

OBJECTIFS

Ce stage «Android Platform: Source Code Building» a pour objectif la maîtrise du déploiement du système Google Android® sur plate-forme matérielle TI OMAP443x.

Cette formation aborde les concepts de construction de l'ensemble de la plate-forme Google Android par l'étude :

- de son architecture et des spécificités de la plate-forme Android/Linux,
- de ses codes sources et de la chaîne de développement,
- des paramétrages et construction du système complet,
- du SDK Android, de l'émulateur et de l'environnement de développement.

PRE-REQUIS

La formation «Android Platform: Source Code Building» est adaptée aux développeurs en informatique et techniciens ayant quelques notions dans le domaine de l'informatique Open Source, confrontés aux problèmes de portage du système Google Android sur une plate-forme matérielle Mobile.

Linux : Chaîne de développement croisé

- Méthodes, outils et chaîne de développement croisé, binutils, glibc, etc...
- Makefile, Compilateur et débogueur GNU
- Mise au point par port série et TCP/IP

Linux Kernel : Présentation

- Vue d'ensemble du système et rôle du noyau
- Historique, numérotation des versions
- Architectures matérielles supportées
- Support processeurs et File system
- Kernel 2.4, 2.6 et systèmes embarqués

Linux Kernel : Configuration et compilation du noyau

- Structure des sources et modules du noyau
- Optimisation - Patches low-latency, préemptifs
- Fichiers utilisés par les outils de configuration
- Application de patches et utilisation de BSP
- Configuration et compilation du noyau

Travaux pratiques

- Installation et configuration d'une chaîne de développement croisé pour cible Cortex A9 i.MX 6
- Compilation de noyau optimisé pour le cross développement sur cible Cortex A9 i.MX 6

1.

Linux Module : Développement de pilotes

- Pilotes de périphériques sous Linux
- Contraintes de programmation et API Kernel Mode
- Chargement, déchargement de modules
- Un module simple
- Accès aux registres d'E/S et à la mémoire
- Gestion de la mémoire en kernel mode
- Pilotes de type caractère

Linux Module : Installation et paramétrage de pilotes

- Paramètres de chargement de modules
- Systèmes de fichiers sysfs et entrées procfs
- Dépendances entre modules
- Intégration de module propriétaire dans la chaîne de configuration et de compilation des sources officielles

Travaux pratiques

- Compilation d'un noyau instrumenté pour le débogage de module
- Ajout de sources d'un module à l'arborescence du noyau
- Création de patches Kernel

2.

Linux Module : Services et configurations pour la mise au point en Kernel Mode

- Console série
- LTT : Linux Trace Toolkit
- Débogueur Kernel Mode : KGDB

Linux Driver : Développement avancé de pilotes sous Linux

- Mise en sommeil, interruptions, mmap, DMA
- Fichiers de périphériques dynamiques avec udevs

Linux Driver : Périphérique de type caractères particuliers

- Driver de port série et support console série...

Travaux pratiques

- Création et installation d'un pilote de périphérique PC104 de type caractère sur cible GEODE x86
- Traitement d'interruption matérielle en Kernel mode sur port E/S
- Mise en oeuvre de débogage GDB / JTAG Abatron BDI3000 sur plate-forme i.MX 6
- Mise en oeuvre débogage avec KGDB via port série

3.

Linux Driver : Hardware detection et classe pilotes de périphériques industriels

- Extensions ISA/PC104, PCMCIA, bus de terrain, I²C, CAN,
- Cas particulier du PCI...

Linux Driver : Développement avancé de pilotes (suite)

- Architecture des pilotes de périphériques de type bloc
- Architecture des pilotes de périphériques de type réseau
- Frame Buffer vidéo

Linux Driver : Etude de périphériques de type USB

- Standard USB et support Linux Host et Device
- USB Core - Architecture des pilotes USB sous Linux
- Descripteur et classe USB device sous Linux : HID, CDC, Masse storage ...

Travaux pratiques

- Création et installation d'un pilote de périphérique de type bloc
- Accès Direct Frame Buffer
- Mise en oeuvre de drivers USB sous Linux avec analyseur Lecroy

4.