

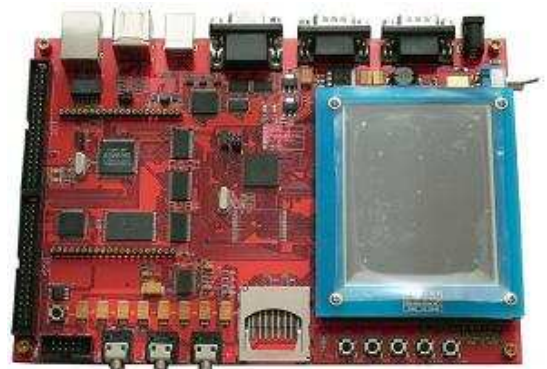
Linux ARM - Faq

Pragmatec

Produits et services dédiés aux systèmes embarqués

Linux ARM

Frequently asked questions



20 Décembre 2008 – Rev 1.00



www.pragmatec.net

Linux ARM - Faq

Linux ARM - Faq

Les kits de développement ARM7 et ARM9 sont des kits réalisés par la société PRAGMATEC S.A.R.L., société située en région parisienne (www.pragmatec.net). Ils sont basés sur des cartes de développement ARM largement utilisée en Asie et aux USA. Il s'agit donc d'un produit efficace, fiable et disponible.

Pragmatec s'est attaché à faire de ces kits un environnement de développement complet et immédiatement opérationnel, avec une introduction en français et le reste des documents et exemples en langue anglaise. En cas de difficultés techniques vous bénéficiez de plus d'un support technique de la part de l'équipe support de Pragmatec sur notre forum www.pragmatux.net.

Ce document a pour but de regrouper l'ensemble des questions types qui nous été posées ces dernières années. Il permettra également de faire des choix techniques à tout ceux qui s'interroge sur l'utilisation de telles architectures, ou qui débute un projet lié à l'informatique embarqué.

Ce document est la propriété de la société PRAGMATEC S.A.R.L. Il ne peut être reproduit et distribué sans l'accord de cette société.

TABLE DES MATIERES

1	<i>Outils et méthode de développement</i>	6
	J'ai l'habitude de Linux sur PC mais je ne connais pas uClinux. Quel est la différence ?	6
	Je développe habituellement sur microcontrôleur PIC18, est-ce que je vais pouvoir réaliser facilement des applications sur processeur ARM ?	6
	Vous proposez des LCD TFT et STN. Quels sont les différences ?	6
	Est-ce que les schémas électriques de vos cartes sont fournis dans les kits de développement ? Peut-on facilement réaliser notre propre carte ARM9 ?	7
	J'ai vu que vous êtes en mesure de réaliser une production en chine. J'imagine que vous ne le faites que pour de gros volumes... Combien de pièces au minimum ?	7
	Comment déboguer une application sous Linux ? Est-ce que vous livrer une sonde JTAG ?	8
	J'ai transféré mon programme sur la cible avec Filezilla et lorsque je le lance j'ai un message m'indiquant un problème de BIN_FLAT. Quel est le problème ?	8
	Il y a quelques temps j'ai réaliser un programme sur PC linux. Puis-je simplement le recompiler et le faire fonctionner sur la cible ARM9 ?	8
	J'utilise une station sous Linux pour compiler le noyau de ma carte ARM9 mais je n'arrive pas à lancer « make xconfig »	8
	Si je réalise un driver, je suis obligé de reconstruire tout le kernel et le charger à nouveau. N'y a-t-il pas un moyen plus rapide ?	9
	Je tente de reconstruire le kernel sur mon PC linux, mais dès le make xconfig j'obtiens une erreur. Pourquoi ?	9
	ARM7 – ARM9 : quels sont les avantages ?	9
	Je réaliser une application qui nécessite de rentrer très souvent des nombres. L'écran tactile me semble trop fastidieux, alors peut-on brancher un clavier PS2 ?	10
2	<i>Hardware et périphériques</i>	11
	Lorsque je fais un accès à une adresse mémoire sur ma carte DEV2410, j'obtiens une erreur « sig segv » et mon application s'arrête. Pourquoi ?	11
	Nous souhaitons utiliser un périphérique I2C sur linux 2.6.25 mais le driver n'existe pas. Faut-il écrire un driver ? Peut-on s'inspirer d'un modèle ?	11
	Mon application a besoin d'effectuer des acquisitions 12 bits à 50KHz de façon très précise. Comment faire avec vos cartes Linux ?	11
	La COM0 de ma carte de développement DEV2410 ne marche pas. Pourtant je parviens à piloter correctement ma COM1. La carte est-elle en panne ?	12
	J'insère une clef USB, elle semble vue par Linux mais rien sous /dev. Au montage de /dev/sda1, le noyau me dit que le périphérique n'existe pas !?	12
	Le contrôleur Ethernet de la carte 2410SODIMM est un 10/100T. Peut-il s'adapter au réseau détecté (croisé ou non) ?	12

Linux ARM - Faq

J'ai besoin de 4 ports RS232 et la carte n'en possède que 3. Est-ce qu'on peut connecter un UART supplémentaire ?	13
Est-ce que vos cartes ARM9 peuvent permettre de réaliser un player MP3 ?	13
Mon application est très simple, elle regarde simplement des entrées et pilote des relais en conséquence. Dois-je réaliser des drivers pour juste des entrées TOR et des relais ? (linux 2.6.25 et ARM9).....	13
J'ai branché un scope sur les broches SDA et SCL de la carte DEV2410 et je ne vois pas les signaux bouger. La carte est-elle en panne ?	13
Sur mon PC, ma clef USB est montée automatiquement après insertion, c'est pas le cas sur ma carte DEV2410. Pourquoi ?	13
3 Applications et paramétrage	15
Comment activer le Watchdog sur Linux 2.6 ?	15
Quelle est la limite en taille des SDCard utilisée ?	15
Notre société a acheté les 2 cartes DEV2410 et 2410SODIMM et à notre surprise elles n'ont pas le même BIOS. Pourquoi ?	15
En combien de temps une carte ARM9 peut démarrer et lancer une application automatiquement ? ...	16
J'ai remarqué que la commande « mdev -s » lancée au démarrage prend beaucoup de temps. Est-on obligé de la lancer ?	16
Comment lancer des drivers et mon application automatiquement au démarrage ?	16
Y a-t-il un serveur web pour la carte DEV2410 ?	17
Est-il possible de faire fonctionner les CPU S3C2410 à plus de 200MHz comme c'est le cas sur vos cartes ?.....	17
Peut-on diminuer la taille des mémoires de la carte SODIMM et espérer ainsi économiser d'autant sur le PU HT du produit ?	17
J'aimerais utiliser une carte 2410SODIMM. Est-il possible de la customiser ? Peut-on changer la taille mémoire ?.....	17
4 Divers	18
La carte SODIMM 2410 est annoncée pour une température commerciale. Est-il possible de la faire fonctionner à des température négative ?	18
La carte 2410SODIMM a-t-elle passé des tests CEM ? Quant est-il pour les vibrations et les chocks ?	18
Quelle est la consommation moyenne d'une carte SODIMM ?	18
Est-ce que vous assurez la formation et le support pour les non linuxiens qui débloquent sur ces CPU ARM et du linux embarqué ?	19
Les composants que vous utilisez ne se trouvent pas chez Radiospares et Farnell. Ou peut-on les trouver ?.....	19
Ma carte consomme trop. Comment réduire la consommation ? Peut-on éteindre des périphériques ?	19



1 Outils et méthode de développement

[J'ai l'habitude de Linux sur PC mais je ne connais pas uClinux. Quel est la différence ?](#)

uClinux est la version de Linux destinée au μ C (microcontrôleur), des puces qui incluent à la fois un CPU et des périphériques... bref c'est une version taillée pour l'embarqué, là où de nombreux périphériques seront utilisés. Nous proposons 2 types de noyau : uClinux 2.4 et Linux 2.6. uClinux repose sur une gestion mémoire simplifiée, sans MMU et utilisant une seule zone mémoire d'allocation dynamique pour toutes les tâches. En d'autres termes, la gestion mémoire est beaucoup moins sécurisée mais du coup elle est beaucoup plus simple et efficace (en terme de charge CPU). Pour donner un exemple concret, un ARM7 à 60MHz avec uClinux 2.4 ira 2 à 3 fois plus vite sur les accès à une adresse spécifique qu'un i86 AMD à 100MHz avec Linux 2.4.

Du point de vue de l'utilisateur, un système uClinux sera identique ou presque à un système Linux. Il sera plus léger, avec un shell possédant moins de programmes, mais vous bénéficierez des fonctionnalités de base : serveur FTP et TELNET, serveur web, filesystem FAT, EXT2, ...

[Je développe habituellement sur microcontrôleur PIC18, est-ce que je vais pouvoir réaliser facilement des applications sur processeur ARM ?](#)

Si vous choisissez de développer sur ARM7, vous pourrez développer vos applications de façon assez similaire à celle que vous utilisiez pour réaliser vos applications sur PIC18 : l'accès à n'importe quelle adresse du système peut se faire en une ligne de C depuis vos applications. Dans les faits, les développeurs dans votre cas procèdent ainsi, et abordent la notion de driver petit à petit.

Sur ARM9 / Linux 2.6, la mémoire virtuelle vous empêchera de faire ainsi (à l'exception du /dev/mem) et il vous faudra passer par la notion de driver. Ceci constituera la plus grande difficulté car votre code devra être particulièrement architecturé pour tirer profit au mieux d'un système Linux embarqué. N'hésitez pas à demander notre assistance dans ce cas précis, au moment même où vous définirez votre architecture logicielle.

[Vous proposez des LCD TFT et STN. Quels sont les différences ?](#)

Le STN est dit à matrice passive, et le TFT à matrice active. La différence se voit surtout lors des mouvements graphiques par exemple ceux d'une souris à l'écran. Sur celui à matrice passive, il y aura un effet de rémanence, peu agréable à l'œil. Toutefois pour l'affichage de données statiques, le rendu est sensiblement le même. Historiquement, les LCD STN étaient réservés pour les LCD de petites tailles (photocopieur) et les TFT pour les grandes tailles (PC portable). Le temps et la technologie ont permis de fabriquer des TFT de petites tailles à des coûts similaires aux STN, si bien que les LCD finissent aujourd'hui par disparaître.

L'autre différence majeure réside dans l'interface. Sur certains TFT le pilotage de l'interface est assez complexe, et il est nécessaire de les piloter à des fréquences élevées (à plus de 50MHz

Linux ARM - Faq

parfois). C'est pour cela que le CPU ARM7 ne permet de piloter que des STN alors que l'ARM9 peut piloter les 2 types de LCD.

Est-ce que les schémas électriques de vos cartes sont fournis dans les kits de développement ? Peut-on facilement réaliser notre propre carte ARM9 ?

Tous les schémas électriques ainsi que les datasheets des principaux composants sont en effet fournis sur le CDROM de nos kits de développement au format PDF. Vous pouvez donc les utiliser voir les recopier directement pour réaliser votre propre carte, nous n'avons déposé aucun copyright sur les schémas, ils peuvent donc être librement diffusés. Pour autant ils ne vous permettront pas de réaliser « facilement » votre propre carte ARM9, la difficulté majeure d'une telle carte étant le routage avant toute chose : un routage en 6 couches est un minimum, avec comme contrainte de conserver sur tout le bus système une cohérence de longueur de bus. Enfin, les coûts de prototypage d'une telle carte sont relativement élevés en France, entre la programmation du robot de pose automatique et les photographies au rayon X pour contrôler l'assemblage de la carte.

C'est pour cela que nous avons créé une carte modulaire, la SODIMM2410, qui vous permettra de ne réaliser que la carte mère, en 2 couches, sans aucune contrainte de routage particulière. Enfin, pour les clients qui le souhaitent, nous pouvons aussi router et prototyper à faible coût une carte ARM9 custom que nous produirons en Chine par la suite.

J'ai vu que vous êtes en mesure de réaliser une production en chine. J'imagine que vous ne le faites que pour de gros volumes... Combien de pièces au minimum ?

La plupart des sociétés européennes qui produisent en Chine le font dans la région de Shenzhen, via une plate-forme «business » basée à Hong-Kong. Ceci permet, certes, de payer en dollars, mais ne permet pas de s'interfacer directement avec l'usine de production. Pour le faire il faut être sur place en Chine, et engager de gros moyens de contrôle et d'inspection. C'est pour cela que ces sociétés européennes sont avant tout de grosses sociétés disposant de moyens humains et financiers importants, utilisés dans le cadre de projet de production d'envergure. Du coup les sociétés chinoises de cette région sont des sociétés de plusieurs milliers de personnes, qui réalisent avant tout des productions par batch de 3000 pièces minimum.

Pragmatec a eu l'opportunité il y a quelques années de travailler directement avec une société de production à taille humaine, située à Pekin même. Ceci a été rendu possible par le fait qu'un de nos collaborateurs français est marié à une chinoise et parle couramment chinois. Il réside en chine la moitié de l'année ce qui permet de contrôler les productions réalisées. Notre capacité de production est de quelques centaines de pièces par mois, sans minimum de quantité de cartes à produire. Nous pouvons donc produire à la fois les prototypes (essentiels pour tester l'unité de production), les pré-série à des prix très attractifs et les séries par batch de 100 à 300 pièces.

Comment déboguer une application sous Linux ? Est-ce que vous livrez une sonde JTAG ?

Nous livrons en effet une sonde JTAG, mais elle est destinée à la programmation de la carte lorsque celle-ci est vierge (sortie d'usine) ou lorsque vous avez fait une erreur de manipulation qui aurait conduit à effacer malencontreusement le BIOS de la carte.

Pour déboguer une application Linux, vous utiliserez une solution logicielle, le débogueur GDB dans sa version croisée. Le debug peut se faire sous Eclipse pendant la phase de développement. Reportez vous à la documentation du kit de développement pour mettre en place votre solution de développement et de debug.

J'ai transféré mon programme sur la cible avec Filezilla et lorsque je le lance j'ai un message m'indiquant un problème de BIN FLAT. Quel est le problème ?

Vous êtes vraisemblablement sur une carte ARM7 avec le système uClinux 2.4. Les programmes exécutables sont au format « FLAT BINARY » et non pas « ELF ». Il y a de forte chance que vous ayez mal transféré votre fichier, par exemple en paramétrant Filezilla en mode « TRANSFERT AUTO » alors qu'il faudrait le paramétrer en mode « TRANSFERT BINARY ».

Il y a quelques temps j'ai réalisé un programme sur PC linux. Puis-je simplement le recompiler et le faire fonctionner sur la cible ARM9 ?

Oui, mais il faudra faire de même pour vos bibliothèques. C'est un problème propre à Linux et pas à votre cible. Si vous utilisez uniquement la bibliothèque C (libc), alors aucun soucis. Par contre si vous utilisez certaines bibliothèques spécifiques, vérifiez que la version de la bibliothèque soit la même que celle présente sous /lib sur votre cible, sinon vous devrez compiler la bibliothèque pour la cible.

J'utilise une station sous Linux pour compiler le noyau de ma carte ARM9 mais je n'arrive pas à lancer « make xconfig »

Vous pouvez parfois obtenir un message d'erreur au lancement du make xconfig, message qui vous indique qu'il vous manque « qtconf ». En fait, depuis la branche Linux 2.6, l'application graphique « xconfig » a été remplacé par le programme « qtconf », si bien qu'il vous faut les programmes et les bibliothèques qt3-conf sous « /usr/share/qt3 ».

Téléchargez et installez les packages concernant « qt-conf » sur votre station en suivant la procédure habituelle de votre distribution (apt-get install, rpm, ...).

Si je réalise un driver, je suis obligé de reconstruire tout le kernel et le charger à nouveau. N'y a-t-il pas un moyen plus rapide ?

Inutile de reconstruire le noyau dans votre cas. Il est possible de ne demander à la distribution de ne reconstruire que les modules (drivers chargeables dynamiquement). Pour cela tapez la commande « make modules ». Vous n'aurez alors qu'à transférer votre module sur la cible (à l'aide d'un transfert Zmodem, FTP, USB, ...) et à le charger à l'aide de la commande « insmod xxxxx.ko ». Vérifiez que le module est bien chargé en mémoire en utilisant la commande « lsmod ».

Je tente de reconstruire le kernel sur mon PC linux, mais dès le make xconfig j'obtiens une erreur. Pourquoi ?

Il manque vraisemblance un programme sur votre station de développement. La plupart du temps il manque le programme WISH qui vous permettra de construire l'interface graphique xconfig. Ceci n'est pas lié à la cross-compilation d'une plate-forme ARM, car vous auriez le même soucis en reconstruisant un noyau pour votre station de développement. Regardez attentivement les messages affichés, ils vous indiqueront le programme qui vous fait défaut.

ARM7 – ARM9 : quels sont les avantages ?

La différence majeure entre les 2 architectures est l'utilisation d'une MMU ou non. La MMU (Memory Management Unit) est une entité en charge de gérer la mémoire RAM en la morcelant et en interdisant son accès physique. Ainsi une application ne pourra pas manipuler une variable en 0x00001000 en écrivant simplement à cette adresse. Il faudra demander à la MMU qu'elle fournisse une adresse translatée, c'est-à-dire une adresse comme 0x0c003fe0 qui correspond à un bloc mémoire virtuel. Ceci se fait automatiquement lorsque vous écrivez un code C, c'est le noyau qui le fait pour vous. Après tout, l'endroit exact où se situe une variable globale vous importe peu.... Sauf lorsque vous voulez accéder à du hardware placé à une adresse en particulier ! Du coup, pour simplement allumer une LED placée à une certaine adresse, il vous faudra demander à la MMU de vous fournir une adresse virtuelle correspondante.... Mais ceci vous sera refusé depuis l'espace utilisateur, depuis votre programme donc. Il vous faudra écrire un driver (espace kernel) pour y arriver. Ce mécanisme constitue la principe de sécurité de la MMU, à savoir une interdiction d'accéder au hardware depuis l'espace utilisateur, et de ne l'autoriser que pour l'espace kernel, notamment les drivers.

Pour allumer une simple LED, ça devient vite compliquée ! Sur une architecture ARM7, il n'y a pas de MMU, il suffira alors d'une simple ligne en C pour allumer la LED. Par contre, attention de ne pas faire n'importe quoi, car en manipulant le hardware sans sécurité vous risquer de planter tout le système.

Je réaliser une application qui nécessite de rentrer très souvent des nombres. L'écran tactile me semble trop fastidieux, alors peut-on brancher un clavier PS2 ?

Le mieux est encore d'utiliser un « keypad » USB. Connectez le à votre carte, il sera vu comme périphérique HID (Human Interface Device) et sera géré par votre système Linux dans même devoir le reconfigurer.

2 Hardware et périphériques



Lorsque je fais un accès à une adresse mémoire sur ma carte DEV2410, j'obtiens une erreur « sig segv » et mon application s'arrête. Pourquoi ?

Vous n'avez pas le droit de faire un accès absolu à une adresse physique de votre carte. La MMU vous en empêche. Sur ARM9 il vous faut réaliser un driver pour accéder à la mémoire physique. Toutefois pour des accès occasionnels, nous avons implanter l'interface « /devmem » sur les cibles Linux 2.6. Cette interface vous permet d'accéder à une adresse quelconque de la mémoire depuis votre application. Dans le répertoire des programmes d'exemple (userland), vous trouverez un exemple d'application GPIO qui permet d'activer des entrées / sorties TOR depuis votre applicaion.

Nous souhaitons utiliser un périphérique I2C sur linux 2.6.25 mais le driver n'existe pas. Faut-il écrire un driver ? Peut-on s'inspirer d'un modèle ?

Il existe certains drivers I2C dans la distribution courante, et les plus courant sont ceux pour le PCF8574 et le PCF8591. Vous pouvez en effet soit vous inspirer d'un de ces 2 drivers pour créer votre propre driver ou bien réaliser votre propre code d'accès au chip depuis l'espace utilisateur. Cette dernière solution est moins performante mais elle présente l'avantage de fournir une solution concrète à ceux pour qui la réalisation d'un driver sous Linux 2.6 est une difficulté majeure.

Mon application a besoin d'effectuer des acquisitions 12 bits à 50KHz de façon très précise. Comment faire avec vos cartes Linux ?

Les CPU ARM7 et ARM9 de la marque Samsung possèdent des convertisseurs analogiques numériques de seulement 10 bits de résolution. Si vous acceptez de descendre à cette résolution, vous pouvez utiliser le driver ADC fourni par défaut pour effectuer les acquisitions sur les voies sélectionnées. Maintenant, le problème du temps réel se pose : si vous souhaitez effectuer des acquisitions avec une précision relativement importante, il vous faudra affecter un timer matériel au sein même du driver ADC de façon à déclencher une conversion sous IT. Toutefois le noyau Linux n'est pas à proprement parler temps réel, la réactivité du système à entrer sous l'IT timer dépendra de l'activité du noyau au moment l'IT.

L'autre façon de faire est d'associer un petit microcontrôleur comme un PIC18 pour effectuer les acquisitions et faire les calculs en temps réel. A vous ensuite de connecter ce PIC18 au CPU par le bus SPI par exemple, voir via le bus système.

[La COM0 de ma carte de développement DEV2410 ne marche pas. Pourtant je parviens à piloter correctement ma COM1. La carte est-elle en panne ?](#)

Le BIOS de votre carte est probablement programmée avec les paramètres par défaut passés au noyau Linux, à savoir « console=ttySAC0,115200 ». Ceci indique que la console utilisée par Linux pour afficher les informations de démarrage et lancer le shell est redirigée vers la COM0. Ceci étant vous ne pourrez pas l'utiliser comme port de communication.

Utilisez donc le port COM1 ou alors modifiez les paramètres de boot du noyau pour avoir « console=null » indiquant à Linux de ne pas avoir de console. Par la suite vous pourrez toujours ouvrir un shell au travers d'une session réseau TELNET.

[J'insère une clef USB, elle semble vue par Linux mais rien sous /dev. Au montage de /dev/sda1, le noyau me dit que le périphérique n'existe pas !?](#)

Que ce soit sous Linux 2.6 ou uClinux 2.4, on ne peut monter un périphérique que si un « node » existe sous /dev (il s'agit de son interface dans le filesystem) et si le driver adéquate est chargé en mémoire. Si le driver du contrôleur USB est chargé (dynamiquement ou statiquement dans le noyau) ainsi que le driver « USB mass storage » alors vous devriez voir apparaître des messages d'information liés à la clef USB détectée (taille mémoire et interface dans le filesystem).

Ainsi, à la fin de la détection de la clef USB, vous devriez voir sous quelle interface est placée cette clef (sda par exemple). Cela signifie que vous pourrez accéder à la première partition (partition pas défaut) via « /dev/sda1 ». Si cette interface n'existe pas dans le filesystem, il vous faut la créer à l'aide de la commande « mknod » ou bien la créer directement dans le cramfs (rootfs) à la construction de votre système embarqué.

Enfin, le noyau Linux 2.6 que nous proposons pour nos cibles dispose de la commande « mdev » qui permet de créer automatiquement les points d'entrées telles que /dev/sda1 lorsqu'un périphérique est détecté. Commencez donc par insérer la clef USB avant d'utiliser la commande « mdev -s », puis monter votre clef à l'aide de la commande « mount ». Le noyau Linux 2.6 dispose aussi d'un mécanisme « hotplug » qui permet de jouer un script automatiquement, notamment un script de montage de clef USB.

[Le contrôleur Ethernet de la carte 2410SODIMM est un 10/100T. Peut-il s'adapter au réseau détecté \(croisé ou non\) ?](#)

La capacité de croiser ou décroiser un câble s'appelle en fait MDI/MDIX. Sur un réseau on considère en effet que les équipements connectés sont soit en mode ÉMETTEUR (MDI) soit en mode RECEPTEUR (MDIX). Lorsque vous connectez un équipement de type PC (avec contrôleur Ethernet) à un hub Ethernet, vous devez utiliser un câble non croisé (droit) car votre carte PC sera en mode MDI et votre hub Ethernet en en mode MDIX (récepteur).

Le contrôleur Ethernet DM9000 utilisé sur nos cartes SODIMM est un contrôleur 10/100T signifiant qu'il s'adapte automatiquement à un réseau 10 Mbits/sec ou 100 Mbits/sec. De plus il possède une capacité de MDI automatique, c'est-à-dire qu'il bascule automatiquement en mode MDI ou MDIX selon le câble utilisé (droit ou croisé) ou les autres équipements détectés (MDI ou MDIX).

J'ai besoin de 4 ports RS232 et la carte n'en possède que 3. Est-ce qu'on peut connecter un UART supplémentaire ?

Il est toujours possible de connecter un UART supplémentaire sur le bus système de l'ARM, mais ce n'est pas la solution la plus simple. Le mieux est encore d'ajouter un chip FTDI sur la carte, puce qui sera connectée à un des 2 ports USB. L'interface RS232 TTL du FTDI sera connectée au PC au travers d'un MAX232 par exemple. Vous pourrez alors utiliser un 4^{ème} port COM via l'interface /dev/ttyUSB0 (par exemple, le nom de l'interface pouvant être différent selon les systèmes).

Est-ce que vos cartes ARM9 peuvent permettre de réaliser un player MP3 ?

Bien entendu. Nous vous recommandons d'utiliser « madplay » qui permet de jouer un fichier MP3. Il est prévu de le fournir en exemple dans une prochaine version de notre distribution Linux 2.6 à venir.

Mon application est très simple, elle regarde simplement des entrées et pilote des relais en conséquence. Dois-je réaliser des drivers pour juste des entrées TOR et des relais ? (linux 2.6.25 et ARM9)

Je vous recommande d'utiliser l'interface « /dev/mem » sous Linux. Elle permet de s'interfacer avec n'importe quelle adresse physique depuis l'espace utilisateur (chose qui est normalement rendu impossible par la MMU). En fait, derrière cette interface se cache un driver qui va demander à la MMU de fournir une adresse virtuelle qui correspondra à une plage d'adresse physique.

Regardez les sources du programme GPIO. Il s'agit d'un programme simple, écrit en quelques lignes de C, qui permet de paramétrer n'importe quel IO du CPU en entrée ou en sortie et de le piloter.

J'ai branché un scope sur les broches SDA et SCL de la carte DEV2410 et je ne vois pas les signaux bouger. La carte est-elle en panne ?

Attention, la carte de développement DEV2410 ne possède pas les résistances de pull-up qui doivent être impérativement connectés dans le cas d'une utilisation du bus I2C. Typiquement leur valeur est de 4,7Ko, mais cette valeur varie en fonction de la topologie du bus (nombre de périphériques connectés, longueur du bus, vitesse, ...).

Sur mon PC, ma clef USB est montée automatiquement après insertion, c'est pas le cas sur ma carte DEV2410. Pourquoi ?

Avec les dernières releases de Pragmatec sur les cartes ARM9 (telles Linux 2.6.25) il est désormais possible d'utiliser la fonctionnalité dite de « hotplug ». Lorsqu'un périphérique amovible est inséré tel qu'une clef USB ou carte SD, un script est exécuté automatiquement par le noyau. Ce script est désigné sous « /proc/sys/kernel/hotplug ».

Linux ARM - FAQ

Nous avons spécifié d'exécuter le script « hotplug.sh » qui se trouve sous le répertoire /home. Ce script ne permet que l'insertion d'une clef USB sous /home/mnt depuis l'interface /dev/sda1. Il est possible de le compléter pour d'autres insertions (/dev/sda2) voir d'autres périphériques comme /dev/mmcblk0 pour ce qui est de la gestion d'une carte SD.



3 Applications et paramétrage

[Comment activer le Watchdog sur Linux 2.6 ?](#)

Afin d'activer le watchdog vous devez exécuter la commande suivante : « /sbin/watchdog -t 5 /dev/watchdog ». Ceci permet d'activer automatiquement la fonction de watchdog sur la carte (dont un driver est en charge au travers de l'interface « /dev/watchdog ») et d'envoyer un « tick » périodiquement avant que le driver de watchdog en reset la carte. Le timeout programmé est de 5 secondes, ainsi la carte sera automatiquement redémarrée si le CPU est planté.

Afin de simuler cette situation, vous pouvez terminer le processus « /sbin/watchdog » à l'aide de la fonction « kill ». La carte redémarrera 5 secondes après l'arrêt du programme.

[Quelle est la limite en taille des SDcard utilisée ?](#)

Avec la dernière version du noyau Linux 2.6 (Linux 2.6.25 et au delà) nous avons testé la gestion des cartes SD jusqu'à 4Go, mais pas au delà. Quant au noyau uClinux 2.4, les cartes SD de 4Go n'ont pas été testés. Seules les cartes jusqu'à 256 Mo sont gérées correctement. Le problème est d'avantage est un problème lié à la vitesse des cartes SD, et non pas lié à la capacité de la carte. Il convient donc de tester vos cartes pour connaître les types de cartes correctement gérées par votre système embarqué.

[Notre société a acheté les 2 cartes DEV2410 et 2410SODIMM et à notre surprise elles n'ont pas le même BIOS. Pourquoi ?](#)

La carte DEV2410 est en effet proposée avec le BIOS Pragmatec par défaut, la carte SODIMM bénéficiant du bios u-boot 1.1.6 porté pour nos cibles. Les 2 BIOS peuvent toutefois fonctionner indifféremment sur les 2 cartes.

Le BIOS u-boot a comme avantages de pouvoir utiliser une connexion réseau pour faire ses transferts d'images (ulmage pour le noyau) et aussi d'être open-source et compilable avec GCC. En conséquence il est aussi très lent, GCC étant un compilateur générique multi-cœur, et donc par définition pas optimisé pour une architecture en particulier.

Le BIOS Pragmatec, quant à lui, est compilé avec le compilateur ADS de ARM, le résultat est un code plus compact, plus optimal, et donc plus rapide à l'exécution. Même si les sources sont fournies sur le CDROM des kits de développement ARM9, il est nécessaire d'obtenir la suite ADS fort coûteuse pour le reconstruire.

En conclusion, le BIOS u-boot est davantage choisi lorsqu'un besoin de maintenabilité par le client se fait sentir, alors que le BIOS Pragmatec est utilisé lorsque des performances de vitesse sont nécessaires au démarrage de la carte.

[En combien de temps une carte ARM9 peut démarrer et lancer une application automatiquement ?](#)

Dans le pire des cas (au niveau temporel j'entends...) votre carte peut être équipée d'un BIOS u-boot 1.1.6 et d'un noyau Linux 2.6.25. Le BIOS u-boot est un BIOS complet et générique, capable de s'adapter à différentes cibles. Il est compilé avec GCC et permet donc des modifications de la part du client. Hélas, GCC n'est pas un compilateur très optimisé et les performances obtenues au final sont assez médiocres. Au final, le BIOS u-boot permet de lancer le noyau Linux au bout de 10 secondes, et le noyau Linux 2.6.25 lui-même met 10 secondes à lancer une application.

Afin d'optimiser le temps de démarrage, il est possible d'utiliser un BIOS allégé, très simple et compilé avec le compilateur ARM, tel que celui que nous proposons. Il permet de lancer le noyau Linux en moins de 3 secondes dans le meilleur des cas.

Ensuite il est aussi possible de customiser le noyau Linux et de l'alléger afin de n'utiliser que les drivers adéquates pour votre carte. Enfin, il convient d'optimiser les scripts de démarrage, en réduisant par exemple le nombre de partition ou en supprimant l'appel à la commande « mdev -s ».

[J'ai remarqué que la commande « mdev -s » lancée au démarrage prend beaucoup de temps. Est-on obligé de la lancer ?](#)

Avec « udev », l'ensemble des interfaces présentes sous « /dev » reflète une réalité : à chaque driver chargé, on inspecte si le hardware correspondant est bien présent (probe) et seuls les périphériques identifiés sont dotés d'une interface sous « /dev ».

Le rôle de la commande « mdev -s » est de scanner automatiquement l'ensemble des périphériques afin de mettre à jour le répertoire « /dev » avec les interfaces adéquates. Cette commande prend prêt de 2 secondes et il est vrai qu'elle n'est pas indiquée pour un système embarqué où la liste des périphériques utilisés est connu par avance.

Soit vous utilisez la commande « mdev -s » à chaque qu'un nouveau périphérique est inséré (USB, SDcard, I2C, ...), soit vous créez au démarrage toutes les interfaces nécessaires à votre équipements (avec la commande « mknod » à ajouter au fichier « /usr/etc/rc.local »).

[Comment lancer des drivers et mon application automatiquement au démarrage ?](#)

Les cartes ARM7 et ARM9 possèdent toutes 2 un fichier de démarrage : c'est le fichier « /home/rc.sh » sous uClinux 2.4 et le fichier « /usr/etc/rc.local » sous Linux 2.6.

Il est alors possible de customiser entièrement le lancement des applications au démarrage au travers de ces 2 fichiers. La règle à respecter pour garder la main sous le shell au démarrage est la suivante : lorsque vous lancer une application qui possède une boucle « while(1) », il est impératif de la lancer en tâche de fond en utilisant un « & » en fin de ligne.

Ainsi, pour lancer un driver avec la commande « insmod », ajouter la ligne « insmod mondriver.ko » (pas de « & » car la commande insmod redonne la main immédiatement), alors que pour lancer votre application, vous devrez faire « monappli & » (l'application ne rend pas la main, il faut donc la lancer en tâche de fond).

Y a-t-il un serveur web pour la carte DEV2410 ?

Le serveur WEB le plus utilisé est le serveur WEB BOA qui permet la prise en compte des scripts CGI. Vous pourrez ainsi facilement afficher des pages HTML et des images sur MS InternetExplorer ou Mozilla par exemple. Mais bien souvent, le but d'un tel serveur WEB est de pouvoir afficher des informations liées à l'équipement (fichier de log, température, ...) et aussi de piloter à distance la carte. Il vous faut donc inter-agir avec du code en C qui vous permettra d'accéder à n'importe quelle fonctionnalité de votre carte. C'est à cela que les scripts CGI sont destinés. Les pages HTML sont ainsi créées dynamiquement par des fichiers C.

Un exemple de serveur WEB est désormais disponible dans la dernière version de notre distribution Linux 2.6 (fichier userland.tar.gz).

Est-il possible de faire fonctionner les CPU S3C2410 à plus de 200MHz comme c'est le cas sur vos cartes ?

Non, pas dans la version commerciale des cartes SODIMM. Nous proposons une version industrielle sur demande uniquement. Celle-ci permet de faire fonctionner la carte dans une gamme de température étendue, et aussi d'augmenter la fréquence CPU du processeur, et ceci jusqu'à 266MHz (augmentation de 30%).

Nous sommes aussi en train de prototyper une version 100% compatible permettant d'atteindre une fréquence CPU de 400MHz.

Peut-on diminuer la taille des mémoires de la carte SODIMM et espérer ainsi économiser d'autant sur le PU HT du produit ?

Afin d'effectuer des économies substantielles sur la carte SODIMM, il convient non pas d'utiliser des mémoires les plus petites possibles, mais les plus disponibles, les plus utilisées et donc au final les moins chères. Typiquement, nous utilisons 2 mémoires SDRAM de 32Mo chacune, il est possible d'effectuer une modification du bus d'adresse de sorte à ne monter qu'une seule des 2 puces mémoires. Enfin, il est possible de remplacer la mémoire flash NAND par une mémoire de moins grande capacité, par exemple 32Mo sans pour autant devoir utiliser des mémoires peut usitées et donc plus chères.

J'aimerais utiliser une carte 2410SODIMM. Est-il possible de la customiser ? Peut-on changer la taille mémoire ?

Nous sommes en effet en mesure de modifier le design de la carte, soit pour en modifier la sérigraphie Pragmatec et la remplacer par votre propre sérigraphie, soit pour changer les composants utilisés, en enlever certains pour en réduire le coût ou bien en ajouter des spécifiques à vos besoins.

Concernant la mémoire, il est possible de ne mettre que 32 Mo de SDRAM et de remplacer la mémoire flash par une de plus grande capacité.



4 Divers

La carte SODIMM 2410 est annoncée pour une température commerciale. Est-il possible de la faire fonctionner à des température négative ?

La carte 2410SODIMM est proposé de base dans sa version commerciale, c'est-à-dire pour des températures de 0° à +70°C. Il est toutefois possible de la faire fonctionner dans une version industrielle, de -40° à +85°C. Pour cela il est impératif de devoir changer les puces qui composent la carte.

Pragmatec n'assure pas de stock France sur la version industrielle, mais uniquement la version commerciale. Nous ne fournissons la version industrielle que sur commande, pour un minimum de commande de 50 pièces. Les délais de production peuvent s'avérer importants, jusqu'à 12 semaines, au lieu de 6 semaines pour une version commerciale.

La carte 2410SODIMM a-t-elle passé des tests CEM ? Quant est-il pour les vibrations et les chocks ?

La carte 2410SODIMM n'est qu'un module, et lors de tests CEM il convient de faire passer les tests sur le produit dans son intégralité, non pas sur les modules qui le composent. Des tests CEM ont été réalisés sur des modules SODIMM dans le cadre de projets spécifiques de certains de nos clients. Des tests ont même été pratiqués pour répondre aux spécifications en vigueur dans les domaines militaires et aéronautiques sur des cartes 2410SODIMM tropicalisées. Les résultats ont été très concluants car l'ensemble des tests a été passé avec succès. Vous pouvez demander à obtenir auprès de Pragmatec un tableau récapitulatif des performances obtenues lors de ces tests.

Quelle est la consommation moyenne d'une carte SODIMM ?

La carte SODIMM entre dans la gamme des systèmes 1W, c'est-à-dire qu'elle consomme typiquement 200mA sous 5V. Cette consommation correspond à un mode de fonctionnement non optimal, avec tous les drivers actifs et un CPU à 200MHz. La consommation de la carte est presque linéaire en fonction de la fréquence CPU. Ainsi, lorsque l'on baisse la fréquence du CPU à 90MHz, la consommation tombe à 0.5W.

Afin de réduire la consommation, il est donc nécessaire de réduire la fréquence CPU et de diminuer le nombre de périphériques en fonctionnement. La limite atteignable est de 20mA sous 5V, soit 0,1W. Dans de nombreux projets, cette limite n'est pas satisfaisante, notamment pour les systèmes sur batterie. Il est donc nécessaire de couper l'alimentation de la carte SODIMM et de la redémarrer périodiquement. Pour cela, on utilise généralement un μC externe, qui lui sera capable de se mettre en veille avec une consommation de quelques μA sous 5V.

Est-ce que vous assurez la formation et le support pour les non linuxiens qui débloquent sur ces CPU ARM et du linux embarqué ?

L'assistance de Pragmatec qui peut être proposée dans le cadre des projets clients s'articule autour de 3 axes :

- une assistance gratuite basée sur un réseau d'entraide maintenu par les utilisateurs de ces cartes (forum des développeurs sur www.pragmatux.net)
- une assistance payante de 300 euros HT par mois, permettant d'assister le client un nombre de fois illimité dans le mois, que ce soit par mail ou par téléphone.
- une formation orientée « coaching », c'est-à-dire réalisée uniquement chez le client et dans le cadre d'un projet précis. La formation n'est pas généraliste, elle est ciblée sur les aspects techniques du projet du client. Elle dure de 1 à 2 jours et l'objectif est de laisser le client afin qu'il soit autonome sur le projet.

Le plus souvent, les clients novices sur ces technologies choisissent la formation « coaching » suivie par une assistance de 1 mois.

Les composants que vous utilisez ne se trouvent pas chez Radiospares et Farnell. Ou peut-on les trouver ?

Hélas, il est assez difficile d'obtenir des composants fortement répandus dans la conception de micro-PC en France. En Asie, au contraire, ces composants n'obtiennent facilement de par la grande proximité des fabricants de ces composants, qui sont presque tous présents en Asie. Nous sommes en mesure de nous faire livrer ces composants et nous disposons d'un petit stock en France, destiné à dépanner nos clients lors de la phase de définition du produit, ou bien pour la réparation de carte ou encore l'élaboration de prototypes.

Ma carte consomme trop. Comment réduire la consommation ? Peut-on éteindre des périphériques ?

Afin de réduire la consommation, il est possible de ne dévalider certains périphériques, mais la consommation d'une carte vient en très grande partie de la fréquence de fonctionnement du CPU. Sur une carte ARM7 avec uClinux 2.4 il n'est pas possible de modifier à la volée la fréquence du CPU. Ceci est possible avec la version Linux 2.6 disponible sur nos cartes ARM9.