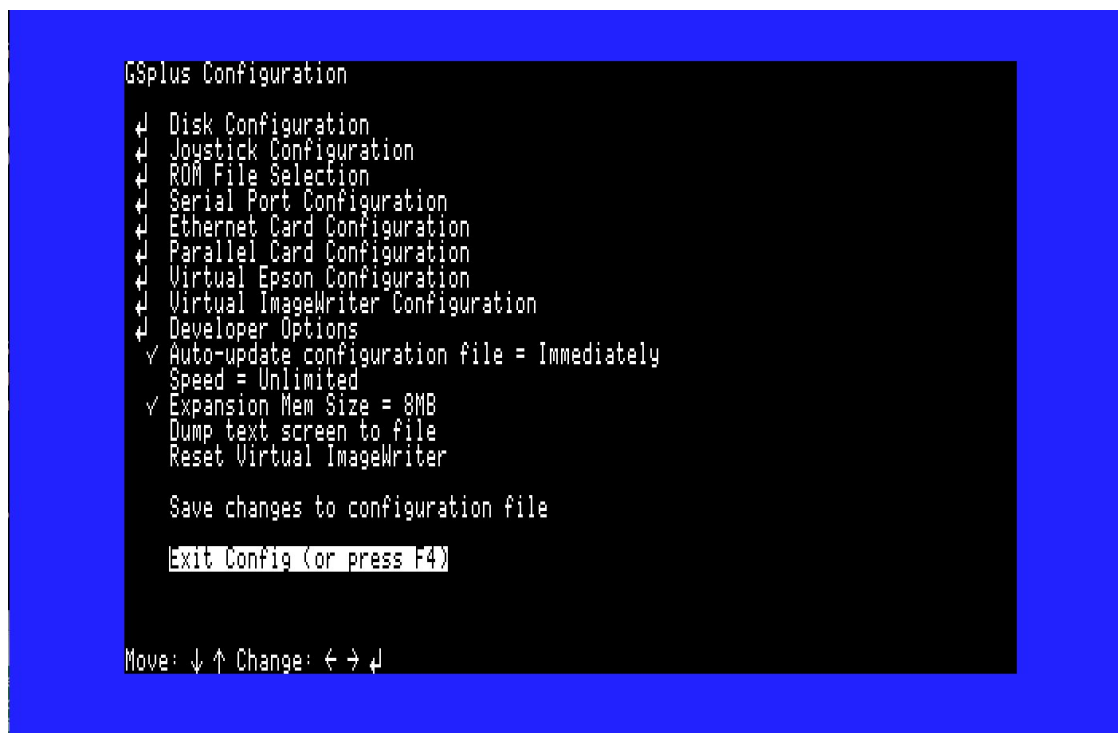
Other Input/Output

1. Mouse and keyboard should automatically work
2. Joystick with SDL2 driver is known to be working. It simply uses the “first” joystick and the first analog pad on that stick plus the first two buttons. The axes and buttons are not yet configurable. Better info forthcoming.
3. There are drivers for serial controller, ImageWriter (II?) printer, and networking. I may have broken some of this functionality to get builds working. I’m not currently familiar with the state of these drivers or their usage. More info will be added as I take time to research more of the modules.

Configuration Menu

Configuration options are loaded from, and saved to, a configuration text file. The default configuration file is *config.txt*. If you are just starting out, it’s recommended to use the configuration menu versus editing the file yourself.

Enter the configuration menu at any time while running gsplus by hitting <F4>.



```
GSplus Configuration
↓ Disk Configuration
↓ Joystick Configuration
↓ ROM File Selection
↓ Serial Port Configuration
↓ Ethernet Card Configuration
↓ Parallel Card Configuration
↓ Virtual Epson Configuration
↓ Virtual ImageWriter Configuration
↓ Developer Options
  v Auto-update configuration file = Immediately
    Speed = Unlimited
  v Expansion Mem Size = 8MB
    Dump text screen to file
    Reset Virtual ImageWriter

  Save changes to configuration file

  Exit Config (or press F4)

Move: ↓ ↑ Change: ← → ↵
```

Most importantly, you can use the Disk Configuration menu option to “mount” and “eject” disks.

Command-line Options

This is from source, some are experimental and not all may be built on your system.

-badrd	Halt on bad reads
-noignbadacc	Don't ignore bad memory accesses
-noignhalt	Don't ignore code red halts
-test	Allow testing
-hpdev	Use /dev/audio (HP/UX?)
-alib	Use Aserver audio server
-24	Use 24-bit display depth
-16	Use 16-bit display depth
-15	Use 15-bit display depth
-bw	Force B/W modes
-joystick	Ignore joystick option
-noshm	Don't use X shared memory
-dhr140	Use simple double-hires color map
-mem <i>value</i>	Set memory size to <i>value</i>
-skip <i>value</i>	Set skip_amt to <i>value</i>
-audio <i>value</i>	Set audio enable to <i>value</i>
-arate <i>value</i>	Set preferred audio rate to <i>value</i>
-v <i>value</i>	Set verbose flags to <i>value</i>
-display <i>value</i>	Set X-Windows DISPLAY= <i>value</i>
-enet <i>value</i>	Set ethernet to <i>value</i>
-config <i>value</i>	Set config file to <i>value</i>
-debugport <i>value</i>	Set debugport to <i>value</i>

Note: The final argument, if not a flag, will be tried as a mountable device.

Debugger Commands

Enter from the debugger prompt (Shift-F6)

GSplus CLI Debugger help (courtesy Fredric Devernay)

General command syntax: [bank]/[address][command]

e.g. 'e1/0010B' to set a breakpoint at the interrupt jump pt

Enter all addresses using lower-case. As with the llgs monitor, you can omit the bank number after having set it: 'e1/0010B' followed by '14B' will set breakpoints at e1/0010 and e1/0014

g	Go
[bank]/[addr]g	Go from [bank]/[address]
s	Step one instruction
[bank]/[addr]s	Step one instr at [bank]/[address]
[bank]/[addr]B	Set breakpoint at [bank]/[address]
B	Show all breakpoints
[bank]/[addr]D	Delete breakpoint at [bank]/[address]
[bank]/[addr1],[addr2]	View memory
[bank]/[addr]L	Disassemble memory
P	Dump the trace to 'pc_log_out'
Z	Dump SCC state
I	Dump IWM state
[drive].[track]I	Dump IWM state
E	Dump Ensoniq state
[osc]E	Dump oscillator [osc] state
R	Dump dtime array and events
T	Show toolbox log
[bank]/[addr]T	Dump tools using ptr [bank]/[addr] as 'tool_set_info'
[mode]V	XOR verbose with 1=DISK, 2=IRQ,4=CLK,8=SHADOW,10=IWM,20=DOC, 40=ABD,80=SCC, 100=TEST, 200=VIDEO
[mode]H	XOR halt_on with 1=SCAN_INT,2=IRQ, 4=SHADOW_REG, 8=C70D_WRITES
r	Reset
[0/1]=m	Changes m bit for I listings

[0/1]=x	Changes x bit for l listings
[t]=z	Stops at absolute time t (obsolete)
S	show_bankptr_bank0 & smartport errs
P	show_pmhz
A	show_a2_line_stuff show_adb_log
Ctrl-e	Dump registers
[bank]/[addr1].[addr2]us[file]	Save mem area to [file]
[bank]/[addr1].[addr2]ul[file]	Load mem area from [file]
v	Show video information
q	Exit Debugger (and GSplus)

Project Info

Developers inquire within :P

Main page and full source code repository: <https://github.com/digarok/gsplus>

This project has an ugly unmaintained homepage at <http://apple2.gs/plus/>

