

# Projet OM-Cube : construire un lecteur multimédia sous Linux embarqué



Patrice KADIONIK,  
[kadionik@enseirb.fr](mailto:kadionik@enseirb.fr)

*Mots clés : lecteur multimédia, système embarqué, Linux embarqué*

## 1. LE PROJET OM-CUBE

OM-Cube est l'acronyme de *Open MultiMedia Machine* (OMMM ou OM3). Ce projet consiste en la réalisation d'un lecteur de salon multimédia permettant de visualiser sur sa télévision tout type de flux multimédia : audio (MP3...), vidéo (VCD, SVCD, DVD, DivX, Xvid...), photo (JPEG...). Le contrôle se fait par une télécommande avec un affichage supplémentaire sur un afficheur LCD d'informations concernant le flux multimédia en cours de lecture.

OM-Cube est un système embarqué autonome fonctionnant sous Linux embarqué en mémoire FLASH. C'est en fait un exemple pratique de mise en œuvre de Linux embarqué.

Les motivations initiales de l'auteur sont apparues lors de la lecture du maintenant célèbre livre "Linux embarqué" de P. Ficheux (amitiés Pierre !) qui décrivait l'utilisation de Linux embarqué pour réaliser un lecteur MP3. De plus, à l'époque, regarder un flux vidéo de type MPEG4 (DivX...) sur sa télévision nécessitait de sortir l'artillerie lourde : portable, carte vidéo avec sortie TV... Les lecteurs DVD de salon compatibles DivX par exemple, n'étaient pas légions et onéreux. L'auteur, enseignant-chercheur à l'ENSEIRB, propose des cours et TP autour des systèmes embarqués, Linux embarqué, Temps Réel... OM-Cube est très vite apparu comme un bon sujet de projets pour les étudiants de l'école.

OM-Cube est avant tout un projet d'intégration de système :

- Intégration matérielle : l'utilisation d'une carte à processeur x86 de type "tout en un" (*barebone*) et de développements matériels spécifiques pour l'affichage LCD, la réception infrarouge IR...
- Intégration logicielle : création d'un système Linux embarqué, lecteur multimédia, contrôle de l'afficheur LCD, contrôle du récepteur infrarouge...

Ce projet d'intégration de système met en œuvre un maximum de matériels standards et banalisés couplés à un ensemble de logiciels libres. Cette approche permet d'être un maximum matériel indépendant, les briques logicielles libres imposant aussi les choix matériels. Cette approche a permis d'éviter les longues périodes de mise au point matérielle et logicielle. On a pu vérifier à plusieurs reprises que lors de l'intégration d'un matériel spécifique, cela marchait du premier coup avec la brique logicielle libre choisie en conséquence.

## 2. LES CONTRAINTES MATERIELLES

La plateforme matérielle OM-Cube doit fournir tous les périphériques matériels pour réaliser un lecteur multimédia de salon sachant qu'en plus :

- Elle doit supporter Linux.
- Elle doit aussi servir de plateforme de développement sous Linux standard (distribution Linux Fedora). Cela signifie que l'on met en place un environnement de développement natif sans avoir à installer et configurer un compilateur croisé. Cela suppose l'usage d'un processeur cible x86.
- La puissance du processeur choisi doit permettre une décompression MPEG2 acceptable si l'on n'a pas un circuit électronique de décompression matérielle. On estime qu'un processeur d'au moins 800 MHz est nécessaire pour avoir un débit d'images par seconde correct (25 fps).

Le choix s'est porté sur une carte "tout en un" de type SBC (*Single Board Computer*) comme plateforme OM-Cube.

Elle doit posséder les périphériques suivants :

- Sortie TV.
- Entrée infrarouge (IR) pour un contrôle par télécommande.
- Mémoire Compact Flash ou Disk On Chip pour embarquer l'application et possibilité de boot.
- Port IDE pour connecter un lecteur de DVD et un disque dur pour le développement.
- Port Ethernet pour le développement et pour de futures extensions (lecture de fichiers multimédia par le réseau...).
- Connecteurs clavier et souris pour le développement.
- Port parallèle ou série pour pouvoir connecter un afficheur LCD et afficher en temps réel des informations sur le fichier multimédia en cours de lecture.

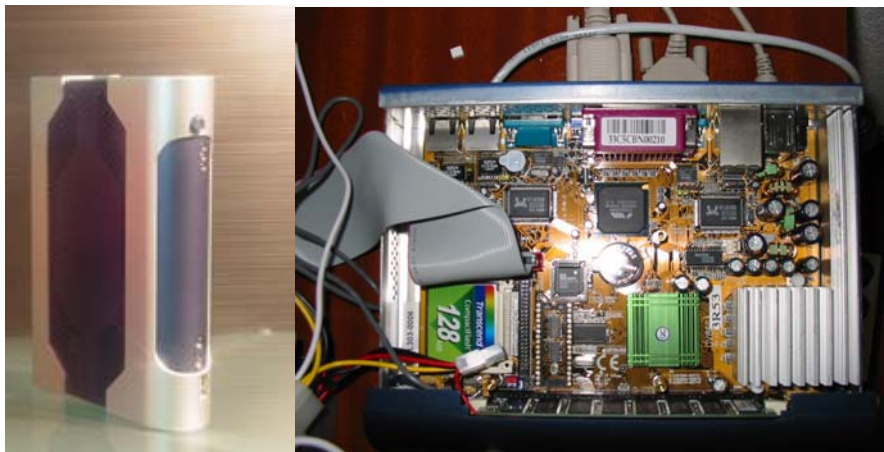
Les cartes SBC choisies proviennent de Lex System revendues par Silink en France (400 € HT). Le premier modèle choisi originellement il y a un an était le CV860A.

Les caractéristiques sont les suivantes :

- Processeur VIA Eden 533 MHz. Consommation de 5 W. Alimentation 12V.
- Bus FSB 133MHz.

- Chipsets : VIA VT8601A et VT82C686B. VGA : Trident VT8601A, frame buffer de 8 Mo. Son : Realtek ALC 201A AC97.
- Réseau : 3 connecteurs 10/100 Mb/s.
- Mémoire PC 133/100 SDRAM DIMM jusqu'à 512 Mo.
- Ports IDE et 2.5" Slim HDD.
- Un connecteur Compact Flash.
- Un support DOC (*Disk On Chip*).
- 3 ports USB 1.1.
- Connecteurs PS/2 clavier et souris.
- BIOS avec boot sur USB et Compact Flash.

Les contraintes matérielles sont à peu près respectées sauf pour la sortie vidéo et la puissance du processeur. Lex System vient de sortir un modèle possédant une sortie TV avec un processeur 800 MHz : le modèle CV860B (sortie TV-OUT en utilisant le circuit VIA VT1621 supporté sous Linux jusqu'en 800x600). Le prototype du projet OM-Cube a été initialement bâti autour du modèle CV860A mais est actuellement en cours de migration vers le modèle CV860B.



Modèles Lex System Light Case CV860A et CV860B

Une deuxième plateforme OM-Cube a été aussi choisie en s'appuyant sur un *barebone* grand public plus classique : le modèle Pundit d'ASUS. Les contraintes matérielles sont aussi respectées sauf pour le boot à partir d'une mémoire Compact Flash, ce qui oblige ici à utiliser soit une clé USB comme mémoire de masse ou un disque interne avec un boot USB ou disque dur...

### 3. LES CONTRAINTES LOGICIELLES

Les choix matériels imposent les choix logiciels en général, mais dans le cas de Linux, les choix logiciels imposent aussi des choix matériels si l'on veut que ces derniers soient supportés par Linux sans tomber dans le piège du développement de drivers spécifiques, ce qui n'est pas le but ici dans le cas d'un projet d'intégration de système.

Un maximum de projets libres doit être réutilisé.

Les contraintes logicielles sont les suivantes :

- La plateforme OM-Cube est sous Linux version 2.4.x ou supérieure. Elle démarre depuis la carte mémoire Compact Flash. On n'a pas besoin de disque dur.
- Un écran de type *splash screen* apparaît au démarrage pour remplacer les traces de boot classiques de Linux.
- Un environnement graphique minimal doit être utilisé : mode *Frame Buffer* ou X11.
- Un lecteur multimédia.
- Un gestionnaire d'affichage sur écran LCD.
- Un gestionnaire de récepteur IR.
- La sortie vidéo est soit la sortie VGA, soit la sortie TV-OUT.

Cela nous donne en fait dans l'ordre les projets libres sélectionnés suivants :

- Distribution Fedora : <http://fedora.redhat.com/>
- Noyau Linux : <http://www.linux.org/>
- Lecteur multimédia : projet Mplayer : <http://www.mplayerhq.hu>
- *Splash screen* : projet boot splash : <http://www.bootsplash.org>
- Afficheur LCD : projet LCDProc : <http://lcdproc.omnipotent.net>
- Contrôleur IR : projet LIRC : <http://www.lirc.org>

La boîte à outils "tout en un" BusyBox (projet BusyBox : [www.busybox.net](http://www.busybox.net)) a été utilisée pour avoir les commandes Linux de base (en particulier `insmod...`) afin de limiter l'empreinte mémoire du projet OM-Cube en mémoire Compact Flash.

## 4. LA PLATEFORME OM-CUBE. DEVELOPPEMENTS MATERIELS SPECIFIQUES

La plateforme OM-Cube se compose de la carte SBC CV860A remplacée par le modèle CV860B et complétée avec des développements matériels spécifiques :

- Afficheur LCD 2x16 ou 4x20 connecté sur le port parallèle. Le schéma électronique, les typons, les fichiers Gerber ainsi que le fichier de perçage sont disponibles en ligne (<http://www.enseirb.fr/~kadionik/om-cube/hardware/hardware.html#LCD>).
- Récepteur IR en cours d'étude.



Plateforme de développement OM-Cube



Afficheur LCD du projet OM-Cube

## 5. DEVELOPPEMENTS LOGICIELS

La première étape a été d'installer l'environnement de développement natif. La méthodologie de mise au point est de tout développer sur le disque dur connecté à la plateforme OM-Cube, de construire une distribution Linux embarqué maison optimisée contenant tous les logiciels nécessaires puis de la recopier en mémoire Compact Flash. Il ne reste plus ensuite qu'à "booter" sur la mémoire Compact Flash pour avoir le résultat final.

Les points essentiels à noter sont :

- La construction d'un Linux embarqué minimal et de son système de fichiers root. La méthodologie à appliquer a déjà été présentée par Pierre Ficheux dans le numéro 17 de Linux Magazine et détaillée dans son livre. Les commandes Linux de base sont fournies par la boîte à outils BusyBox.
- Le noyau Linux (version 2.4.24) a été configuré et optimisé pour les cartes cibles CV860A et CV860B. Le mode *Frame Buffer* VESA a été validé.
- L'écran de *boot splash* a été intégré au noyau et est configuré via le fichier `/etc/lilo.conf`.
- Le fichier `/etc/lilo.conf` est configuré pour valider le mode *Frame Buffer* et lancer l'écran de *boot splash*.
- Le démon Linux LCDProc est lancé pour gérer l'afficheur LCD connecté sur le port parallèle de la plateforme OM-Cube. La mise en œuvre a été présentée par Denis Bodor dans le numéro 60 de Linux Magazine.
- Le logiciel Mplayer a été modifié pour fournir à un client LCDProc les informations (durée, piste...) concernant le fichier multimédia en cours de lecture qui les passera ensuite au serveur LCDProc pour affichage.



Ecran de *boot splash* au boot de la plateforme OM-Cube



Lecture d'un DVD avec la plateforme OM-Cube. Affichage sur écran LCD



Mode *Frame Buffer* de la plateforme OM-Cube

Les composants logiciels du projet OM-Cube sont disponibles en ligne et sous licence GPL. On peut récupérer les sources du projet ainsi qu'un système de fichiers complet à recopier directement en mémoire Compact Flash.

Dans ce dernier cas, si l'on possède une carte cible CV860A ou CV860B, l'installation se fait très simplement comme suit. Il faudra avoir au préalable un disque dur avec un système Linux installé connecté à la carte cible :

1. Création d'une partition Linux sur la mémoire Compact Flash :
  - # fdisk /dev/hdc
2. Création d'un système de fichiers ext3 sur la partition /dev/hdc1 :
  - # mkfs.ext3 /dev/hdc1
3. Décompression du logiciel OM-Cube :
  - # cd
  - # tar -xvzf omcube-x.x.x.tar.gz
4. Montage de la partition /dev/hdc1 :
  - # mkdir /cf
  - # mount -t ext3 /dev/hdc1 /cf
5. Copie du logiciel OM-Cube sous /cf :
  - # cd omcube-x.x.x
  - # cp -r \* /cf
6. Installation de LILO sur la mémoire Compact Flash en correspondance avec le fichier /cf/etc/lilo.conf :
  - # lilo -r /cf -v

Il reste ensuite à rebooter la carte cible et dans le setup du BIOS, choisir de booter depuis le périphérique HD1.

## 6. CONCLUSION

Le projet OM-Cube est un projet en constante amélioration (version actuelle 0.1.0).

Le premier prototype basé sur le modèle CV860A a permis de valider les choix matériels et logiciels. L'utilisation actuelle du modèle CV860B a permis d'atteindre une bonne fluidité des images.

Un certain nombres de points restent à traiter :

- Activation sous Linux de la sortie TV de la carte CV860B. Le circuit VIA VT1621 utilisé est supporté sous Linux par XFree, les drivers étant fournis par VIA. Cela oblige à utiliser un environnement X11 (Xv). Une autre possibilité est d'utiliser le mode *Frame Buffer*. Un projet libre existe alors pour valider la sortie TV-OUT dans ce mode (<http://epiafb.sourceforge.net/>).
- Ajout d'un module récepteur IR connecté sur le port série de la carte CV860B. On utilisera alors le logiciel LIRC qui s'interface directement avec Mplayer. Cela permettra à terme de piloter le lecteur avec une télécommande universelle.
- Optimisation de l'empreinte mémoire . Utilisation de la libc uClibc en lieu et place de la GNU libc standard (projet uClibc : <http://www.uclibc.org/>).

- Mise en coffret de l'ensemble des éléments matériels.

Il faut bien sûr noter que le prix de l'ensemble n'est pas optimisé ici, l'idée première étant de construire son propre lecteur de salon multimédia sous Linux. Il existe sinon des solutions commerciales à moins de 150 €...

L'auteur tient aussi à remercier les étudiants de l'ENSEIRB qui ont contribué au projet OM-Cube : V. Hiribarren, N. Marchand et N. Talfer.

## 7. LIENS

Liens du projet OM-Cube :

- Le projet OM-Cube : [www.enseirb.fr/~kadionik/om-cube/](http://www.enseirb.fr/~kadionik/om-cube/)
- Les développements matériels spécifiques pour OM-Cube : [www.enseirb.fr/~kadionik/om-cube/hardware/hardware.html](http://www.enseirb.fr/~kadionik/om-cube/hardware/hardware.html)
- Distribution Fedora : <http://fedora.redhat.com/>
- Noyau Linux : [www.linux.org](http://www.linux.org)
- Projet Boot Splash : [www.bootsplash.org](http://www.bootsplash.org)
- Projet Busybox : [www.busybox.net](http://www.busybox.net)
- Projet Mplayer : [www.mplayerhq.hu](http://www.mplayerhq.hu)
- Projet LCDProc : [lcdproc.omnipotent.net](http://lcdproc.omnipotent.net)
- Projet LIRC : [www.lirc.org](http://www.lirc.org)
- Projet uClibc : <http://www.uclibc.org/>
- Projet *Frame buffer* epia : <http://epiafb.sourceforge.net/>.

Projets Similaires à OM-Cube :

- Projet Freevo : [freevo.sourceforge.net](http://freevo.sourceforge.net)
- Projet Xine : [xinehq.de](http://xinehq.de)
- Projet OXine : [oxine.sourceforge.net](http://oxine.sourceforge.net)
- Projet LOMB : [lomb.abeille.com](http://lomb.abeille.com)
- Projet OpenBrick : [openbrick.org](http://openbrick.org)

Bibliographie :

- Linux embarqué. P. Ficheux. Editions Eyrolles. 2002.
- Building Embedded Linux Systems. K. Yaghmour. Editions O' Reilly. 2003.