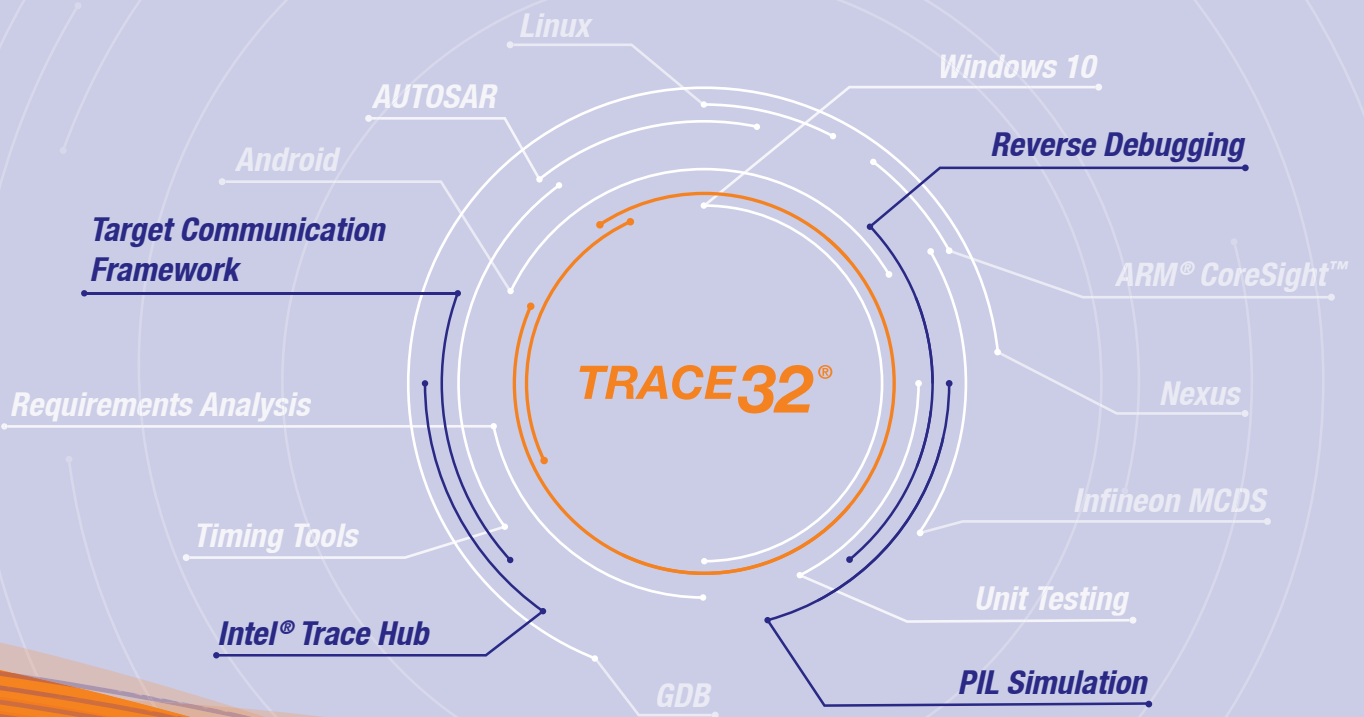


NEWS 2016



SOMMAIRE

Simulation PIL avec TRACE32	2
Nouveaux modules de base	4
PowerTrace Série	5
Support pour Intel® Trace Hub	6
TRACE32 possède un Agent TCF	7
Wind River Workbench	
Eclipse	
TRACE32 le Front-End pour UndoDB	8

Simulation PIL avec TRACE32

Courant mars 2016, Lauterbach proposera son plug-in Simulink pour la simulation PIL (Processor-in-the-Loop). Grâce à ce nouveau plug-in, l'environnement de modelage pourra communiquer directement avec la cible via le débogueur TRACE32.

Ces dernières années, les méthodes basées sur des modèles sont devenues de plus en plus importantes dans le développement logiciel. L'avantage de ces méthodes basées sur des modèles est la vérification continue du design logiciel. La simulation PIL (Processor-in-the-Loop) est désormais une étape importante dans la vérification de Design.

Simulation PIL

La simulation PIL permet de s'assurer que les algorithmes développés fonctionneront comme prévu sur cible. Cette vérification peut être réalisée sur les cibles suivantes :

- Cible matériel finale / carte d'évaluation
- Cible Virtuel / Simulateur de Cœur
- Simulateur du jeu d'instruction TRACE32

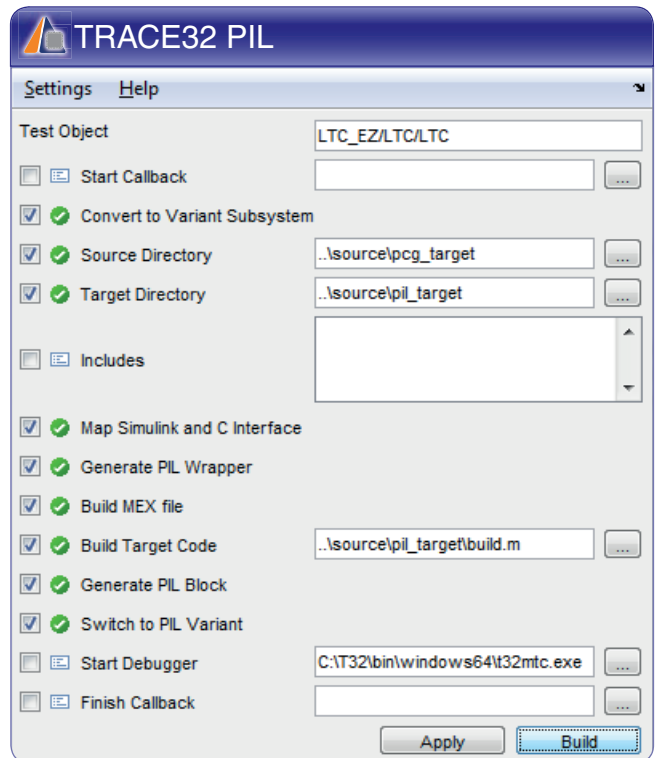
Pour exécuter une simulation PIL, l'algorithme à tester doit être remplacé par un block PIL dans le modèle d'environnement.

TRACE32 PIL

Le plug-in TRACE32 PIL (voir copie d'écran sur cette page) est utilisé pour configurer la simulation PIL. La séquence de configuration la plus importante est présentée rapidement ci-dessous.

Checkbox: Map Simulink and C Interface

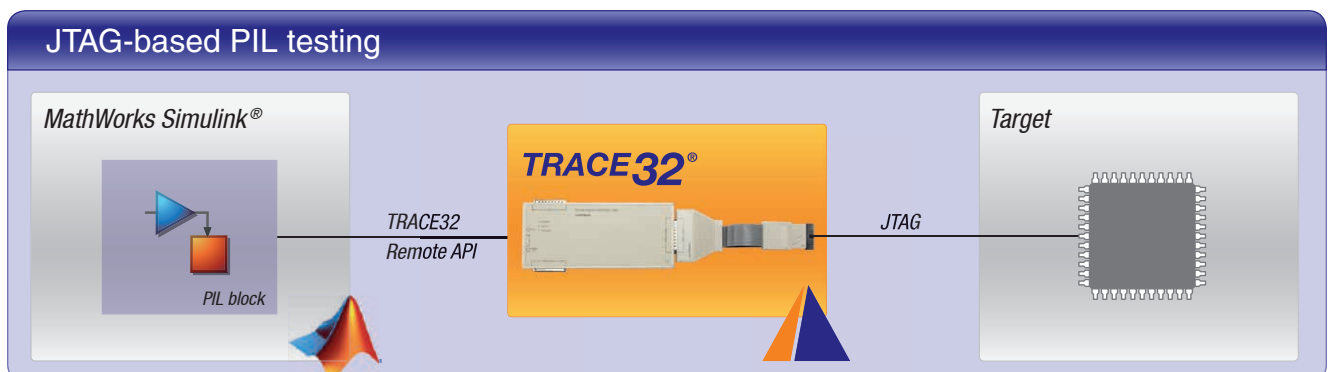
L'interface entre Simulink et l'application cible doit être



configurée avant la génération du block PIL. Pour des raisons de configuration, la fenêtre de dialogue Mapping GUI (voir copie d'écran en page 3) générera automatiquement des suggestions de Mapping entre les callbacks des fonctions S (Block défini par l'utilisateur) et les fonctions C correspondantes. Ces suggestions peuvent être validées ou corrigées dans la boîte de dialogue. Le même Mapping devra aussi être créé entre les paramètres modèles et les variables C.

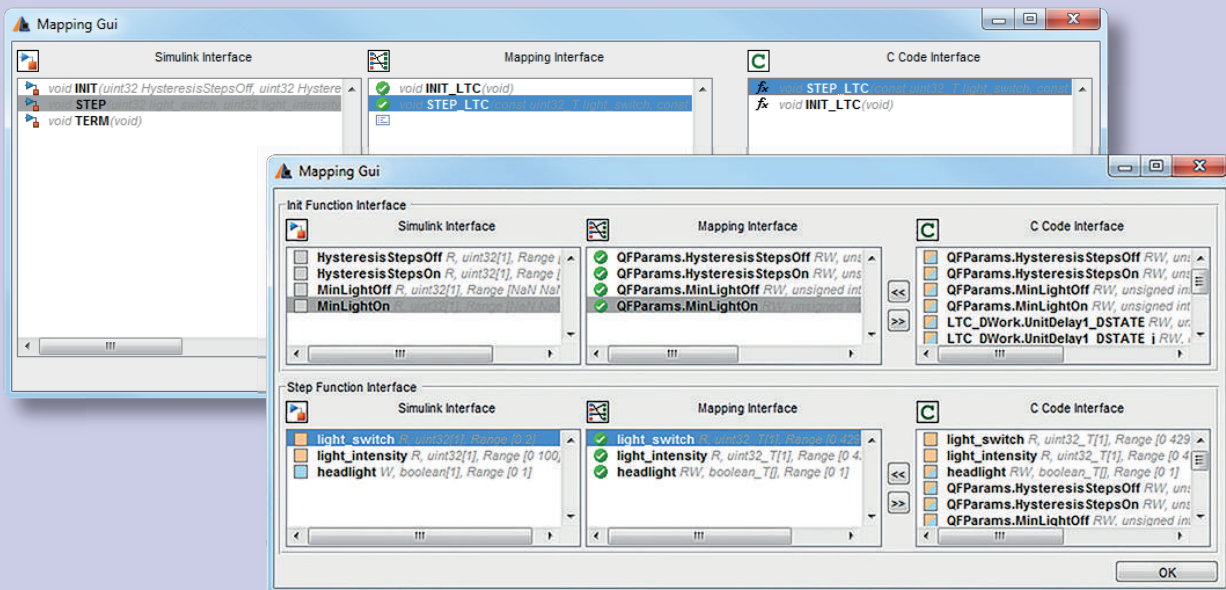
Checkbox: Generate PIL Wrapper

Basée sur le Mapping des callbacks et des paramètres du modèle, l'interface suivante est créée automatiquement dans la séquence de configuration suivante :



Mapping Simulink and C Interface

Mapping Simulink callbacks → C functions



Mapping Simulink parameters → C variables

- implémentation des callbacks sur la cible du système
- interface du modèle vers la Remote API TRACE32

Ensuite, le code exécutable peut être créé.

Checkboxes:

Generate PIL Block and Switch to PIL Variant

Après avoir créé la fonction d'interface entre le modèle et la Remote API TRACE32, le block PIL peut enfin être créé et affiché dans le modèle de simulation. La configuration de la simulation PIL se termine par le démarrage de TRACE32.

Avantages

1. Toutes les méthodes de générations de code peuvent être utilisées.

À travers la possibilité de configurer dynamiquement l'interface pour les callbacks, la solution Lauterbach peut être utilisée avec tous les générateurs de code ainsi qu'avec du code créé manuellement.

2. Adaptation Direct avec les nouvelles cibles système

TRACE32 supporte une très large gamme d'architectures processeurs et de compilateurs. Les scripts de démarrage de la cible système peuvent être créés directement par le développeur. Vous n'avez pas besoin d'attendre une mise à jour d'aucun fournisseur d'outils. A partir du moment où notre débogueur est capable de communiquer avec la cible, les bases de la simulation PIL sont prêtes.

3. Debugge immédiat

Si le résultat de la simulation ne correspond pas à ce qui était attendu, il est parfaitement possible de déboguer directement les fonctions C.

4. Licence

Les développeurs qui utilisent déjà les sondes de débogage TRACE32 ou bien sur cible virtuelle avec les licences flottantes TRACE32 n'auront besoin que de la licence TRACE32 PIL Simulation. Les clients qui voudront utiliser le simulateur de jeu d'instruction TRACE32 pour la simulation, auront besoin de la nouvelle licence du Simulateur TRACE32.

Mise en place des nouveaux modules de base, finalisée en 2015

Tous les modules PowerDebug TRACE32 sont désormais équipés avec l'interface USB3.

Les modules PowerDebug PRO proposent en plus une interface Gigabit Ethernet ainsi qu'une interface PodBus Express qui permet la connexion des mo-

dules TRACE32 PowerTrace PX (nouveau) ou bien TRACE32 PowerTrace II. Les anciennes générations de modules de base peuvent toujours être utilisées sur les nouvelles architectures processeurs sans aucune restriction.

PowerDebug USB 3

Previous generation PowerDebug USB 2



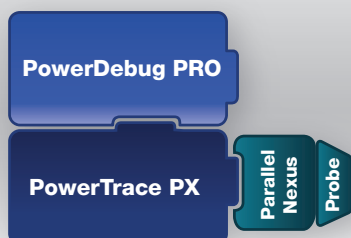
PowerDebug PRO

Previous generation PowerDebug Ethernet or PowerDebug II



PowerDebug PRO + PowerTrace PX

Previous generation PowerTrace Ethernet

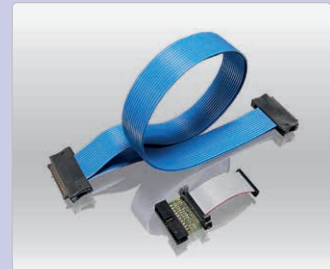


PowerDebug PRO + PowerTrace II

Previous generation PowerDebug II + PowerTrace II



PowerTrace Série



TRACE32 Accessory Set

Le nouveau module de base TRACE32 PowerTrace Série sera disponible à la livraison début Q2/2016.

Lauterbach supporte les interfaces de trace série depuis 2008 avec une sonde de trace série supportant de nombreuses architectures. La sonde actuelle possède les caractéristiques techniques suivantes :

- jusqu'à 4 canaux RX
- 6.25 Gbit/s par canal jusqu'à 3 canaux
- 3.125 Gbit/s par canal jusqu'à 4 canaux
- pour des protocoles de trace utilisant Aurora

Sur les Micro-processeurs actuels, la bande passante des interfaces de trace série n'est pas toujours suffisante pour permettre au développeur de pouvoir visualiser les opérations internes complexes d'un système Multicoeur. C'est pourquoi, certains fabricants de Micro-processeurs, sont en cours d'intégration d'interfaces, possédant de très hauts débits de transferts de données sur plusieurs canaux. En parallèle le PCI Express est couramment mentionné pour être utilisé comme une interface d'exportation de la trace série.

Fonctionnalités du PowerTrace Série

Le module de base TRACE32 PowerTrace Série a été développé pour répondre aux dernières exigences du marché.

- jusqu'à 8 canaux
- jusqu'à 12.5 Gbit/s par canal
- Xilinx Aurora et bien d'autres protocoles supportés, mais tout particulièrement le PCI Express
- 4 giga de mémoire de trace

Comme toutes ces fonctionnalités ne peuvent être implémentées que dans un FPGA beaucoup plus puissant, le nouveau PowerTrace Série a été pensé comme une solution tout-en-un.

Ce qui implique que la sonde de trace série, disponible auparavant comme une sonde séparée, a été intégrée dans le PowerTrace Série. Lauterbach offre désormais divers convertisseurs permettant de connecter le PowerTrace Série à la cible finale. Le set d'accessoires standard se composera d'une nappe flexible munie d'un adaptateur spécifique. Le PowerTrace Série intègre une licence par architecture spécifique, lui permettant de décoder les informations de trace d'un cœur. Le décodage de la trace d'une autre architecture pourra se faire très simplement par ajout de la licence de l'architecture concernée.

Connecteurs PowerTrace Série

Le PowerTrace Série est équipé des interfaces suivantes :

Port de trace série 0 (connecteur Samtec ERF8, 40-pin)

- pour des protocoles de trace utilisant Aurora
- 6 canaux RX
- fréquences de référence 0.325 - 6.25 GHz

Port de debugge cible (connecteur 34-Pin MIPI)

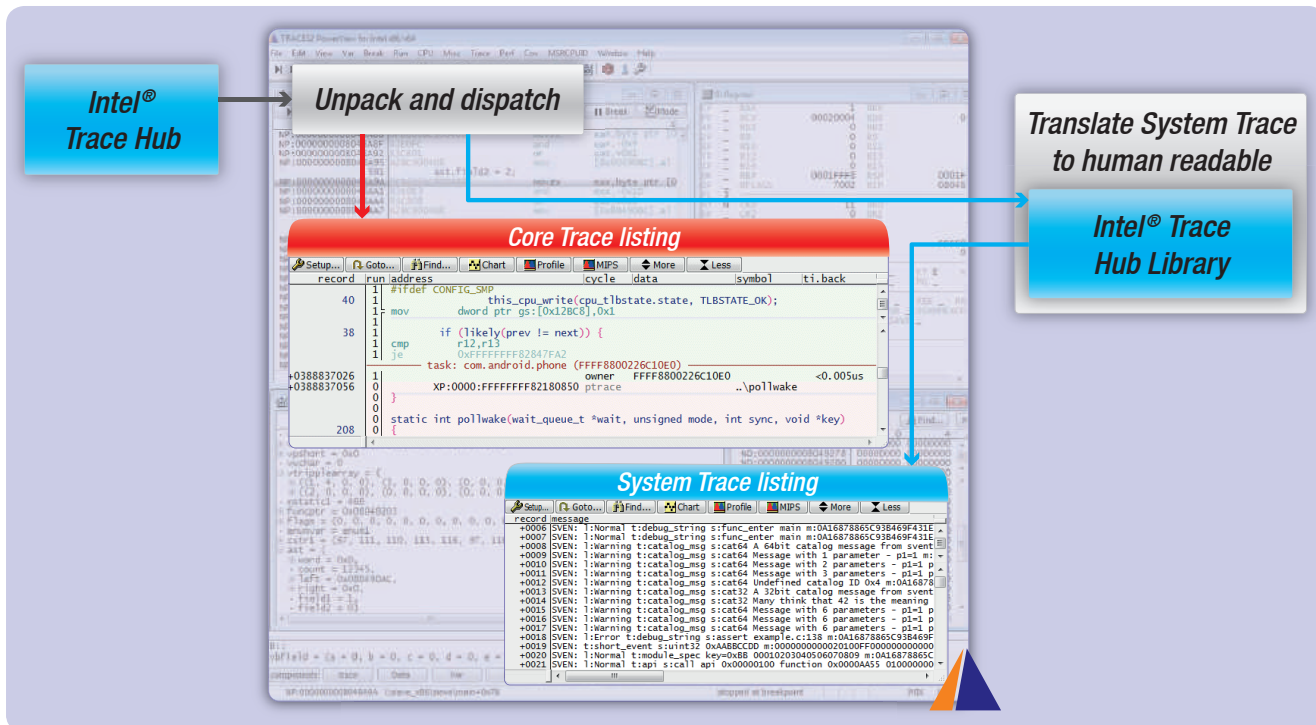
Si les signaux de debugge (JTAG/SWD/cJTAG) sont aussi routés sur le connecteur Samtec 40-pin, alors un câble de debugge TRACE32 pourra aussi se connecter ici.

Port de trace série 1 (connecteur Samtec ERM-ERF, 80-pin)

- pour tous les protocoles
- 8 canaux RX / TX
- fréquences de référence 0.325 - 6.25 GHz ou 10 - 500 MHz

En proposant ces deux ports de trace, le PowerTrace Série peut être utilisé dans les développements actuels mais aussi pour les futurs projets.

Support pour Intel® Trace Hub



En mai 2016, TRACE32 supportera l'Intel® Trace Hub et ses Framework associés.

Intel® Trace Hub

Intel® Trace Hub (Intel® TH) c'est le nom donné à l'infrastructure de trace qu'Intel propose dans ses nouvelles plateformes matériel. Cette infrastructure de trace permet notamment :

1. de fournir un timestamp commun aux traces de donnée Intel® Processor pour chaque cœur et une information de Trace System pour une multitude de sources.
2. de fusionner toutes les données dans un seul flux de trace utilisant le protocole MIPI STPv2.1.
3. de transmettre ce flux vers une trace sélectionnée de destination.

Pour permettre aux outils de debugge et de Trace comme TRACE32 de supporter cet infrastructure facilement, Intel fourni pour cela un logiciel Framework.

API de configuration Intel® Trace Hub

Le but de l'API de configuration Intel® Trace Hub est de simplifier la configuration de l'infrastructure de trace à travers l'outil de debugge. TRACE32 n'a pas besoin de connaître les séquences de programmation spécifiques

à la plateforme, mais peut simplement envoyer une requête de configuration à l'API de configuration Intel® TH. L'API va alors proposer la bonne séquence de programmation, que TRACE32 copiera dans les registres de contrôle via l'interface JTAG.

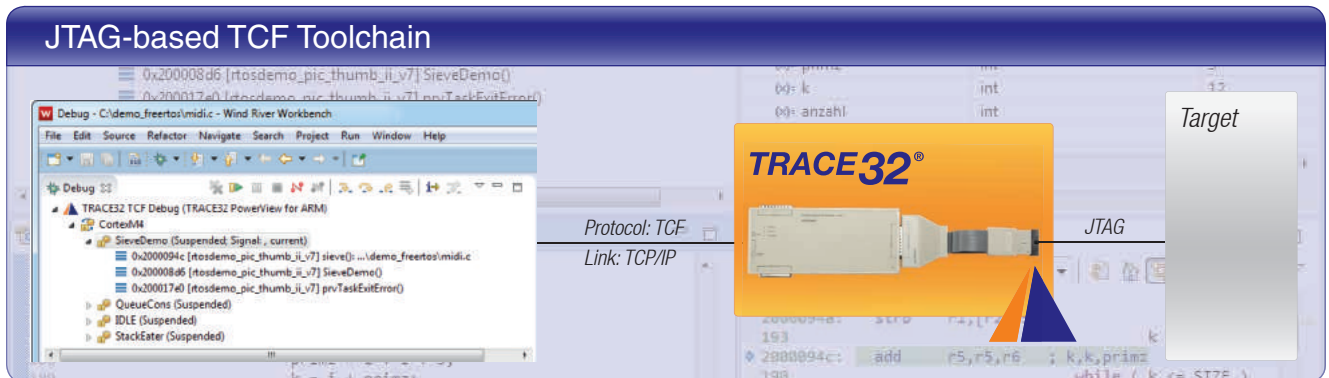
Librairie Intel® Trace Hub

Le but de la Librairie Intel® TH est de convertir les paquets de trace systèmes en messages de trace compréhensibles par le développeur. Comme Intel® TH a fusionné tous les messages de trace en utilisant le protocole MIPI STPv2.1, ils doivent en premier lieu être séparés. Pour ensuite être distribués pour le décodage (voir figure ci-dessus) :

- les paquets de Trace Intel® Processor restent dans TRACE32, où ils seront directement décodés pour être affichés et évalués.
- les paquets de Trace System sont eux dirigés vers la Librairie Intel® TH. Les messages de trace décodés et reçus en retour peuvent être ensuite affichés et évalués directement dans TRACE32 grâce à des commandes dédiées.

Le simple fait d'avoir une vue disponible corrélée dans le temps de toutes les informations de trace, affichée dans TRACE32, donne une vue complète sur l'activité de la plateforme.

TRACE32 possède un Agent TCF



Depuis octobre 2015, TRACE32 peut être utilisé aussi comme un Agent TCF. Cela permet d'utiliser le Workbench Wind River ou Eclipse comme IDE et TRACE32 comme outil de debugge côté cible.

TCF

Le *Target Communication Framework* (TCF) a été développé par l'Eclipse Fondation comme un Protocole Framework dans le but de définir un protocole de communication universel entre les différents IDE et une cible système.

Le modèle de communication TCF est basé sur l'idée de service. Un service est un groupe de commandes, d'événements, avec leur sémantique correspondante. Le service Memory, par exemple, définit le groupe de commandes et d'événements permettant la lecture et l'écriture mémoire.

TCF définit une série de services standards. Le Framework étant totalement ouvert, il permet la définition de services propriétaires.

TRACE32 TCF

Après avoir démarré TRACE32 comme un agent TCF, il propose ses services au Workbench Wind River ou au débogueur Eclipse via TCP/IP.

La requête de services est assurée par TRACE32 avec l'aide du débogueur connecté sur la cible. Cela importe peu qu'il s'agisse d'un débogueur physiquement connecté par JTAG à la cible, où bien d'un débogueur purement logiciel communiquant avec une cible virtuelle.

A ce moment précis, TRACE32 offre tous les services traditionnels assurant le debugge. Selon le retour ou les demandes de nos clients, le développement

de services spécifiques TRACE32 dans le cadre de fonctions étendues pourrait être planifié. Lauterbach offre son plug-in TRACE32 TCF pour les clients qui souhaitent configurer et démarrer TRACE32 à partir du Workbench Wind River ou du débogueur Eclipse.

Wind River Workbench

Jusqu'à présent, Lauterbach ne proposait aucune solution de debugge intégré pour les développeurs qui préféraient travailler avec le workbench Wind River, cette limitation n'existe plus désormais.

Débogueur Eclipse

La version précédente de notre connexion basée sur GDB était limitée aux architectures processeur et compilateur supportées par GDB. Désormais, grâce à l'utilisation des services TCF, TRACE32 propose une interface de communication ouverte pour debugger avec Eclipse, et cela pour toutes les architectures processeur et compilateur supportées par TRACE32.

TRACE32 Support for Wind River

VxWorks 5/6/7

VxWorks 653 2.x

VxWorks 653 3.x

Wind River Linux

Wind River Hypervisor 2.x

VxWorks Microkernel Profile

WIND RIVER

TRACE32 le Front-End pour UndoDB

TRACE32 peut être utilisé comme front-end sur le débogueur réversible UndoDB depuis la mi-2015. Les architectures ARM/Cortex et Intel® x86/x64 sont actuellement supportées.

Grâce au serveur cible UndoDB, les développeurs Linux possèdent désormais un outil qui leur permettent de déboguer une application user et même d'enregistrer tous les détails de son exécution. En parallèle de pouvoir contrôler le debugge d'un process, le front-end TRACE32 permet aussi de traiter et d'afficher les données enregistrées par le serveur cible UndoDB tout cela dans l'interface graphique TRACE32.

Exactement comme l'enregistrement d'une TRACE, le développeur peut déboguer son application en

avançant ou reculant dans l'exécution de son code ("reverse debugging"). L'utilisation de cette fonctionnalité permet de trouver plus facilement et plus rapidement des erreurs dans son application process. Pour permettre de visualiser graphiquement l'activation du debugge des données, le temps d'enregistrement affiché dans statusline de TRACE32 devient négatif. De même que, les boutons de debugge dans la fenêtre source-list sont affichés en jaune. L'interface graphique TRACE32 bascule automatiquement dans le mode d'affichage suivant.

- Dans la fenêtre source-list, le pointeur d'instruction prend la valeur qu'il avait au moment de l'enregistrement.
- Le contenu de la mémoire et des variables suivent l'enregistrement.

The screenshot shows the TRACE32 GUI with several key features highlighted:

- Forward & backward debugging:** The source list window shows assembly code with a yellow highlight on the instruction `reverse (44)` at address `ZUT:0A73:000093BC`. The status bar at the bottom indicates `reverse (44)`.
- Variable reconstruction:** The 'Variables' window shows the state of variables like `static1`, `static2`, `rcount`, and `array`.
- Recording point displayed:** A blue arrow points to the `reverse (44)` instruction in the source list.

Protocol: GDB
Link: TCP/IP

Target

UndoDB server

Notre adresse e-mail a changé ou vous ne recevez plus notre bulletin d'information, dites-le nous à l'adresse suivante:

info_fr@lauterbach.fr



LEADING through Technology