

# sstfb

## Introduction

This is a frame buffer device driver for 3dfx' Voodoo Graphics (aka voodoo 1, aka sst1) and VoodooÂ² (aka Voodoo 2, aka CVG) based video boards. It's highly experimental code, but is guaranteed to work on my computer, with my "Maxi Gamer 3D" and "Maxi Gamer 3dÂ²" boards, and with me "between chair and keyboard". Some people tested other combinations and it seems that it works. The main page is located at <<http://sstfb.sourceforge.net>>, and if you want the latest version, check out the CVS, as the driver is a work in progress, I feel uncomfortable with releasing tarballs of something not completely working...Don't worry, it's still more than usable (I eat my own dog food)

Please read the Bug section, and report any success or failure to me (Ghozlane Toumi <[gtoumi@laposte.net](mailto:gtoumi@laposte.net)>). BTW, If you have only one monitor , and you don't feel like playing with the vga passthrou cable, I can only suggest borrowing a screen somewhere...

## Installation

This driver (should) work on ix86, with "late" 2.2.x kernel (tested with x = 19) and "recent" 2.4.x kernel, as a module or compiled in. It has been included in mainstream kernel since the infamous 2.4.10. You can apply the patches found in *sstfb/kernel/\*-2.{2|4}.x.patch*, and copy *sstfb.c* to *linux/drivers/video/*, or apply a single patch, *sstfb/patch-2.{2|4}.x-sstfb-yymmdd* to your linux source tree.

Then configure your kernel as usual: choose "m" or "y" to 3Dfx Voodoo Graphics in section "console". Compile, install, have fun.. and please drop me a report :)

## Module Usage

### Warning

1. You should read completely this section before issuing any command.
2. If you have only one monitor to play with, once you insmod the module, the 3dfx takes control of the output, so you'll have to plug the monitor to the "normal" video board in order to issue the commands, or you can blindly use *sst\_dbg\_vgapass* in the tools directory (See Tools). The latest solution is pass the parameter *vgapass=1* when insmodding the driver. (See Kernel/Modules Options)

## Module insertion

1. `insmod sstfb.o`  
you should see some strange output from the board: a big blue square, a green and a red small squares and a vertical white rectangle. why? the function's name is self-explanatory: "sstfb\_test()"... (if you don't have a second monitor, you'll have to plug your monitor directly to the 2D videocard to see what you're typing)
2. `con2fb /dev/fbx /dev/ttyx`  
bind a tty to the new frame buffer. if you already have a frame buffer driver, the voodoo fb will likely be `/dev/fb1`. if not, the device will be `/dev/fb0`. You can check this by doing a `cat /proc/fb`. You can find a copy of `con2fb` in `tools/` directory. if you don't have another fb device, this step is superfluous, as the console subsystem automatically binds ttys to the fb.
3. switch to the virtual console you just mapped. "tadaaa" ...

## Module removal

1. `con2fb /dev/fbx /dev/ttyx`  
bind the tty to the old frame buffer so the module can be removed. (how does it work with vgacon ? short answer : it doesn't work)
2. `rmmmod sstfb`

## Kernel/Modules Options

You can pass some options to the sstfb module, and via the kernel command line when the driver is compiled in: for module : `insmod sstfb.o option1=value1 option2=value2 ...` in kernel : `video=sstfb:option1,option2:value2,option3 ...`

sstfb supports the following options:

Module	Kernel	Description
--------	--------	-------------

Module	Kernel	Description
vgapass=0	vganopass	Enable or disable VGA passthrou cable.
vgapass=1	vgapass	When enabled, the monitor will get the signal from the VGA board and not from the voodoo. Default: nopass
mem=x	memx	Force frame buffer memory in MiB allowed values: 0, 1, 2, 4. Default: 0 (= autodetect)
inverse=1	inverse	Supposed to enable inverse console. doesn't work yet...
clipping=1	clipping	Enable or disable clipping.
clipping=0	noclipping	With clipping enabled, all offscreen reads and writes are discarded. Default: enable clipping.
gfxclk=x	gfxclk:x	Force graphic clock frequency (in MHz). Be careful with this option, it may be DANGEROUS. Default: auto <ul style="list-style-type: none"> <li>• 50Mhz for Voodoo 1,</li> <li>• 75MHz for Voodoo 2.</li> </ul>
slowpci=1	fastpci	Enable or disable fast PCI read/writes.
slowpci=1	slowpci	Default : fastpci
dev=x	devx	Attach the driver to device number x. 0 is the first compatible board (in lspci order)

## Tools

These tools are mostly for debugging purposes, but you can find some of these interesting:

- *con2fb*, maps a tty to a framebuffer:

```
con2fb /dev/fb1 /dev/tty5
```

- *sst\_dbg\_vgapass*, changes vga passthrou. You have to recompile the driver with SST\_DEBUG and SST\_DEBUG\_IOCTL set to 1:

```
sst_dbg_vgapass /dev/fb1 1 (enables vga cable)
sst_dbg_vgapass /dev/fb1 0 (disables vga cable)
```

- *glide\_reset*, resets the voodoo using glide use this after rmmoding sstfb, if the module refuses to reinsert.

## Bugs

- DO NOT use glide while the sstfb module is in, you'll most likely hang your computer.
- If you see some artefacts (pixels not cleaning and stuff like that), try turning off clipping (clipping=0), and/or using slowpci
- the driver don't detect the 4Mb frame buffer voodooos, it seems that the 2 last Mbs wrap around. looking into that .
- The driver is 16 bpp only, 24/32 won't work.
- The driver is not your\_favorite\_toy-safe. this includes SMP...

[Actually from inspection it seems to be safe - Alan]

- When using XFree86 FBdev (X over fbdev) you may see strange color patterns at the border of your windows (the pixels lose the lowest byte -> basically the blue component and some of the green). I'm unable to reproduce this with XFree86-3.3, but one of the testers has this problem with XFree86-4. Apparently recent Xfree86-4.x solve this problem.
- I didn't really test changing the palette, so you may find some weird things when playing with that.
- Sometimes the driver will not recognise the DAC, and the initialisation will fail. This is specifically true for voodoo 2 boards, but it should be solved in recent versions. Please contact me.
- The 24/32 is not likely to work anytime soon, knowing that the hardware does ... unusual things in 24/32 bpp.

## Todo

- Get rid of the previous paragraph.
- Buy more coffee.
- test/port to other arch.
- try to add panning using tweeks with front and back buffer .

- try to implement accel on voodoo2, this board can actually do a lot in 2D even if it was sold as a 3D only board ...

Ghozlane Touni <[gtouni@laposte.net](mailto:gtouni@laposte.net)>

Date: 2002/05/09 20:11:45

<http://sstfb.sourceforge.net/README>