

OM-Cube project

V. Hiribarren, N. Marchand, N. Talfer

hiribarr@enseirb.fr - marchann@enseirb.fr - talfer@enseirb.fr

Abstract. The OM-Cube project is composed of several components like a minimal operating system, a multimedia player, a LCD display and an infra-red controller. They should be chosen to fit the hardware of an embedded system. Several other similar projects can provide information on the software that can be chosen.

This paper aims to examine the different available tools to build the OM-Multimedia machine. The main purpose is to explore different ways to build an embedded system that fits the hardware and fulfills the project.

1 A Minimal Operating System

The operating system is the core of the embedded system, and therefore should be chosen with care. Because of its popularity, a Linux based system seems the best choice, but other open systems exist and should be considered. After having elected a system, all unnecessary components may be removed to get a minimal operating system.

1.1 A Linux Operating System

Using a Linux kernel has several advantages. As it's a popular kernel, many drivers and documentation are available. Linux is an open source kernel; therefore it enables anyone to modify its sources and to recompile it. Using Linux in an embedded system requires adapting the kernel to the hardware and to the system needs. A simple method for building a Linux embedded system is to create a partition on a development host and to mount it on a temporary mount point. This partition is filled as one goes along and then, the final distribution is put on the target host [Fich02] [LFS]. The structure of an embedded Linux is made up of four main components: a kernel, a C-library, a boot loader and system utilities.

The Kernel. It carries out the essential functions such as tasks scheduling, memory management or hardware-software interfacing thanks to device drivers. It must be tuned according the needs, and operates a well-chosen system file. Several ones exist such as ext2, ext3 or JFFS2. Ext2 is commonly used in Linux distributions, and the ext3 file system is the journalised version of ext2. A journalised file system enables a better security and a faster reboot in case of an inopportune system shutdown. JFFS2 use a garbage collector and has been specifically created for Flash file system, like a Compact Flash card [Fich02].

The C-library. It contains the basic functions used by software. The GNU libc (glibc) is usually used on a Linux distribution, but alternatives libraries have better footprint and performance for an embedded system. The μ Clibc comes from the μ CLinux project, but can be used on its own. It provides most of the same functionality than the glibc. Diet libc has the same objectives than the μ Clibc but has been built from scratch. Substantial differences between diet libc and μ Clibc do not seem to exist [Yagh03].

System Utilities. They are used for special needs of the embedded system. In an embedded system, only few unix commands are needed like *init* or *mount*. The BusyBox Project [BUSY] provides tiny versions of many common unix utilities into a single small executable.

The Boot Loader. It loads the system at the boot. The most commonly used are LILO and Grub, but several lightweight loaders exist for embedded systems, like Syslinux [Sysl].

1.2 Other Systems

Linux distributions are the most used and popular open source operating systems for embedded devices. But other open operating systems exist, so a look on them should be taken in order to know if they can be better than Linux in this project. Two other kernels with their tools seem quite interesting and come from the Berkley Software Distribution family. They are under the BSD license which allows the user more freedom than with the GPL license. Unfortunately, there is a lack of drivers for BSD systems as they are quite less popular than Linux.

NetBSD. This operating system contains not only a kernel, but user land utilities too. As a consequence, user land tools are in better interaction with the kernel than a GNU/Linux system. It is designed to work on more than fifty different processor architectures [BSD01], so it could be attractive for embedded systems. Moreover, tools to create crunched binaries like BusyBox or a bootable image file are provided [BSD02].

FreeBSD / PicoBSD. FreeBSD is another full operating system, but unlike NetBSD it is designed to work preferentially on a i386 architecture [BSD03]. A recent survey shows similar performances between FreeBSD-5.1 and Linux 2.6 [BSD04]. As in NetBSD, it is possible to make an embedded system by building FreeBSD from scratch. Furthermore, the PicoBSD project provides some tools to build very small FreeBSD systems [BSD05].

2 Some Multimedia Tools

2.1 Graphic Layers

As a multimedia machine is going to be built, the ability to display graphics is important. Usual multimedia software do not communicate directly with the hardware but use a graphic interface layer instead. Here are some of these layers which could be interesting for a multimedia embedded system.

X Window System. A graphic layer usually chosen is the Xfree86 Window System. It provides a graphic application interface in the user mode. Lots of graphic device drivers are designed for X and quite all unix graphic software rely on it. However this layer uses too many processor and memory resources, so this choice may not be reliable for an embedded Linux system [Fich02].

Framebuffer. A way to avoid the use of X would be to enable the framebuffer. This unix abstraction provides a display management through the kernel, so it should free resources. However, there are two limitations. First, very few software use the Framebuffer. Then, very few graphic cards are supported. If the card is VBE 2.0 compliant, the graphic card VESA BIOS Extension allows the user to bypass this problem with a VESA framebuffer driver. Quite all graphic cards are VBE compliant but using these BIOS disables the card accelerated mode [Fich02]. Nevertheless, Trident cards seem to be supported by the Linux kernel, so as a Trident chipset is currently used, better performances could be expected [Kern].

Xkdrive and DirectFB: Xkdrive is a part of the Xfree86 project [XF01]. While it relies on the VESA mode or on the Linux framebuffer only, it provides the same API than the X Window System. So most X graphic software should work on it. Other projects to build a graphic layer abstraction are in progress, like DirectFB [DF01]. The Mplayer multimedia player can use DirectFB, and this layer runs on other systems like the *BSD ones.

2.2 Multimedia Software

Here are some popular unix multimedia players which could be used in the OM-Cube project.

MPlayer. This software can play most of the popular multimedia file formats [MPlay]. A great feature of MPlayer is the wide range of supported output drivers. It works with lots of graphic layers like XFree86, the framebuffer, OpenGL or DirectFB. MPlayer has an onscreen display (OSD) for status information, nice big anti-aliased shaded subtitles and visual feedback for keyboard controls. DVD subtitles are supported.

Xine. This is another popular multimedia player [Xine]. The features are quite similar to MPlayer, except that Xine does not support the DirectFB layer. Tests should be carried out in order to choose the least resources demanding one between Xine and MPlayer.

Freevo. It is an open-source digital video jukebox based on Linux in addition to a number of open-source audio/video tools [Freev]. MPlayer is used by Freevo to play audio and video files. Freevo can be used both for a standalone personal video recorder computer with a TV and remote control, as well as on a regular desktop computer using the monitor and keyboard. According to the documentation, either X or the framebuffer is supported.

3 Hardware Components

3.1 IR

LIRC is a package which allows the system to decode and to send infra-red signals of many commonly used remote controls [LIRC]. The most important part of LIRC is the lircd daemon that decodes IR signals received by the device drivers and that provide the information. The second daemon program connects to lircd and translates the decoded IR signals to control user-land applications through an IR remote control. Moreover a project has been carried out last year by T2 students in order to implement an IPAQ as a remote control. This project could be an interesting basis to configure a remote control for the OM-Cube.

3.2 LCD Display

A LCD screen to display information about the current multimedia stream should be used for the OM-Cube Project. Some libraries like LCDmod [LCD01] provide software interface to drive the LCD display, and schemes on the Internet show how to use a parallel port to connect it [LCD02]. If the display is not ready to be used, it is possible to build an emulated interface to continue the project [Fich02].

4 Similar Projects

Some similar projects have been carried out. Since the entire operating system is loaded into RAM from a bootable CD-ROM, neither hard disk nor Compact Flash Card is needed. They can provide us reliable sources of information. The key benefit to begin a similar project is that it offers better control of the system without relying on someone else's Linux implementation.

These projects are Linux distributions running with MPlayer and several other open source software. Among these projects, there are GeeXboX, MoviX, eMoviX, MoviX², WOMP! [SProj]. They use the Linux framebuffer or the X layer. Several components are recurrent, like the Syslinux boot loader, the BusyBox tools, the LIRC library or ALSA sound drivers.

References

- [Kern] The Linux kernel web site, <http://www.kernel.org/>
- [BUSY] The BusyBox Project web site, <http://www.busybox.net>
- [Fich02] Pierre Ficheux, *Linux embarqué*, Eyrolles, 2002.
- [Sysl] Syslinux web site, <http://syslinux.zytor.com/>
- [LFS] Linux From Scratch, <http://www.linuxfromscratch.org/>
- [Yagh03] Karim Yaghmour, *Building Embedded Linux Systems*, O'Reilly, 2003, pp. 134-141
- [BSD01] NetBSD web site, <http://www.netbsd.org>
- [BSD02] Brian Rose, Building tiny systems with embedded NetBSD, <http://www.bsdnewsletter.com/2003/09/Features102.html>
- [BSD03] FreeBSD web site, <http://www.freebsd.org>
- [BSD04] Unix scalability benchmarks, Felix von Leitner, <http://bulk.fefe.de/scalability/>
- [BSD05] Small FreeBSD and PicoBSD, <http://people.freebsd.org/~picobsd/>
- [XF01] XFree86 project, Xkdrive, <http://www.xfree86.org/current/Xkdrive.1.html>
- [DF01] DirectFB web site, <http://www.directfb.org/>
- [MPlay] MPlayer, the Movie Player for Linux - official site : <http://www.mplayerhq.hu>
- [Freev] Freevo - official web site, <http://freevo.sourceforge.net>
- [Xine] Xine - official web site, <http://xinehq.de>
- [LIRC] LIRC web site, <http://www.lirc.org/>
- [LCD01] LCD-MOD web site, <http://lcd-mod.sourceforge.net/>
- [LCD02] Displaying Real Time System information on a LCD Display, <http://www.linuxgazette.com/issue77/taneja.html>
- [SProj] Womp!, <http://womp.sourceforge.net/>
GeeXboX, <http://www.geebox.org/en/index.html>
MoviX, <http://movix.sourceforge.net/>