



Retour d'expérience GNU/Linux sur plateforme Arm Cortex A8

Nicolas Aguirre
Octobre 2012

- 
- Cahier des charges
 - Prototypage
 - Benchmarks
 - OpenEmbedded
 - Difficultés rencontrées
 - Points positifs



Cahier des charges

- Etude et réalisation matérielle d'un « panel PC » à base d'ARM Cortex A8.





Cahier des charges

- Distribution GNU/LINUX
- Boot < 1min
- Java
- Adobe Flash
- Classes Java four faciliter l'accès au hardware
(LEDs, numéro de série, CAN, Tension Pile ...)
- Performances CPU/RAM (basée sur des résultats de bench)
- Performances CAN
- Boot + Rootfs sur



Prototypage

- Carte beagle board
 - Excellent support de la part de la communauté.
 - Supportée par la plus part des distributions linux.
 - + ou – supportée par Texas Instrument.
- Essais/evaluation de différentes distributions linux
 - Distros classiques :
 - Ubuntu 10.04 et 11.10.
 - Meego
 - Arch linux, Fedora
 - Distros et environnement de compilation :
 - Arago (Distribution fournie par TI), Openembedded, Angstrom, Poky.
 - Buildroot
 - Openbricks



Benchmarks

- Expedite
- Nbench (BYTEmark benchmark program)
- Scimark2 (Bench Java)
- Bench CAN :
 - Utilisation d'applications « maison »
 - Utilisation du module CAN-USB/2 de ESD electronics.
- NetPerf pour le réseau
- Dd pour nand, disque dur, eeprom ...



BSP

- Développement du BSP avec un partenaire (Eukrea)
 - Linux 3.2.39 upstream + serie de patchs spécifiques pour support de la carte Actia.
 - U-boot version 2012.04 + serie de patchs spécifiques pour support de la carte Actia.
- Développement des couches basses :
 - CAN :MCP2515, utilisation du driver linux et de la couche libsocket CAN.
 - Développement de classes JAVA pour abstraction des fonctions matérielles.
 - Portage d'applications Win32 et WinCE utilisant QT.



OpenEmebedded

- Choix final d'OpenEmbedded et d'Angstrom.
- Distribution / environnement de compilation basé sur les sources.
- Paquets pour tous les logiciels.
- Support de la beagleboard et des processeurs OMAP3 TI.
- Paquets pour chromium et Java 6.
- Documentation (via le projet yocto et la Linux Foundation).
- Support d'autres architectures utilisées chez Actia : Intel (cedar view, ...) Freescale iMx 5/6.
- Génération de SDK contenant toolchain et bibliothéques.
- Séparation des métiers : intégrateur / développeurs.
- S'inscrit dans le cadre qualité global logiciel.
- S'intègre facilement dans des IDE type eclipse.
- Debug et profiling avec eclipse.
- Outils proches de ceux de Wince CE 6.0.



Difficultés rencontrées

- Java
 - Version ARM de OpenJDK fonctionnelle, performante mais bugguée
 - Version Oracle de Java 6 SE pour arm, non performante (Bench scimark2 10fois moins performant que OpenJDK)
 - Build de openjdk pas toujours reproductible.
- Noyau linux, instabilités de certains modules, générant des Kernel Panic.
- Firefox : Lourd et lent.
- Chromium 50MB, 4 heures de compilation, mais se lance en moins de 5 secondes.
- BUG sur systemd v44 => Passage a systemd v189 correction du problème



Conclusion et points positifs

- OpenEmbedded dans son ensemble
 - pour la génération des toolchains pour différentes architectures,
 - pour la bibliothèque de logiciels supportés
- Aide et support de la communauté quand on prends la peine d'expliquer le problème. Aide par le biais des liste de diffusions et par IRC.
- Aide et support de la part de TI
- Le Noyau Linux
- LibscoketCAN
- Systemd
- UBI/UBIFS



Merci pour votre attention, des Questions ?

- http://www.openembedded.org/wiki/Main_Page
- <http://www.angstrom-distribution.org/>
- <http://www.yocto.org>
- <http://elinux.org>