

DÉBUGGEUR, TRACE TEMPS-RÉEL, ANALYSEUR LOGIQUE



energy profiling



android debugging



trace-based debugging



multicore debugging



serial trace



long-time trace

Toujours en avance sur son temps

Nous produisons sous cette devise depuis plus de 30 ans des outils de développement pour l'industrie embarquée. Lauterbach est souvent en avance sur son temps et intègre aussitôt les nouvelles architectures et tendances technologiques.

Ainsi nous avons obtenu la reconnaissance de tous les grands fabricants de semi-conducteurs dans le monde. Une coopération étroite avec Lauterbach dans le développement et la mise en œuvre de nouvelles techniques est pour cela depuis plusieurs années recherchée et appréciée. Par le biais de cette collaboration, des idées innovatrices ont pu être reconverties en produits et fonctionnalités aujourd'hui avancés.

En outre, Lauterbach s'appuie sur une orientation client sans compromis. Les désirs et les suggestions des utilisateurs de TRACE32 représentent une contribution précieuse au développement de nos produits. En pratique, des suggestions client ont été directement mises en œuvre et sont désormais disponibles dans TRACE32.

Quelles sont les tendances envisagées par Lauterbach actuellement? Quelles nouvelles technologies sont en cours d'établissement sur le marché?

Le débuge d'Android

Aujourd'hui le débuge du système d'exploitation Android est l'un des sujets prioritaires, particulièrement dans le

domaine de la téléphonie mobile, où les applications sont de plus en plus développées pour être indépendantes de l'architecture grâce à l'utilisation de machines virtuelles (VM). Android, de Google et sa VM appelée Dalvik sont aujourd'hui très utilisés. L'un des défis du débugeur est de détecter les erreurs complexes générées seulement à travers l'interaction des éléments suivants: les applications, la machine virtuelle, le système d'exploitation ainsi que la couche matérielle. Pour atteindre cet objectif, il est nécessaire d'avoir une transparence parfaite à travers ces différentes couches.

Sur la demande de certains fabricants de téléphones mobiles, depuis mi-2010, Lauterbach a initié le développement d'une interface de programmation (API) permettant le débuge de machines virtuelles. La plate-forme Android »

CONTENU

Nouvelles architectures supportées	4
Trace des plates-formes virtuelles	5
API débuge de machines virtuelles	6
Extensions et nouvelles versions RTOS	8
Port de trace série	9
Débit Augmenté pour le <i>Real-Time Streaming</i>	10
Analyse de performance d'énergie	11
Le forum des experts	12

est utilisée dans ce contexte comme référence. L'objectif de ce développement est d'avoir une interface ouverte qui permet aux fournisseurs de machines virtuelles Open et Closed-Source d'adapter leurs produits pour un débogue avec TRACE32. Pour plus d'informations sur ce sujet, n'hésitez pas à consulter l'article "API avec intégration des machines virtuelles".

Analyse de performance d'énergie

A l'heure des changements climatiques mondiaux, la mesure de l'énergie consommée par un système embarqué est devenue un point crucial dans l'émergence de l'informatique "verte". Aujourd'hui dans la plupart des revues spécialisées, on peut observer une multitude d'articles vantant tous dispositifs sous batterie et micro-contrôleurs basse consommation. Les prix d'innovation sont de nos jours attribués en majorité sur des technologies développées dans ce domaine.

Quoi qu'il en soit, la téléphonie mobile a toujours porté une attention particulière dans l'analyse des temps d'exécution et de mise en veille de leurs applications. Depuis plusieurs années déjà, une politique de réduction des consommations en énergie est mise en place. Néanmoins, elle reste efficace uniquement lorsque le logiciel prend en compte toutes les possibilités d'économie d'énergie fournies par l'architecture processeur.

Depuis 2006 déjà, les outils Lauterbach intègrent cette fonctionnalité de mesure. Cela permet de comparer et d'analyser d'une façon très simple la relation entre le code exécuté et le courant consommé pour en déduire d'un simple clic la puissance absorbée par votre application. Depuis mi-2010, la sonde *CombiProbe* a été mise à jour pour supporter ce type de mesure. Pour plus d'informations, consulter en page 11 l'article "Analyse de performance d'énergie avec la *CombiProbe*".

Débogue Multi-Cœurs

Les systèmes Multi-Cœurs sont utilisés depuis plus de dix ans déjà sur le marché de l'embarquée. En effet Lauterbach offrait déjà en 2001 des outils pour ce type de systèmes. Ce sont aujourd'hui des sujets, plus que jamais d'actualité. La recherche d'une visibilité accrue sur les processus internes exige l'intégration de nouvelles cellules de trace à l'intérieur même des systèmes de débogue de ces composants.

Alors qu'initialement les informations de trace étaient uniquement générées par le processeur, il existe aujourd'hui une multitude d'autres sources de trace.

a) Génération d'une trace permettant la visualisation des transferts sur les bus internes du CPU :

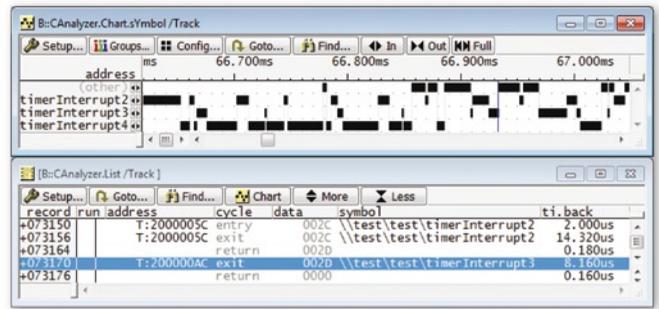


Fig. 1: Trace des interruptions pour le Cortex-M3.

- ARM CoreSight avec AMBA/AHB Trace Macrocell (HTM)
- MCDS avec le "System Peripheral Bus" (SPB) et le "Local Memory Bus" (LMB) pour le TriCore d'Infineon
- RAM Trace Port pour les processeurs de TI
- Trace DMA et FlexRay pour NEXUS Power

b) Génération d'une Trace pour les IP (Intellectual Property) internes au processeur comme la trace des interruptions (figure 1).

c) Génération d'une Trace permettant d'afficher des informations produites par le logiciel lui même entre autres *Instrumentation Trace Macrocell* (ITM) et *System Trace Macrocell* (STM) sur ARM CoreSight.

Le développement poussé des concepts du débogueur TRACE32, assure une intégration parfaite de toutes ces nouvelles sources de trace. Cela permet une configuration facilitée et une analyse simplifiée des informations de trace reçues.

Les ports de trace Série

Ces nouvelles informations de trace requièrent une augmentation de bande passante et de vitesse de transfert sur ces ports de trace.

Une des importantes innovations dans l'industrie embarquée de ces dernières années a été le développement d'interfaces de trace série haute vitesse. Cette technologie, issue des transferts de données série "haute vitesse", est utilisée par les fabricants de disques durs depuis des années. Celle-ci a été appliquée pour la première fois en 2008 pour véhiculer la trace d'exécution code à travers la standardisation d'un protocole *High Speed Serial Trace Port* (HSSTP) par ARM. En parallèle Lauterbach développait déjà une nouvelle sonde de trace intégrant cette technologie.

Désormais, il existe d'autres familles de processeurs utilisant cette technologie. Les développements actuels dans ce domaine sont disponibles en page 9. »

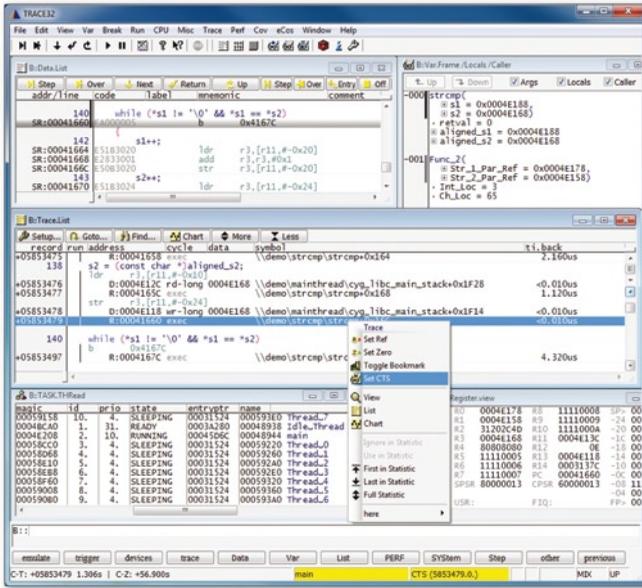


Fig. 2: Débugge basé sur la trace.

L'augmentation de la mémoire de trace

Une interface de trace très rapide intégrant un débit plus important exige obligatoirement une taille de mémoire de trace plus grande. Sinon il serait impossible de capturer une portion de code exécuté suffisamment large, permettant la recherche d'erreur d'un système embarqué.

Néanmoins, fournir une mémoire de trace plus importante n'a de sens que s'il existe une infrastructure capable de gérer et d'analyser rapidement ces données. Cela s'applique en particulier à l'analyse d'une trace intensive de données et de fonctions comme le débuge basé sur la trace (figure 2). L'augmentation de la taille mémoire des SDRAM, les architectures PC de plus en plus puissantes et l'interface Ethernet Gigabit ont permis à Lauterbach de lancer en 2007 l'outil de trace *PowerTrace II* équipé de 4GB de mémoire.

Mi-2008, Lauterbach développa une nouvelle méthodologie d'enregistrement et d'analyse de la trace appelée *Real-Time-Streaming*. Cette démarche issue d'une collaboration client, a ainsi permis d'augmenter les temps d'enregistrement de la trace assurant une analyse de couverture de code à long terme. Nous avons aussi pu obtenir une analyse détaillée des temps d'exécution du système ainsi qu'une étude facilitée des erreurs sporadiques.

La vraie nouveauté du *Real-Time Streaming* est de permettre la transmission "à la volée" des données de trace vers le PC pendant l'exécution CPU. Ces données sont directement analysées à la réception et peuvent optionnellement être enregistrées sur le disque dur à ce moment là.

Le *Real-Time Streaming* ne peut fonctionner que si toutes les étapes de traitement de données sont optimisées.

Débuge basé sur la trace

Le débuge basé sur la trace, aussi connu sous le nom de CTS (Context Tracking System), permet à l'utilisateur d'effectuer le débuge pas-à-pas de son application et cela après qu'elle se soit exécutée. TRACE32 rend cela possible car il peut reconstruire l'état de la cible sur chaque enregistrement. Cette reconstruction inclut entre autres, le contenu des registres, de la mémoire, l'état des variables, l'état de la pile, l'état des tâches et bien plus encore.

Après avoir choisi son point de départ dans la trace, toutes les commandes de débuge peuvent être utilisées. Ces commandes sont alors traitées en se basant sur la reconstruction de la trace. De nombreux utilisateurs de ce mode de débuge apprécient le fait de pouvoir faire du pas-à-pas arrière et donc de pouvoir revenir sur l'appel d'une fonction.

Le débuge basé sur la trace offre également d'autres fonctions très utiles :

- L'affichage de la trace en langage de haut niveau avec les variables locales
- L'analyse des temps d'exécution et l'affichage de l'arbre d'appel des fonctions
- Comblent les lacunes de la trace qui se produisent quand la quantité des données générées dépasse la capacité du port de trace

www.lauterbach.com/cts.html

Ceci inclut la transmission et l'analyse, ainsi que la recherche ciblée d'informations de trace dans un fichier enregistré sur le disque dur.

Ces nouvelles techniques de trace sont réutilisées. Il est prévu, par exemple, d'appliquer la compression des données mise au point pour le *Real-Time Streaming* lorsque les données sont transférées sur le système hôte à la fin de l'enregistrement. Plus de détails sur ce sujet sont disponibles en page 10.

Perspectives

En plus des tendances actuelles, une multitude de développements et d'améliorations sont en cours. En feuilletant notre lettre d'information 2011, vous serez probablement intéressés par certaines nouveautés qui pourraient concerner votre projet. Nous présenterons plusieurs d'entre elles les 29 et 31 Mars 2011 au Salon RTS Embedded Systems.



Venez-nous rendre visite :
Hall 8, Stand D14.

Nouvelles Architectures supportées

Nouveaux dérivés	
Actel	LA-7844 (Cortex-M) • A2F060, A2F200, A2F500
AppliedMicro	LA-7723 (PPC400) • APM80186, APM821x1 • APM86290 LA-7752 (PPC44x) • PPC460SX
ARM	LA-7843 (Cortex-A/R) • Cortex-A15 • Cortex-A15 MPCore LA-7844 (Cortex-M) • Cortex-M4 • SC000, SC300
Atmel	LA-7844 (Cortex-M) • AT91SAM3S, AT91SAM3N LA-3779 (AVR32) • AT32UC3A/B/C/D/L
Broadcom	LA-7760 (MIPS32) • BCM3549/35230/4748 • BCM5354/5358/5331X • BCM6816/6328/6369 • BCM7407/7413/7420
Cavium	LA-7761 (MIPS64) • CN63XX
Ceva	LA-3711 (CEVA-X) • CEVA-X1643, CEVA-XC
Cortus	LA-3778 (APS) • APS3/B/BS/S
Cypress	LA-7844 (Cortex-M) • PSoC5
Faraday	LA-7742 (ARM9) • FA726TE
Freescale	LA-7736 (MCS12X) • MCS9S12GC/GN/Q LA-7732 (ColdFire) • MCF5301x, MCF5441x LA-7845 (StarCore) • MSC8156 LA-7742 (ARM9) • i.MX28 LA-7843 (Cortex-A/R) • i.MX53 LA-7844 (Cortex-M) • Kinetis
Freescale (Cont.)	LA-7753 (MPC55xx/56xx) • MPC5602D/P • MPC564XA/B/C/S • MPC567XF/R LA-7729 (PowerQUICC II) • MPC830X LA-7764 (PowerQUICC III) • P10xx, P20xx, P40xx • P3041 (2H/2011) • P5010, P5020 (2H/2011)
Fujitsu	LA-7844 (Cortex-M) • FM3
Infineon	LA-7756 (TriCore) • TC1182, TC1184 • TC1782, TC1782ED • TC1784, TC1784ED • TC1791, TC1791ED • TC1793, TC1793ED • TC1798, TC1798ED LA-7759 (XC2000/C166S V2) • XC22xxH/I/L/U • XC23xxC/D/E/S • XC27x2/x3/x7/x8 • XE16xFH/FU/FL
Intel®	LA-3776 (Atom™/x86) • E6xx, Z6xx, N470 • Core i3/i5/i7, Core2 Duo
Lantiq	LA-7760 (MIPS32) • XWAY xRX200
LSI	LA-7765 (ARM11) • StarPro2612, StarPro2716 LA-7845 (StarCore) • StarPro2612, StarPro2716
Marvell	LA-7742 (ARM9) • 88F6282, 88F6283, 88F6321 • 88F6322, 88F6323 LA-7765 (ARM11) • 88AP510-V6 LA-7843 (Cortex-A/R) • 88AP510-V7
MIPS	LA-7760 (MIPS32) • MIPS M14K, MIPS M14KC
Netlogic	LA-7761 (MIPS64) • XLR, XLS
NXP	LA-7844 (Cortex-M) • LPC11xx • EM773

Nouveaux dérivés

Ralink	LA-7760 (MIPS32) • RT3052, RT3662
Renesas	LA-3777 (78K0R/RL78) • 78K0R/Hx3/Lx3/Ix3 • 78F804x, 78F805x • RL78/G12, RL78/G13 LA-3786 (RX) • RX610/6108/621/62N/630
STMicro-electronics	LA-7753 (MPC55xx/56xx) • SPC560D/P, SPC56APxx • SPC564Axx, SPC56ELxx LA-7844 (Cortex-M) • STM32F100, STM32L15x
ST-Ericsson	LA-7843 (Cortex-A/R) • DB5500, DB8500
Tensilica	LA-3760 (Xtensa) • LX3

Texas Instruments	LA-3713 (MSP430) • MSP430xG461x • MSP430x20x1/x2/x3 LA-7742 (ARM9) • AM1707/1808/1810 LA-7843 (Cortex-A/R) • OMAP36xx LA-7838 (TMS320C6x00) • OMAP36xx
Toshiba	LA-7742 (ARM9) • TMPA900, TMPA910 LA-7844 (Cortex-M) • TMPM330, TMPM370
Trident	LA-7760 (MIPS32) • HiDTV PRO-QX
Wintegra	LA-7760 (MIPS32) • WinPath3, WinPath3-SL
Zoran	LA-7760 (MIPS32) • COACH 12

Trace des plates-formes virtuelles

Depuis Novembre 2010 Lauterbach supporte la trace des modèles rapides ARM.

Dans le but de développer le logiciel avant même la disponibilité des premiers prototypes matériels, la modélisation logicielle du matériel est souvent utilisée. Avec les

modèles rapides, ARM offre à ces clients un kit logiciel de programmation pour les nouveaux designs ARM.

Lauterbach supporte depuis 2008 déjà le débuge des modèles rapides à travers l'interface CADI. TRACE32 inclut désormais aussi le modèle d'une interface de trace qui a été introduite dans la version 5.1 des modèles rapides. Pour pouvoir préparer les informations de la trace et les enregistrer dans la plate-forme virtuelle, les fabricants de débugeurs ont la possibilité de charger leur propre plug-in de trace. La figure 3 montre une vue d'ensemble de l'interaction de TRACE32 avec les modèles rapides.

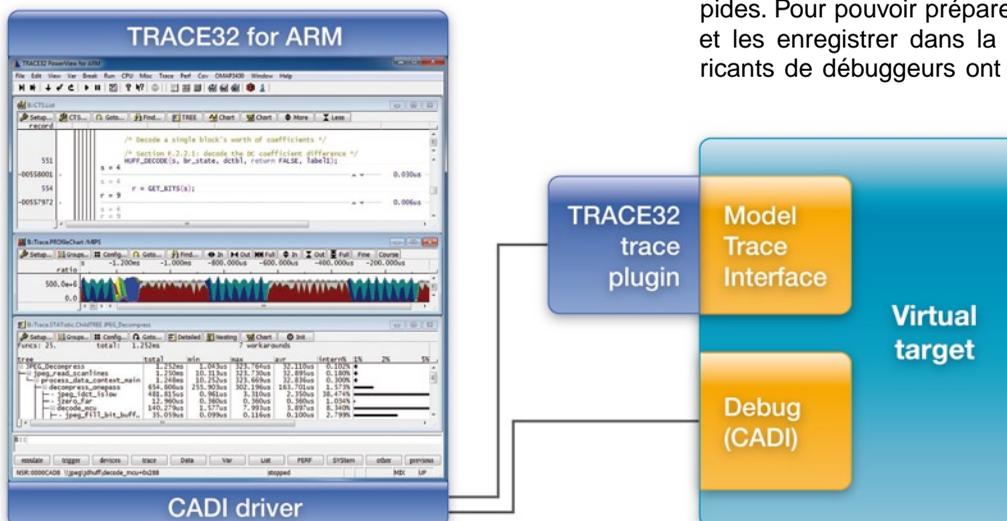


Fig. 3: TRACE32 inclut le débuge et la trace des plates-formes virtuelles.

Plus d'informations sur le débuge des plates-formes virtuelles sont disponibles à l'adresse suivante :

www.lauterbach.com/frontend.html

API débogue de machines virtuelles

TRACE32 inclut depuis mi-2006 le débogue d'applications Java pour les machines virtuelles J2ME CLDC, J2ME CDC et Kaffe. Étant donné que les machines virtuelles ont une popularité croissante, le nombre de fournisseurs de ces machines ne cesse d'augmenter. Un bon nombre de ces VMs ne sont pas open source. Pour permettre aux fournisseurs de VMs et à leurs clients d'adapter le débogue de leurs VMs de manière rapide et flexible, Lauterbach développe depuis mi-2010 une solution Open-Source.

La machine virtuelle Dalvik d'Android pour les processeurs ARM est utilisée comme référence pour le développement d'une interface de programmation pour le débogue en mode stop.

Deux mondes de débogue

Pour ceux qui ne sont pas familiers avec Android, voici donc une courte introduction. Android est, du point de vue du développeur, un système opérationnel open-source constitué par les différentes couches logicielles suivantes :

- Un noyau linux avec ces pilotes hardware.
- Le moteur d'exécution Android avec la machine virtuelle Dalvik et une série de bibliothèques : les bibliothèques classiques du cœur Java, les bibliothèques spécifiques à Android, les bibliothèques créées en C/C++.
- Les applications Java et le framework les supportant.

Les logiciels du système Android sont écrits en différents langages de programmation :

- Le noyau Linux et quelques bibliothèques ainsi que la machine virtuelle Dalvik sont codés en C/C++ ou assembleur.

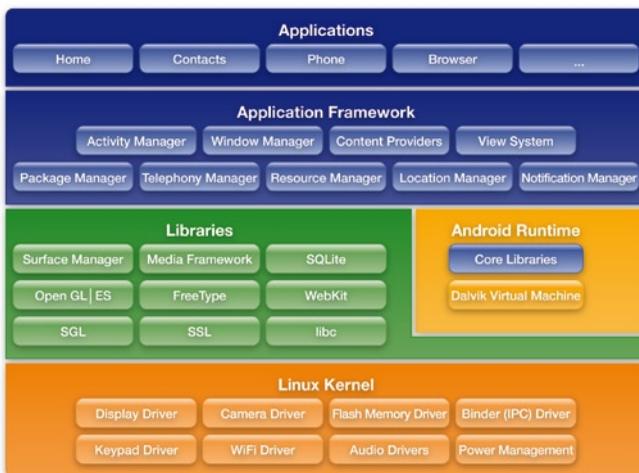


Fig. 4: Le Software-Stack open source d'Android.

- Les applications de la VM et le framework les supportant sont programmés en Java.

Le code respectif est testé dans son propre monde de débogue.

Débogue du code C/C++ et assembleur

La partie d'Android codée en C/C++ et assembleur peut être déboguée grâce à l'interface JTAG en mode stop. Pour cela, TRACE32 communique directement avec le processeur de la plate-forme matérielle (figure 5).



Fig. 5: En mode stop, le débogueur communique directement avec le processeur de la plate-forme matérielle du système Android.

Une caractéristique du mode de débogue stop est que tout le système Android est arrêté lorsque le processeur est stoppé pour effectuer le débogue.

Les avantages importants du débogue en mode stop :

- Il suffit d'avoir une communication JTAG entre le débogueur et le processeur.
- Aucun serveur de débogue sur la carte cible n'est nécessaire. Le mode stop est par la suite approprié pour tester la version release du logiciel.
- Il est possible de tester le code sous des conditions temporelles réelles et par la suite d'avoir un débogue efficace des problèmes qui surviennent seulement en temps réel.

Aujourd'hui, le débogue en mode stop n'inclut pas le débogue des applications de la VM Dalvik. »

Débugge du code Java

Le code Java d'Android est habituellement testé en utilisant l'outil de développement d'Android ADT fourni par Eclipse. Le serveur adb (*Android Debug Bridge*) sur le système hôte communique par USB ou Ethernet avec le daemon adb sur le système cible (figure 6).

Pour pouvoir tester son logiciel avec ADT, il est indispensable de compiler les applications de la VM en version de débuge et de les déclarer dans l'*Android Manifest* comme ouverte au débuge. En outre, le débuge de la plate-forme matérielle doit être activé.

Le débuge du code Java avec ADT est donc possible. ADT a toutefois dans les cas suivants ses limites :

- Si les erreurs se produisent seulement dans la version release du code.

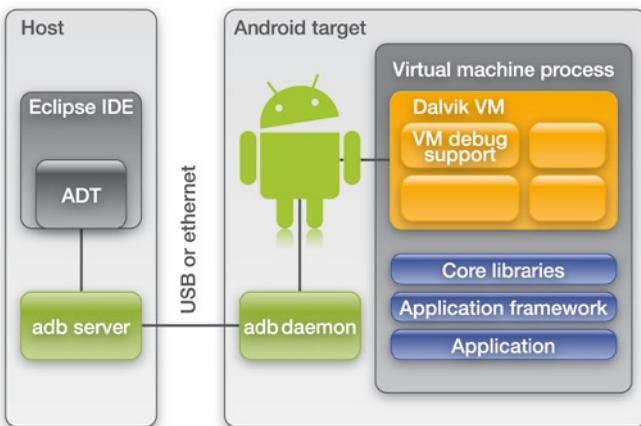


Fig. 6: L'outil de développement ADT pour le débuge du code Java.

- Si les erreurs surviennent uniquement lorsque l'application Java interagit avec un service C/C++ ou bien un pilote matériel du noyau Linux.
- Le débuge n'est plus possible si la communication entre le serveur et le daemon adb est rompue.

Débugge en mode stop

Afin de tester tout le système Android, de l'application Java jusqu'aux pilotes matériels du noyau Linux, dans des conditions temps réel, Lauterbach a enrichi le débuge en mode stop par une connexion de débuge sur les machines virtuelles.

Le débugeur JTAG communique directement avec le processeur sur la plate-forme matérielle d'Android. Une fois le processeur arrêté, il est capable d'accéder à toutes les informations du système. Le défi pour le débugeur sera de rechercher les informations importantes en mémoire et de les afficher d'une façon simple et compréhensible pour les utilisateurs.

Un premier niveau d'abstraction fourni par TRACE32 avait permis aux utilisateurs de débuge les systèmes d'exploitation à travers plusieurs espaces d'adressages virtuels. Puis un autre niveau d'abstraction avait permis de débuge du code Java et cela indépendamment du système d'exploitation utilisé.

Pour pouvoir débuge des applications s'exécutant sur une VM pour un système comme Android, où les machines virtuelles sont instanciées dans les processus du système, il est nécessaire de combiner le débuge du système d'exploitation et le débuge Java.

Pour la mise en œuvre de cette nouvelle complexité, Lauterbach développe une solution inédite, ouverte et facile à déployer.

La solution ouverte

Le débuge en mode stop de Lauterbach va inclure, dans un futur proche, les niveaux d'abstractions suivants :

- Débuge en langages de haut niveau
- Intégration du débuge des systèmes d'exploitation
- Intégration du débuge des machines virtuelles

Le débuge en langages de haut niveau est un élément essentiel de TRACE32, possible pour une application pourvue de symboles et d'informations de débuge.

L'intégration du débuge des systèmes d'exploitation sur la cible doit toujours être configurée par l'utilisateur de TRACE32. Lauterbach fournit des exemples de configuration pour les principaux systèmes d'exploitation. Néanmoins, en utilisant l'API RTOS incluse dans TRACE32, il est possible de l'adapter aux systèmes d'exploitation propriétaires.

L'intégration du débuge des machines virtuelles est déjà disponible pour les plateformes J2ME CLDC, J2ME CDC et Kaffe. Lauterbach fournit une configuration pour la machine virtuelle Dalvik qui gère la plate-forme Java sur les systèmes Android. Une machine virtuelle autre pourra aussi être adaptée grâce à la VM API. »

La machine virtuelle Dalvik

Dalvik est le nom de la machine virtuelle utilisée par Android. La machine virtuelle Dalvik est le modèle logiciel d'un processeur qui exécute le code machine Java. Les machines virtuelles permettent de développer des applications indépendamment de l'architecture du processeur. Dans le cas d'une transition vers une nouvelle plate-forme matérielle, il suffit de faire uniquement le portage de la VM.

Il est donc possible pour les fournisseurs de VMs closed-source d'intégrer le débogue de leurs VMs dans TRACE32 et de le mettre à disposition de leurs clients.

Le modèle d'implémentation

Afin de pouvoir déboguer un système Android sur plate-forme ARM, des applications Java jusqu'aux pilotes matériels du noyau Linux, les extensions suivantes seront nécessaires (figure 7).

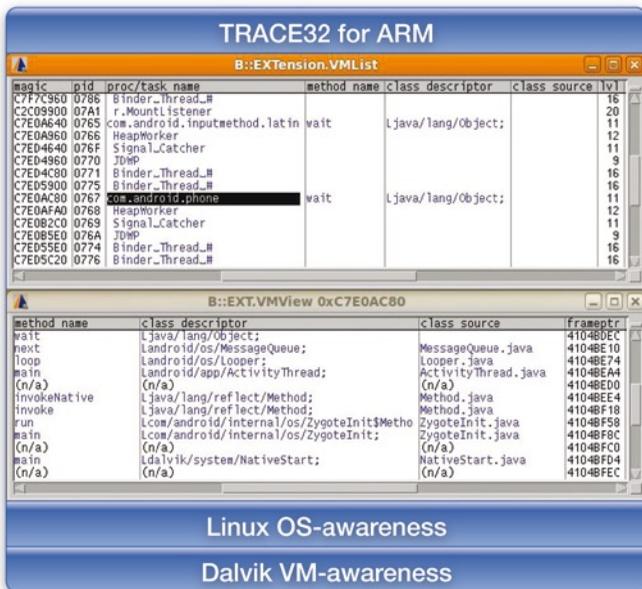


Fig. 7: Dans le modèle d'implémentation, il est nécessaire de charger l'OS-Awareness et la VM-Awareness de Dalvik dans TRACE32.

Extensions et nouvelles versions RTOS

- Les scripts de TRACE32 ont été adaptés au Timesys embedded Linux.
- OSEK ORTI assure maintenant la génération de messages de trace (NEXUS Ownership Trace Messages) dans le cas de basculement des tâches. De cette façon, il est possible d'effectuer dans TRACE32 des mesures de temps de basculement des tâches pour les processeurs MPC55xx/MPC56xx même si la trace NEXUS ne produit pas l'information sur les données.

Les adaptations pour les versions suivantes ont été déjà appliquées ou bien sont envisagées :

- OSEck 4.0
- QNX 6.5.0
- Symbian^3 sur ARM,
- Symbian^4 prévu pour Q1/2011
- Windows CE6 sur Atom™

- Le débogue Linux Lauterbach, disponible sur TRACE32 depuis 1998
- Le débogue VM pour Dalvik, dont une pre-release peut être téléchargée sur le site web Lauterbach. Il devra être configuré pour la plate-forme utilisée.

www.lauterbach.com/vmandroid.html

Il est actuellement possible de déterminer et de lister toutes les applications Java en cours d'exécution (EXTension.VMList sur la figure 7). De même qu'il est possible d'afficher la pile de la VM pour une application Java déterminée (EXTension.VMView sur la figure 7). La prochaine étape sera l'affichage du code source qui est en cours d'exécution par la VM. Le but du développement est bien sûr la mise au point d'un mode stop pour les applications de la VM avec toutes les fonctionnalités d'un débogueur moderne.

Nouveaux RTOS supportés

DSP/BIOS sur ARM	Q2/2011
OSEK/ORTI SMP	Q2/2011
PikeOS V3.1 de SYSGO sur PowerPC	disponible
T-Kernel sur ARM	disponible
Windows Embedded Compact 7 sur ARM	disponible
µC/OS-III sur ARM	disponible

PikeOS de SYSGO

PikeOS est un produit innovant de SYSGO fournissant une plateforme logicielle pour systèmes embarqués sur laquelle plusieurs VMs peuvent fonctionner simultanément. La technologie de virtualisation sûre et sécurisée permet à de nombreuses APIs de système d'exploitation, appelées "personnalités", de fonctionner en même temps sur une seule machine. Le micronoyau de PikeOS dispose d'une architecture qui lui permet d'être utilisé dans des appareils limités en ressources comme dans des équipements complexes. PikeOS supporte de multiples architectures de processeurs mono- et multi-cœurs et dispose d'une grande gamme de personnalités.



Port de trace série, utilisation accrue

Plus rapide, plus haut, plus fort ! Ce n'est pas seulement une devise sportive. Ce slogan semble également devenir un principe dans le domaine de la microélectronique. Des fréquences d'horloge toujours plus importantes et des étapes de traitement en parallèle amélioré, traduisent depuis plusieurs années un accroissement incroyable des vitesses de traitement. Ce n'est donc pas surprenant, si aujourd'hui ces technologies sont utilisées pour la transmission des informations de trace.

L'interface de trace, à travers laquelle les processeurs révèlent des informations détaillées sur leur fonctionnement interne, doit particulièrement suivre le rythme du flux d'informations croissant. En effet, de nombreux développeurs de systèmes embarqués ne souhaitent plus se passer de ce type d'informations sur le fonctionnement interne du processeur. Par conséquent, on ne compte plus les efforts entrepris pour augmenter le débit de l'interface de trace. Depuis des années, la meilleure solution pour augmenter les volumes de données transmises par la trace était l'augmentation de la fréquence d'horloge et de la taille du bus de trace.

Cette solution a toutefois un coût non négligeable. D'une part, l'augmentation de la taille du bus de trace utilise, des broches précieuses du processeur. D'autre part, l'augmentation des fréquences d'horloge engendre souvent une dégradation de la qualité des signaux ce qui oblige à prévoir une compensation sur tous les signaux du bus de trace. Ce n'est que par l'optimisation des algorithmes de sa technologie *AutoFocus* que Lauterbach a pu garantir un enregistrement sans erreur des signaux de trace à haute fréquence.

Les architectures processeurs étant toujours plus rapides et complexes, les interfaces de trace appliquent désormais une technique utilisée depuis longtemps dans d'autres domaines d'application. Par exemple, Les technologies *SATA*, *Fibre Channel*, *PCI Express* et *USB 3.0* utilisent une transmission série haute vitesse. Le nombre très réduit des lignes différentielles de données utilisées est largement récompensé par des taux de transmissions extrêmement élevés.

L'intégration d'interfaces série très haute vitesse sur le composant peut se révéler coûteux et générer certains problèmes. Par exemple, les entrées/sorties doivent absolument fonctionner à des vitesses beaucoup plus élevées. Néanmoins, l'expérience croissante dans la mise en œuvre de ce type d'interfaces série, dans la bande du gigahertz, agrandissent le spectre des connaissances permettant aujourd'hui de résoudre les différents problèmes rencontrés avec les interfaces série de la trace.

ARM a introduit pour la première fois cette technologie en 2008 via l'interface *High Speed Serial Trace Port*



Fig. 8: Un hardware universel supporte les différents protocoles de trace après une adaptation du firmware et de TRACE32.

(HSSTP). Suivi très rapidement par AMCC avec la technologie Titan, ainsi que Freescale avec les processeurs QorIQ (P4040 et P4080) et Marvell avec SETM3.

Lauterbach a accompagné dès le début cette technologie avec un outil pour la trace série. Un préprocesseur universel a été développé en se basant sur le protocole Aurora. Pour pouvoir s'adapter aux différents protocoles, il nous suffit d'adapter le firmware et TRACE32. Ce qui veut dire que votre système est déjà prêt pour les différentes variantes des protocoles de trace série.

L'outil inclut une taille de trace maximum de 4 Go, permettant de supporter le taux de transfert élevé des interfaces de trace série.

Les ports de trace série supportés

AMCC	APM83290 Trace programme	2009
ARM-HSSTP	ETMv3, PTM, CoreSight ETMv3, CoreSight PTM Trace programme, donnée et Context-ID	2008
Freescale	NEXUS QorIQ P4040 et P4080 Trace Branch et Ownership, Data Write Messages	2010
Marvell-SETM3	CoreSight ETMv3 Trace programme données et Context-ID	2009

Débit Augmenté pour le *Real-Time Streaming*

Le *Real-Time Streaming* est une technique qui permet de transmettre les informations de trace vers le système hôte en cours d'exécution et de les analyser à la volée. Cela nécessite de la part de l'outil de trace la transmission d'une quantité importante de données surtout dans le cas de systèmes multi-cœurs. Pour rendre possible ce scénario, les informations de trace sont compressées par l'outil de trace PowerTracell avant d'être transmises vers le système hôte. Cette fonctionnalité existe dans TRACE32 depuis Décembre 2010.

Le *Real-Time Streaming* inclut actuellement les protocoles ARM ETMv3 et PTM.

La compression au niveau matériel

Le débit de transmission maximum vers le système hôte représente la principale restriction pour le *Real-Time Streaming*. En utilisant une connexion Gb Ethernet point à point, on arrive à atteindre les 500 Mbit/s. Ce débit de transmission maximum devra être suffisant pour transmettre sans aucune perte les données produites du port de trace vers le système hôte.

Pour avoir une estimation réaliste du volume de données à transmettre, il est important de connaître les contraintes du *Real-Time Streaming*.

1. Les principales utilisations du *Real-Time Streaming* sont la couverture de code et le taux de charge CPU. Dans les deux cas, il suffit simplement d'avoir les informations de trace code. Pour obtenir des mesures plus précises sur le taux de charge CPU, il est possible d'activer le mode de trace *Cycle-Accurate*.

2. Afin d'avoir une estimation réaliste du débit de transmission nécessaire, il suffit d'observer la charge moyenne au niveau du port de trace. Les pics de charge sont amortis par le *PowerTrace II* qui agit comme une FIFO, large de 4 Go.

La figure 9 montre un aperçu des charges maximales/moyennes au niveau du port de trace pour les processeurs Cortex. La charge réelle dépend de l'application s'exécutant sur le noyau Cortex.

En implémentant une compression matérielle basée sur un FPGA, il a été possible d'atteindre un débit de transmission maximum vers le système hôte de 3.2 Gbit/s.

La trace à long-terme

On parle de trace à long-terme dans le cas où les données de la trace produites pour le *Real-Time Streaming* ne sont pas seulement analysées sur le système hôte mais aussi enregistrées sur le disque dur.

Afin de pouvoir effectuer une trace à long-terme pour d'autres protocoles comme le NEXUS, Lauterbach offrira en 2011 un streaming des données pures sans pré-analyse. De cette manière, il sera possible de réaliser un enregistrement de trace allant jusqu'à 100 Tera frames sur un système d'exploitation 64 bits.

Pour plus d'informations sur les sujets *Real-Time Streaming* et trace long-terme, connectez-vous à l'adresse suivante :

www.lauterbach.com/tracesinks.html

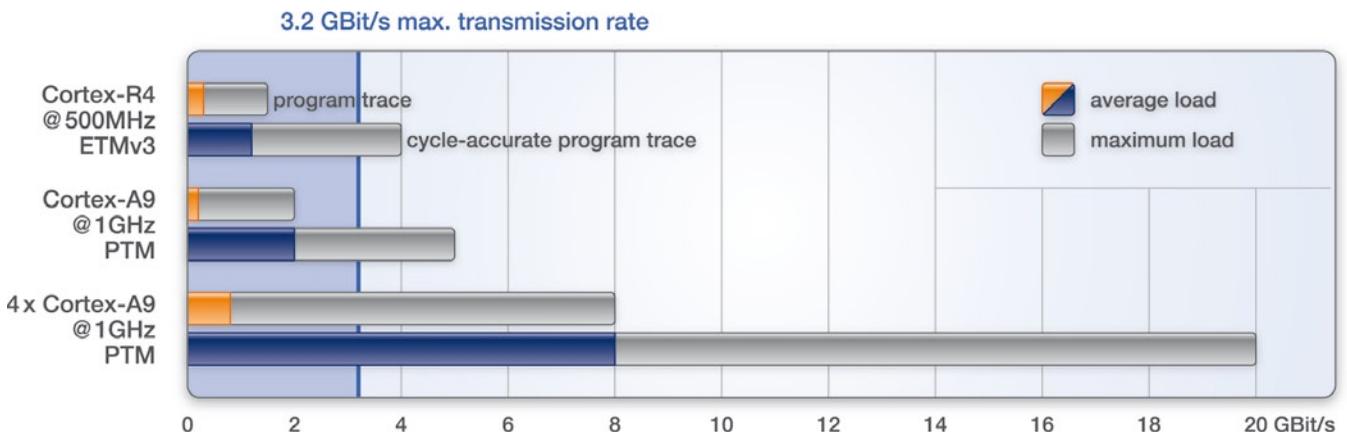


Fig. 9: Un débit de 3,2 Gbit/s est généralement suffisant pour transmettre le flux du programme vers le système hôte au cours de l'enregistrement.

Analyse de performance d'énergie avec le CombiProbe

Si vous disposez de l'outil **CombiProbe** de Lauterbach, vous pouvez dès maintenant l'utiliser pour la mesure d'énergie.

Les analyses suivantes sont désormais possibles :

- Affichage graphique courant/tension sur trois points de mesure différents synchronisés avec la trace code.
- Consommation en énergie de tout le système représenté par fonction.

Quelle partie du code consomme le plus d'énergie ? Quel impact a un changement de code sur les besoins énergétiques de tout le système embarqué ? De telles questions peuvent être analysées et corrigées très rapidement grâce au boîtier **CombiProbe**.

Pour déterminer la consommation en énergie de chaque partie de l'application, les données suivantes doivent être collectées :

- La trace exécution de code de l'application via le port de trace du processeur.
- La mesure d'un courant et d'une tension prélevés sur la carte cible.

Grâce à cela, l'analyse courant / tension est rendue possible sur trois domaines de puissance maximum en connectant la sonde analogique Lauterbach sur un **CombiProbe**.

Tous les échantillons mesurés, obtenant un time-stamp par l'horloge interne du **CombiProbe**, permettent la visualisation d'une façon simple et rapide de la relation tem-

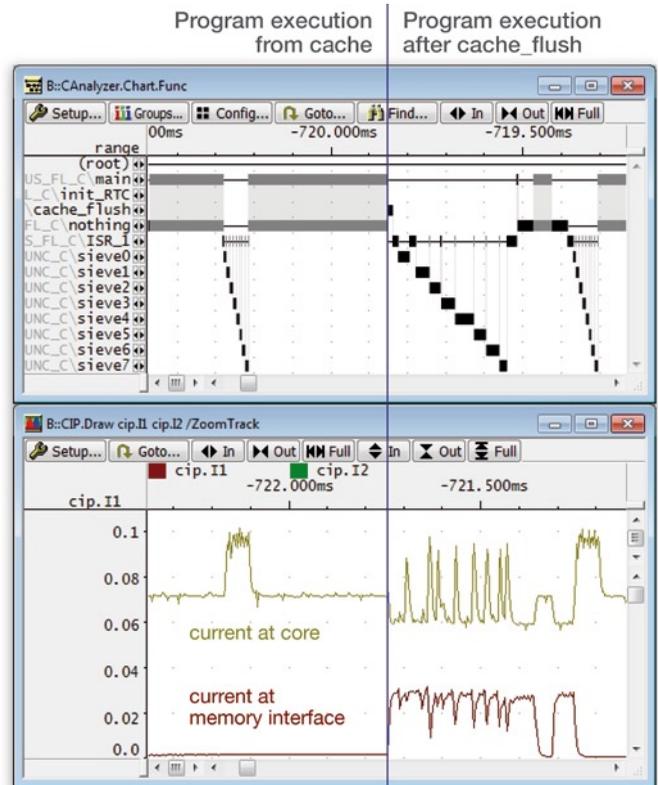


Fig. 10: Une partie de l'application s'exécutant en mémoire externe nécessite plus de temps et consomme plus d'énergie.

porelle entre le code exécuté et le courant ou la tension absorbée par le système.

La figure 10 montre que la partie de l'application s'exécutant en mémoire externe au lieu de s'exécuter en cache ne nécessite pas seulement un temps de traitement supérieur mais consomme aussi beaucoup plus d'énergie.

La figure 11 montre la consommation d'énergie sous forme statistique. La consommation d'énergie minimale, maximale et moyenne est affichée pour les différentes fonctions.

range	total	min	max	avr	count
(root)	163.554mJ	-	163.554mJ	163.554mJ	1. (0/1)
sinus_FL\SINUS_FL_C\main	163.540mJ	-	163.540mJ	163.540mJ	1.
s_FL\SINUS_FL_C\init_RTC	24.646uJ	24.646uJ	24.646uJ	24.646uJ	176.
User\Global\cache_flush	1.578mJ	8.732uJ	15.077uJ	8.964uJ	1079368.
us_FL\SINUS_FL_C\nothing	104.771mJ	0.000	45.392uJ	97.066mJ	1755.
flus_FL\SINUS_FL_C\ISR_I	98.184mJ	20.229uJ	346.980uJ	55.945uJ	1755.
FI\SIEVE0_FUNC_C\sieve0	9.865mJ	1.947uJ	32.718uJ	5.621uJ	1755.
FI\SIEVE1_FUNC_C\sieve1	11.368mJ	2.536uJ	39.485uJ	6.478uJ	1755.
FI\SIEVE2_FUNC_C\sieve2	9.063mJ	2.410uJ	28.901uJ	5.164uJ	1755.
FI\SIEVE3_FUNC_C\sieve3	10.937mJ	2.574uJ	38.119uJ	6.232uJ	1755.
FI\SIEVE4_FUNC_C\sieve4	13.244mJ	2.640uJ	50.595uJ	7.547uJ	1755.
FI\SIEVE5_FUNC_C\sieve5	10.098mJ	2.504uJ	34.293uJ	5.754uJ	1755.
FI\SIEVE6_FUNC_C\sieve6	9.912mJ	2.393uJ	33.921uJ	5.648uJ	1755.
FI\SIEVE7_FUNC_C\sieve7	7.620mJ	2.576uJ	19.322uJ	4.342uJ	1755.

Fig. 11: La consommation minimale, maximale et moyenne des différentes fonctions.

Le CombiProbe

Le **CombiProbe** est une sonde de débogge équipée d'une mémoire de trace de 128 Mo. Le **CombiProbe** a été spécialement développé pour des processeurs possédant un port de trace de 4 bits. L'enregistrement du flux du programme inclut actuellement les protocoles de trace suivants :

- ARM-ETMv3 en mode continu (ARM)
- IFLOW Trace pour PIC32 (Microchip)
- MCDS Trace pour X-Gold102 et X-GOLD110 (Infineon)

www.lauterbach.com/cobstm.html

Le forum des experts



Fig. 12: Les détails techniques sont abordés autour d'une discussion avec nos experts.

Forum des experts ARM

A deux reprises l'année dernière, clients et professionnels intéressés ont eu l'occasion de discuter avec nos experts TRACE32 sur le débogge des cœurs ARM. Un succès immédiat et une réservation très rapide sur le forum du 5 Mai 2010 nous ont conduits à organiser un second forum, les 24 et 25 Novembre. Ce dernier a été proposé cette fois en langue anglaise afin de permettre à tous nos clients d'y participer.

Lors des deux événements, les participants ont eu l'opportunité d'obtenir des informations sur tous les aspects de débogge et de trace des cœurs ARM. Nos experts ont abordé très précisément l'interaction des différentes composantes ainsi que les différentes caractéristiques de chaque CPU, mais aussi les possibilités de connections aux différents ports de débogge et de trace pour obtenir une analyse optimale des données de la trace.

Une des sessions était dédiée au débogge de système multi-cœurs. En utilisant un exemple basé sur Linux, nous avons pu définir les différentes exigences auxquelles un système de débogge moderne doit faire face aujourd'hui.

Cette présentation a été complétée avec les possibilités de débogge d'Android.

L'intervention de notre invité M. Ian Johnson (ARM Ltd.) a été un moment fort du forum des experts dont la présentation détaillait les processeurs ARM et a surtout apporté d'intéressantes perspectives de développement.

Forum des experts automobile

Les réactions positives des participants aux deux dernières éditions organisées, ainsi que la forte demande pour renouveler ces journées, nous ont montré l'importance des manifestations au cours desquelles les utilisateurs et les développeurs de nos systèmes de débogge ont l'opportunité d'échanger leurs connaissances.

Forum des experts automobile le 18 Mai 2011

Aperçu des produits TRACE32

Développement de système AUTOSAR/avec TRACE32

Session TriCore

- Les News
- Débogge "concurrent" entre TriCore et PCP
- MCDS/Langage de Trigge OCTL
- Analyse de performance
- Q&R avec nos experts

Session Power Architecture

- Les News
- Débogge Multi-cœurs/Trace
- Analyse de performance basée sur le NEXUS
- Q&R avec nos experts

D'autres éditions du forum des experts seront donc organisées en 2011. La première en date est fixée au 18 mai 2011. Elle sera consacrée autour des solutions de débogge spécifiques au domaine de l'automobile. Cet événement inclura entre autres les processeurs TriCore d'Infineon, les MPC55xx/MPC56xx de chez Freescale et les familles de processeurs SPC56xx de chez ST. L'un des sujets majeurs abordés sera le débogge avec AUTOSAR. Re-trouvez ci-dessous un aperçu des autres sujets qui seront traités et nous vous invitons également à consulter notre site web pour plus d'informations.

FILIALES À TRAVERS LE MONDE



- France
- Allemagne
- Grande-Bretagne
- Italie
- États-Unis
- Chine
- Japon

Représentés par nos partenaires dans tous les autres pays.

BULLETIN D'INFORMATION

Si vous souhaitez dorénavant recevoir notre bulletin d'information par voie postale, veuillez nous envoyer vos coordonnées sur l'adresse e-mail suivante :

info_fr@lauterbach.com

Une désinscription est possible à tout moment.