

By Li, Xiao

# FreeBSD-powered LCD Advertising Displays

In 2011, 101 freestanding LCD advertising displays were installed in 11 waiting rooms along the high-speed railway between Shanghai and Xuzhou. In 2013, seven LCD advertising display towers were installed in four waiting rooms along the high-speed railway between Shenyang and Dalian. In these devices, tailored FreeBSD images with Xorg run with Intel CPUs and flash memories. The strong body of FreeBSD ensures that the advertising pictures and videos are stably played and that the electric motors in the towers behave correctly under control of the advertising programs.



Above: Freestanding LCD displays. Only the lower LCDs were driven by hosts running FreeBSD. They were playing advertising pictures customized by Mega-info Media.

Since 2000, electronic displays such as LCD and LED have been widely used in advertising all over the world. Inside those advertising devices, open-source software compiled for embedded systems supports playing audio and video, accessing an SD card or USB flash drive, and networking and remote control.



Geographical positions of Beijing-Shanghai and Harbin-Dalian high-speed lines on China's mainland.

Since 2000, China has built the longest high-speed railway lines in the world [1]. In order to develop the economic value of waiting rooms in railway stations along those high-speed lines, LCD and LED displays have been installed to play commercial advertising pictures and videos.

From 2011 to 2013, I was an advertising equipment provider for some of those locations. And I adopted FreeBSD with Xorg as the operating system in my products. The stability of FreeBSD and my software product surprised many customers with their very rare software failures.

## Products and Installations

I secured two equipment supply contracts in 2011 and 2013, respectively for a section of the Beijing-Shanghai high-speed line and a section of the Harbin-Dalian high-speed line. Map above shows the geographical positions of the two lines on China's mainland [17].

### Beijing-Shanghai High-speed Railway Line

In the section between Shanghai and Xuzhou (about 626 kilometers [6]) of the Beijing-Shanghai high-speed railway line [2-3] (about 1318 kilometers [6]), the customer ordered the freestanding LCD advertising displays shown in the inset photo on page 10 [9]. They were installed in waiting rooms of the railway stations listed in Figure 1.

The hardware configuration to run FreeBSD is shown in Table 1 [7-8]. In 2011, the ethernet and the audio controllers had been fully supported by FreeBSD's drivers `re(4)` and `snd_hda(4)`. But the GPU was somewhat of a newbie for FreeBSD, and required a newer Xorg driver `xf86-video-intel29`, which didn't belong to the main line of Xorg dependencies

in the FreeBSD ports tree.

At that time, Xorg had imported a kernel mode setting (KMS) mechanism in the GNU/Linux community. But FreeBSD hadn't kept up with them. Fortunately, `x11-drivers/xf86-video-intel29` worked fine with Xorg 7.5.1 and the onboard GPU in user mode. The display mode could be smoothly set to `1920x1080p@60Hz`, which fully conformed to the monitors' best performance. And the X-Video extension of Xorg also worked fine with

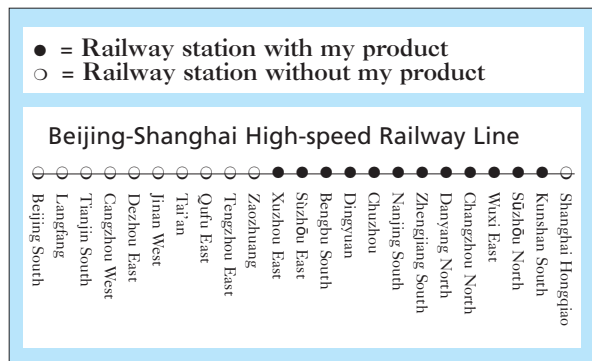


Fig. 1. The main line of the Beijing-Shanghai high-speed railway line. Since 2016, some of my products have been removed from some of those railway stations such as Nanjing South.

Motherboard	Intel Desktop Board D410PT or D425KT
CPU	Intel Atom D410 or D425
GPU	Intel GMA 3150
Chipset	Intel NM10
Ethernet	Realtek 8103EL or 8105E
Audio	Realtek ALC662
RAM	1GB or 2GB
Storage	8GB, USB flash drive

Table 1. Hardware configuration of my products for the Beijing-Shanghai high-speed railway line.

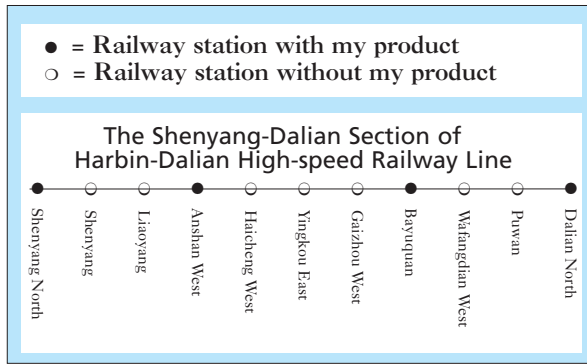


Fig. 2. The Shenyang-Dalian section of the Harbin-Dalian high-speed railway line.



LCD advertising display tower. It is playing advertising contents customized by Xinhua News Agency.

the driver, which prevented the CPU from being overloaded when playing video.

## The Harbin-Dalian High-speed Railway Line

In the section between Shenyang and Dalian (about 380 kilometers [6]) of the Harbin-Dalian high-speed line (about 918 kilometers [6]), the customer ordered the LCD advertising display towers shown in photo at left [16]. They were installed in waiting rooms of the railway stations listed in Figure 2 [4-5].

The hardware configuration to run FreeBSD is shown in Table 2 [10-11]. In 2013, the ethernet controller Atheros GbE hadn't been supported by FreeBSD. But I didn't have enough time to hack the driver on my own. I had to install an additional PCI ethernet card with RTL8139 onto the motherboard.

In this application case, every host was designed to drive 5 LCD displays via VGA ports. Besides onboard GPU, I had to install two PCI-E graphics cards with VGA and DVI-I ports onto every motherboard. At that time, the GPU of NVIDIA worked fine with NVIDIA's closed source driver [12], but the Intel GPU could only be driven by xf86-video-vesa. Fortunately again, since the Xinerama extension of Xorg was applied to combine five LCD displays as a single virtual display, the XVideo extension wasn't necessary (actually, Xinerama disables most hardware rendering); xf86-video-vesa was sufficient for this case.

## Technical Points

Tailored FreeBSD images with Xorg and my software were written into a USB flash drive or SATA SSD.

## Tailored FreeBSD File Set

To simplify the work, an MBR partition table was created in every USB flash drive or SATA SSD. Only one partition was created in every drive, and a BSD label "a" was created in the partition. The slice /dev/da0s1a (for USB flash drive) or /dev/ada0s1a (for SATA SSD) was formatted into FreeBSD FFS 2 with soft updates and journaling.

In actual use, writing operations to drives was not frequent, at no more than about 100 MB a day. But abnormal power-off might occur 10 times a day. Thus, the robust filesystem could effectively reduce booting failure. Having benefited from FreeBSD FFS 2 with soft updates and journaling, booting failures of my products have

Motherboard	GIGABYTE GA-B75-D3V
CPU	Intel Core i5 (LGA1155)
GPU	Intel HD Graphics 2500 (onboard) NVIDIA GeForce 210 (PCI-E, doubled)
Chipset	Intel B75
Ethernet	Atheros GbE (onboard, unused) Realtek RTL8139 (PCI)
Audio	Realtek ALC887
RAM	2GB
Storage	SATA SSD 120GB, Intel

Table 2. Hardware configuration of my products for the Harbin-Dalian high-speed railway line.

been caused by hardware damage only recently.

The slice `/dev/da0s1a` or `/dev/ada0s1a` is used as the root filesystem of FreeBSD. The hierarchy is shown in Table 3. Since I found “mfsBSD” made by Martin Matuška [13], I have looked through his work for references.

## Tailored Xorg File Set

To run the GUI, a file set of Xorg had to be written into the filesystem. The hierarchy is shown in Table 4.

I hadn’t integrated any typical X11 toolkit such as GTK into the flash drive.

## Use of wxWidgets

wxWidgets [14] has been collected in the FreeBSD ports tree, which is compiled with GTK 2, but wxWidgets can work directly with X11. The port wxWidgets with X11 is called wxX11. Although wxX11 is incomplete and buggy so far, it can well support picture file handling and basic sub-window management which is required for playing advertising contents.

For my product, wxWidgets 2.8.x can be configured as:

```
./configure --with-x11 --with-opengl\
--disable-shared --enable-unicode
```

After compiling wxX11 (GNU make is required), the script `wx-config` can be used to get C++ compiling and linking parameters for the header and library files of wxX11. The commands look like:

```
# To compile C++ source files
~/wxWidgets-2.8.12/wx-config --cxxflags
# To link object files
~/wxWidgets-2.8.12/wx-config --libs
~/wxWidgets-2.8.12/wx-config --gl_libs
```

## Motor Control via RS-232-C Port

To simplify my program, `stty(1)` was used against `/dev/ttyu0.init` to set default baud rate, bit count, and other options before my program opened `/dev/ttyu0`.

The onboard serial port was connected with a printed circuit board (PCB) I designed to set the electric motors’ behaviors: start and stop, rotating direction, and rotating speed. At the same time, some physical quantities (e.g., ambient temperature and angular displacement) captured by sensors on my PCB were sent to FreeBSD via the serial port.

<code>/boot/</code>	FreeBSD kernel and driver modules, loader(8) and configuration files
<code>/libexec/</code>	Run time link-editor, ld-elf.so.1
<code>/bin/</code> , <code>/sbin/</code> , <code>/usr/bin/</code> , <code>/usr/sbin/</code>	Command line utilities (e.g. <code>/bin/sh</code> , <code>/bin/rm</code> , <code>/usr/bin/killall</code> , <code>/sbin/ifconfig</code> and <code>/sbin/mount</code> )
<code>/dev/</code>	An empty directory for devfs(5)
<code>/var/</code> , <code>/tmp/</code>	Multipurpose files (e.g. <code>/var/empty</code> )
<code>/etc/</code>	Initializing scripts and configuration files
<code>/lib/</code> , <code>/usr/lib/</code>	Shared objects
<code>/usr/local/</code>	Xorg
<code>/root/</code>	My own software files

Table 3. The outlined layout of the filesystem in the USB flash drive or SATA SSD.

## Networking

SSH daemon configured with public key authentication was started in system. SSH can supply secure and multipurpose channels for various remote accessing. An `rsh(1)`-like remote command executing via `ssh(1)` simplifies networking into writing some small `/bin/sh` scripts, avoiding the designing of a tiring and buggy transport protocol.

SSH can also act as a secure transport mechanism of `rsync(1)`, which is used to transfer large files.

## A Customized Script `/etc/rc`

After the FreeBSD kernel is started, usually `/sbin/init` is executed, which tries to run `/etc/rc` to do most of the essential initialization [15]. My customized script `/etc/rc` contained initializing steps

<code>bin/</code>	Standalone executable files (e.g. Xorg, <code>xrandr</code> and <code>xkbcomp</code> )
<code>etc/</code>	<code>xorg.conf</code> and configuration files for <code>font-config</code> and <code>pango</code>
<code>lib/</code>	Shared objects
<code>lib/X11/fonts/</code>	Basic fonts
<code>lib/dri/</code>	DRI drivers
<code>lib/pango/</code>	Pango modules
<code>lib/xorg/modules/</code>	Drivers
<code>share/X11/xkb/</code>	Resource files concerning keyboard

Table 4. The outlined file layout of tailored Xorg under `/usr/local/`. I hadn’t integrated any typical X11 toolkit such as GTK into the flash drive.

reduced from a complete FreeBSD base and a complete Xorg distribution. At the end of the script, it started my own program. The main line of the script is:

1. Waiting for the USB device to get ready in several seconds.
2. Finding and checking the storage device in a possible set of `/dev/da0s1a` and `/dev/ada0s1a` and mounting it in read-write mode.
3. Finding and configuring the network interface in a possible set of `re(4)`, `rl(4)`, and `em(4)`. Also setting network routing. The driver `em(4)` is used when debugging the file set in VirtualBox.
4. Adjusting the time zone setting of the kernel by `adjkerntz(8)`.
5. Tuning the sound volume.
6. Starting the SSH daemon.
7. Starting the Xorg server.
8. Starting my own program.

## Proposals

In the source tree of FreeBSD, there is a great thing—`ndis(4)`. It can run the network device driver written for Microsoft Windows inside the FreeBSD kernel.

But the module lacks maintenance. It often doesn't work with up-to-date device drivers for Windows, especially wireless ethernet device drivers.

What's more, `ndis(4)` can only cover quite a

few of the network devices. Actually, in reality, although FreeBSD has a very robust and well-designed kernel, it leaves people close to GNU/Linux because of its shortcomings.

For the future of FreeBSD, my proposal has to do with the device driver:

1. Move the Windows kernel compatibility layer of `ndis(4)` from the FreeBSD kernel to user mode, which means the layer and device driver will run in a process. In user mode, all programs are easier to hack and debug than in kernel. At least most faults in user mode don't cause kernel crash.
2. Extend the coverage of the compatibility layer, which means covering more network device drivers and trying to cover other types of device drivers such as that of the GPU.

## Acknowledgment

Thanks to Liang, Li (also known as Kylie Liang) for her help with this paper. ●

---

**Li, Xiao** is a software and hardware engineer living in China. He runs his tiny company in Beijing. He is an experienced FreeBSD developer. He worked on ports of **LaTeX-CJK** and **Linux compatibility** in 2006. He is enthusiastically involved in the FreeBSD community of China. He is interested in design of printed circuit board and cross-platform software.

## REFERENCES

- [1] "High-speed rail in China," Wikipedia, online: [https://en.wikipedia.org/wiki/High-speed\\_rail\\_in\\_China](https://en.wikipedia.org/wiki/High-speed_rail_in_China).
- [2] "Beijing–Shanghai High-Speed Railway," Wikipedia, online: [https://en.wikipedia.org/wiki/Beijing%E2%80%93Shanghai\\_High-Speed\\_Railway](https://en.wikipedia.org/wiki/Beijing%E2%80%93Shanghai_High-Speed_Railway).
- [3] (In Chinese Only) "Beijing–Shanghai High-speed Railway," Baidu Encyclopedia, online: <http://baike.baidu.com/view/141756.htm>.
- [4] "Harbin-Dalian High-Speed Railway," Wikipedia, online: [https://en.wikipedia.org/wiki/Harbin%E2%80%93Dalian\\_High-Speed\\_Railway](https://en.wikipedia.org/wiki/Harbin%E2%80%93Dalian_High-Speed_Railway).
- [5] (In Chinese Only) "Harbin-Dalian High-speed Railway," Baidu Encyclopedia, online: <http://baike.baidu.com/view/1213823.htm>.
- [6] (In Chinese Only) High-speed Railway Web, online: <http://www.gaotie.cn/>.
- [7] "Technical Product Specification for the Intel® Desktop Board D410PT," Intel, online: <http://www.intel.com/content/www/us/en/support/boards-and-kits/desktop-boards/000021877.html>.
- [8] "Technical Product Specification for Intel Desktop Board D425KT/D425KTW," Intel, online: <http://www.intel.com/content/www/us/en/support/boards-and-kits/desktop-boards/000020992.html>.
- [9] Mega-info Media Co. Ltd., the official website: <http://www.megainfomedia.com/>.
- [10] GA-B75-D3V User's Manual, GIGA-BYTE Technology Co. Ltd., online: [http://download.gigabyte.cn/FileList/Manual/mb-manual\\_ga-b75-d3v\\_v1.2\\_e.pdf](http://download.gigabyte.cn/FileList/Manual/mb-manual_ga-b75-d3v_v1.2_e.pdf).
- [11] Introduction to Intel Core i5-3450, Intel, online: [http://ark.intel.com/products/65511/Intel-Corei5-3450-Processor-6M-Cache-up-to-3\\_50-GHz](http://ark.intel.com/products/65511/Intel-Corei5-3450-Processor-6M-Cache-up-to-3_50-GHz).
- [12] GeForce driver for FreeBSD, NVIDIA, online: <http://www.geforce.com/drivers>.
- [13] Martin Matuška, "mfsBSD", online: <http://mfsbsd.vx.sk/>.
- [14] wxWidgets, a cross-platform C++ GUI library, online: <http://www.wxwidgets.org/>.
- [15] "Kernel Initialization," "FreeBSD Architecture Handbook," online: [https://www.freebsd.org/doc/en\\_US.ISO8859-1/books/arch-handbook/boot-kernel.html](https://www.freebsd.org/doc/en_US.ISO8859-1/books/arch-handbook/boot-kernel.html).
- [16] Xinhua News Agency, China, the official website: <http://www.news.cn/english/>.
- [17] A reworked profile map of China's mainland, referenced from the governmental digital map website "Tianditu" (The Chinese meaning of this name is "Sky-land Map" or "Sky-earth Map"), online: Chinese: <http://map.tianditu.com/> English: <http://en.tianditu.com/>.