

Android au pays des liseuses

Proposé et encadré par : Ollinger Nicolas

Fontorbe Jordan, Guillaume Arthur, Monediere Tristan,
Rubagotti Joris

Université d'Orléans

23 Mai 2013



- 1 Introduction au domaine
- 2 Description générale du logiciel
- 3 Architectures
- 4 Points techniques
- 5 Exemples de fonctionnement
- 6 Bilan du projet



- 1 Introduction au domaine
 - Technologie E-Ink
 - Liseuse Sony PRS-T1
- 2 Description générale du logiciel
- 3 Architectures
- 4 Points techniques
- 5 Exemples de fonctionnement
- 6 Bilan du projet



Technologie E-Ink

Intérêts

- Faible consommation en énergie
- Idéal pour la lecture

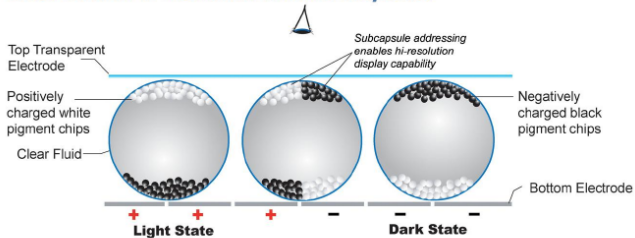
Inconvénient

- Long temps de mise à jour de l'affichage (environ 1s)



Écrans électrophorétiques

Cross-Section of Electronic-Ink Microcapsules



NOTE: Copyright E Ink Corporation, 2002. Image not drawn to scale - for illustration purposes only.



Sony PRS-T1

Caractéristiques principales

- Processeur iMX508
- Écran E-Ink 6 pouces
- Résolution jusqu'à 16 niveaux de gris
- Interface USB
- WiFi
- Mémoire : 2Go (extensible par microSD)



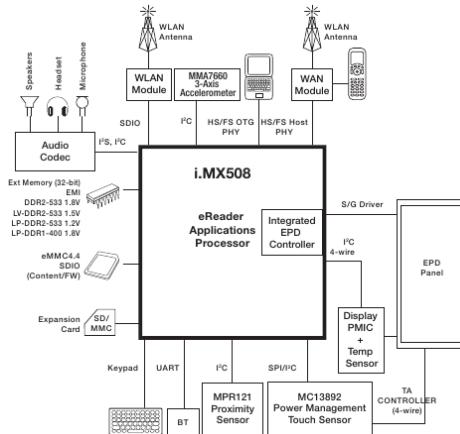
Processeur iMX508

iMX508

- Développé par Freescale
- Architecture ARM Cortex A8
- Faible consommation d'énergie
- Bonnes performances
- Contrôleur d'écran intégré



Architecture du processeur iMX508



Modules

EPDC (Electrophoretic Display Controller)

- Dirige les signaux (waveform)
- Mise à jour partielle ou totale
- Gestion de mises à jour concurrentes

ePXP (enhanced Pixel Pipeline)

- Transparence
- Rotation d'image
- Agrandissement / Réduction d'image



Hack de la liseuse

Mise à jour du firmware

- Nécessite les clés privées de Sony
- Accès total à la liseuse
- Risque d'endommager la liseuse

Mode Recovery

- Nécessite la recompilation du noyau
- Modifications sans risques



- 1 Introduction au domaine
- 2 Description générale du logiciel
- 3 Architectures
- 4 Points techniques
- 5 Exemples de fonctionnement
- 6 Bilan du projet



Cahier des charges

Documentation

Lecture de la documentation

Compréhension des différentes couches matérielles et logicielles

Production d'un document de synthèse (**Mémoire intermédiaire**)

Client RFB+

Ajout du gadget USB au noyau de la liseuse

Connexion via SSH

Ajout du support E-ink à DirectFB



Cahier des charges (suite)

VNC

Création d'un client et d'un serveur VNC
Ajout de VNC à l'émulateur QEMU

QEMU

Utilisation de VNC avec QEMU
Écriture d'applications pour la liseuse

Simulation d'écran

Écriture d'un simulateur d'écran E-Ink



- 1 Introduction au domaine
- 2 Description générale du logiciel
- 3 Architectures**
 - Le système de fichiers
 - Driver EPDC, ioctl et DirectFB
- 4 Points techniques
- 5 Exemples de fonctionnement
- 6 Bilan du projet



Le système de fichiers

Image initial

- Busybox
- Un serveur DHCP
- Un daemon telnet
- Un accès au port USB en mode série



Le système de fichiers

Image final

- Un accès au port USB par connexion Ethernet
- Le support du protocole SSH
- La librairie DirectFB



Driver EPDC et ioctl

Définition ioctl

- Appel système pour des opérations d'entrée/sortie
- Prend en paramètre un code requête

Les fonctions ioctl du driver permettent :

- Mettre à jour l'affichage de l'écran
- Récupérer des informations relatives au driver
- Modifier des paramètres du driver



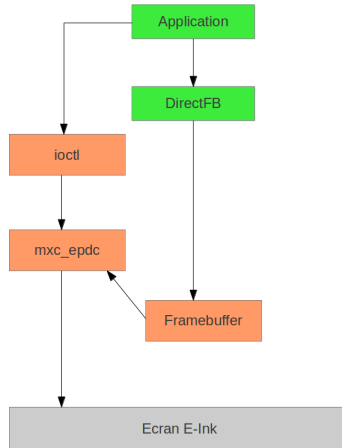
DirectFB

Intérêts

- Ensemble d'API graphiques
- Interaction directe avec le framebuffer
- Aucune modification du kernel
- Aucune dépendance (sauf libc mais déjà présent)



Schéma fonctionnement DirectFB



- 1 Introduction au domaine
- 2 Description générale du logiciel
- 3 Architectures
- 4 Points techniques**
 - La cross-compilation
 - Le driver et les ioctl
- 5 Exemples de fonctionnement
- 6 Bilan du projet



Multiple plateformes

- Développement sur plusieurs systèmes d'exploitation
- Choix du compilateur difficile

Scratchbox

- Environnement de cross-compilation
- Machine virtuelle reproduisant une plate-forme ARM
- Simplification des commandes de compilation



Les ioctl

Les commandes ioctl

- MXCFB_SEND_UPDATE
 - la structure `mxcfb_update_data`
 - `update_region`
 - `waveform_mode`
 - `update_mode`
 - `update_marker`
 - `temp`
 - `flags`
 - `alt_buffer_data`



Les ioctl

Les commandes ioctl

- `MXCFB_WAIT_FOR_UPDATE_COMPLETE`
 - synchronisation de la mise à jour définie par l'update marker
- `MXCFB_SET_AUTO_UPDATE_MODE`
 - active les `deferred_io`
 - dépend de l'option
`CONFIG_FB_MXC_EINK_AUTO_UPDATE_MODE`



DirectFB

Les primitives

- Traçage d'éléments géométriques (rectangle, triangle)
- Chargement d'image

Interaction avec le framebuffer

- Sauvegarde des modifications
- Appel à *Flip()*



- 1 Introduction au domaine
- 2 Description générale du logiciel
- 3 Architectures
- 4 Points techniques
- 5 Exemples de fonctionnement**
- 6 Bilan du projet



Affichage via DirectFB

Différence d'affichage entre les waveforms

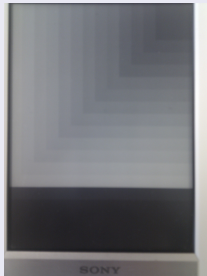


- DU ou A2
 - seulement 2 niveaux de gris
 - temps de rafraîchissement :
 - mesuré : 126ms (A2) 280ms (DU)
 - annoncé : 300ms



Affichage via DirectFB

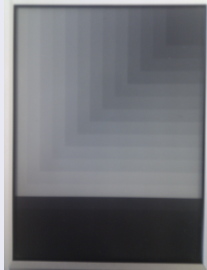
Différence d'affichage entre les waveforms



- GC4
 - 4 niveaux de gris
 - temps de rafraîchissement :
 - mesuré : 610ms
 - annoncé : 600ms

Affichage via DirectFB

Différence d'affichage entre les waveformes



- GC8 ou GC16
 - 16 niveaux de gris
 - temps de rafraîchissement :
 - mesuré : 610ms
 - annoncé : 900ms



- 1 Introduction au domaine
- 2 Description générale du logiciel
- 3 Architectures
- 4 Points techniques
- 5 Exemples de fonctionnement
- 6 Bilan du projet
 - État actuel du projet
 - Suite du projet



État actuel du projet

État d'avancement

- Connexion SSH sur la liseuse depuis un PC hôte :
 - via le port USB
 - émulation d'une connexion Ethernet
- Modification de l'affichage de l'écran
 - programme utilisant DirectFB
 - mise à jour de l'affichage via ioctl



Suite du projet

VNC

- Client sur le PC hôte
- Serveur sur la liseuse
- Traitements côté client
- Modification du protocole RFB

QEMU

- Ajouter l'option VNC à QEMU

