The Debugging Evolution

KMC
Member of eSOL

新たなデバッグ時代を切り拓く第三世代 JTAGエミュレータ

PARTNER Jet3

マルチコア・対応
マルチOS 対応

Arm® CPU : 64bit/ 32bit 対応
RISC-V CPU : 64bit/ 32bit 応

京都マイクロコンピュータ株式会社

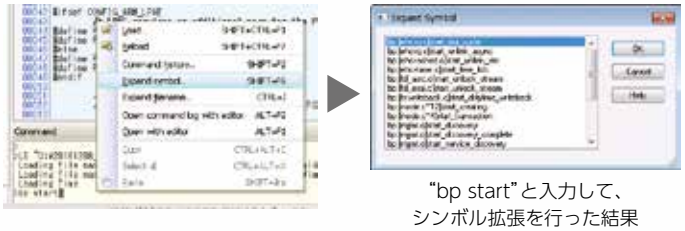
https://www.kmckk.co.jp/

PARTNER Jet3 デバッガの主な機能

ソースコードデバッガ

ビルド・ロードしたプログラムは、ソースコード上で実行指定やプログラムカウンタの移動、ソースコード行のクリックでブレークポイント設定などの操作が可能です。またソースコード上の変数をクリックするだけで変数の値をインスペクト表示します（C/C++/Rust 言語に対応）。
さらにコーディングやデバッグに便利な以下の機能も備えています。

- 実行プログラムをロードした後で、シンボル情報などのデバッグ情報だけを追加で読み込むことができます（複数読み込み可）。
- シンボル名の入力途中で候補を表示し選択できる、入力補完機能があります（シンボル拡張、下図）。
- プリプロセッサマクロ展開結果をデバッガ表示（対応コンパイラのみ）。
- 連続ステップ実行した履歴をたどることができます。
- 関数呼び出し履歴表示および、履歴に表示された関数内でのローカル変数の値を連動表示します。

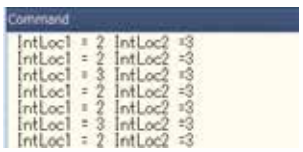


“bp start”と入力して、
シンボル拡張を行った結果

ソフトウェアブレークとハードウェアブレーク

デバッガの基本機能であるブレーク機能として、ソフトウェアブレークとハードウェアブレークがあります。
S/W ブレークは指定した箇所の命令をブレーク専用の命令に置き換えて、プログラムの実行を停止させます。PARTNER デバッガでは、S/W ブレークが発生したときにマクロの実行、コマンド実行や printf のようなメッセージをデバッガのウィンドウに出力する機能があります（下図）。
H/W ブレークは、デバッグ対象のプロセッサーのオンチップデバッグ機能を使ってプログラムを停止させる機能です。アドレス条件、データ条件やリード・ライト条件などを指定（CPU により条件が異なります）することで、不正ライトによるメモリ破壊などの検出に役立ちます。

printf ブレークの設定



ブレークの度に設定した
プリント文をコマンドウィンドウに
表示し、自動的に再実行する。

スクリプト機能

デバッグ時の作業をプログラマブルに実行するためのスクリプト機能を搭載しています。C 言語に似た構文で使いやすい Squirrel 言語の他に、多く使われている Python にも対応しており、利用目的や、好み・慣れで使い分ける事ができるようになっています。

USBバスパワー動作

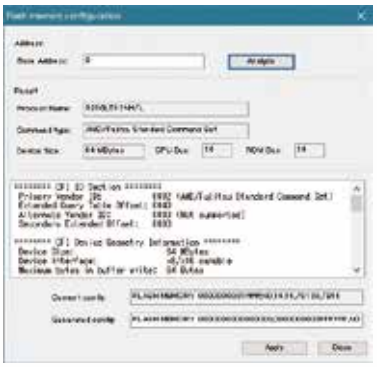
PARTNER-Jet3 は、USB3.2Gen1 または USB2.0 でホスト PC と接続することで、外部電源無しのバスパワー動作が可能です。

実行時間簡易測定機能

プログラムがブレークポイントで停止したとき、実行開始から停止までの時間を自動的に測定し、表示します（100μs単位）。

フラッシュメモリ対応

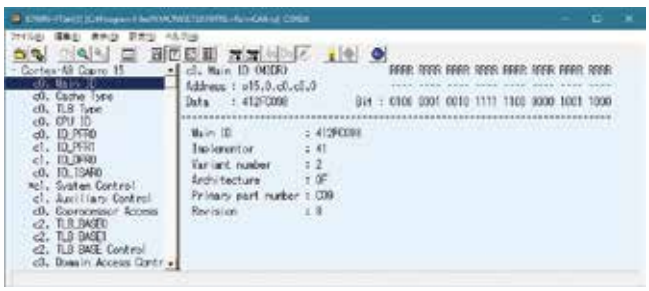
各種 NOR フラッシュメモリの書き込みに対応し、また NOR フラッシュ上を XIP 実行するプログラムに対してブレークポイントの設定も可能です。
SPI 接続した NOR フラッシュメモリについては、KMC が提供する SDK で書き込み用のモニタプログラムを作成することにより、標準サポートのパラレル接続の NOR フラッシュと同様に書き込みができます。HyperFlash™ の場合もパラレル NOR フラッシュと同等の書き込み機能を利用することができます。
パラレル NOR フラッシュと HyperFlash™ では CFI からフラッシュの情報を認識することで書き込みに必要な設定を自動的に行う機能があります。



フラッシュメモリの自動認識

I/Oレジスタ機能

周辺系 I/O のレジスタを定義したファイルをロードすることにより、レジスタの名前やビットアサイン、ビットの意味などを分かりやすい GUI で表示したり、ビットフィールド単位で値の変更が可能です。定義ファイルはテキストエディタ等で作成できます。
PARTNER-Jet3 独自の IO 定義ファイル以外に Arm CMSIS-SVD ファイルにも対応しています。



I/O レジスタの表示

マクロ機能

デバッガのコマンドライン操作をバッチファイルにして、一括で自動実行するマクロ機能が使用できます。マクロ機能ではデバッガの操作コマンドだけでなく、シンボルの値を評価（判断）した条件判定や、繰り返し実行のような制御構文を使った高度なマクロ実行が可能です。

PARTNERは、開発エンジニアの皆様と共に進化を極め、ソースコードデバッグを行う時に必要な機能を豊富に備えたデバッガです。ハードウェアの立ち上げ段階からソフトウェアの性能評価まで、開発の全てのステージでエンジニアを支援します。またデバッガ操作中のエンジニアの思考がすぐに行動に移せるよう、GUIとCUIを統合したスムーズで使いこなしやすいユーザインタフェースに仕上げています。

リアルタイムトレース

Model 15

高速トレース

高速でデリケートなトレース入力信号はPARTNER-Jet3のキャリブレーション機能でスキュー調整を行うことで、Mictor-38pinプローブ使用時に最大400Mbps/端子(200MHzDDR駆動、8 or 4bit幅)まで対応可能です。

大容量トレース

Model 15では512Mバイトのトレースメモリを備えており、大容量トレースデータを取得することができます。

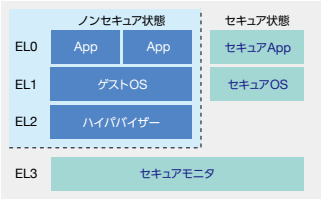
デバッグ対象CPU

PARTNER-Jet3では、Armプロセッサと RISC-Vプロセッサに対応しています。

- Armv9、Armv8をはじめとしたArm Cortexプロセッサ(32/64bit、シングルコア/マルチコア)
- RISC-V プロセッサ(32/64bit、シングルコア/マルチコア)

Arm-v8 64bit アーキテクチャ(AArch64以降のプロセッサ)に対しては、新たに定義された4階層のException Level(EL0 ～ EL3)とセキュア/ノンセキュア状態での動作がデバッグしやすいよう、デバッガ機能を拡張しています。

- H/Wブレーク条件に、Exception Levelとセキュア状態を設定可能
- H/Wブレーク条件に、仮想マシンIDを指定可能
- 指定したException Levelへの遷移をブレーク条件に指定できる、Exceptionキャッチブレークを新設

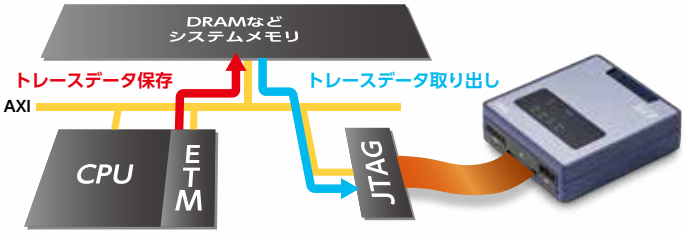


AArch64の動作モード例

CoreSight ETR 対応のトレース

Arm向け応用機能

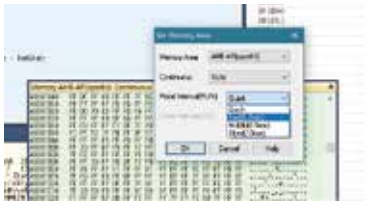
Arm CoreSight ETR(組み込みトレースルーター)に対応したトレース機能が使用できます。この機能では、Arm CortexプロセッサのETM(組み込みトレースマクロセル)から出力される実行命令トレース情報をターゲット上のDRAMのような大容量のシステムメモリに保存し、デバッガでトレース情報を参照するときには、JTAG信号経由でシステムメモリからトレース情報をホストPCに転送します。
ターゲットシステム上にETMトレース用のコネクタを実装しない場合や、SoCのトレース端子が兼用機能でトレース用途に利用できない場合に有効なトレース取得方法です。



RAMモニタ対応

Arm向け応用機能

- CoreSight AXI-AP(AHB-AP)を利用したCPUを介さないメモリアクセスにより実行中のRAMの内容を、メモリウィンドウでリアルタイムに表示できます。
- こんな時に有効です。
  - ・停止できないメカ制御でのパラメータ領域の監視ができます
  - ・PARTNER-Jet3のプローブホットプラグ機能と組み合わせることで、長時間稼働中のシステムをリセットやブレークさせることなくデバッガを接続して、データを監視することが可能です。

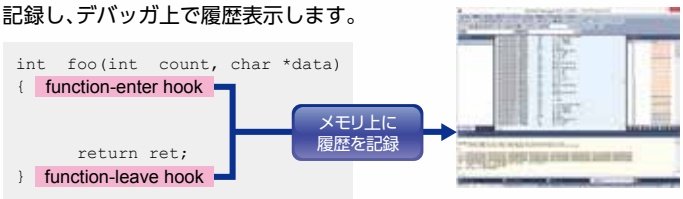

RAMモニタ機能の設定


変化した値をハイライト表示

埋め込みトレース(トレースポイント)

Model 15

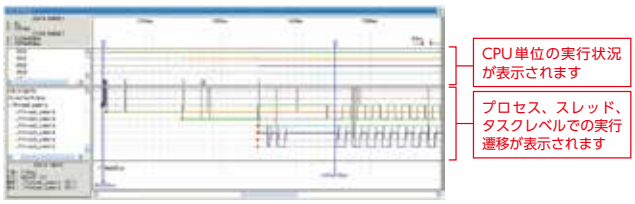
トレース信号を持たないターゲットハードウェアに対して、ターゲット上のメモリを使いJTAG接続だけで関数レベルの実行履歴を取得する機能です。
関数の入口/出口にトレースフックルーチンを埋め込み、そのフックルーチン通過の際に出力されるデータをターゲットシステム上のメモリに記録し、デバッガ上で履歴表示します。



PARTNERデバッガ

イベントトラッカー

タスクやスレッドの遷移をGUIで表示し、OSを使ったソフトウェアの動作解析を容易にします。イベント情報はターゲットメモリ内に保存する他、Model15ではリアルタイムトレースとの連携や、埋め込みトレースとの連携も可能です。



タスクやスレッドの遷移

プローブホットプラグ対応・電源監視機能

Arm向け応用機能

プローブホットプラグ対応

ターゲットの通電・動作中であっても、デバッグプローブの挿抜が可能です。テスト中に異常が発生した時点でターゲットにデバッグプローブを接続しデバッグが開始できるので、障害再現に時間をとられず効率よく作業ができます。

電源監視機能

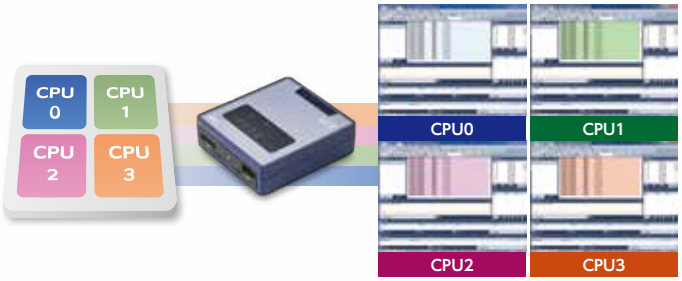
JTAGデバッグ中にターゲットボードの電源断になっても、デバッガがそれを認識し、電源再投入時に自動的にデバッガの再接続が可能です。再接続時には、設定しておいたハードウェアブレークなどを復元するので、電源断～再投入の経過をデバッグする際に有効な機能です。

マルチコアデバッグ

PARTNER-Jet3のマルチコアデバッグは、高速排他制御と仮想化技術で、
シングルコアデバッグの様に、高速でストレスのないデバッグ環境を提供します。

AMPデバッグ

マルチコアSoC において 各プロセッサーが異なるOSを実行したり 、独立した処理を行う場合には、プロセッサーごとに別々のデバッガウィンドウを起動してデバッグができます。まるで、プロセッサーごとに別々のJTAGデバッガを接続しているかのような独立した操作が可能です。

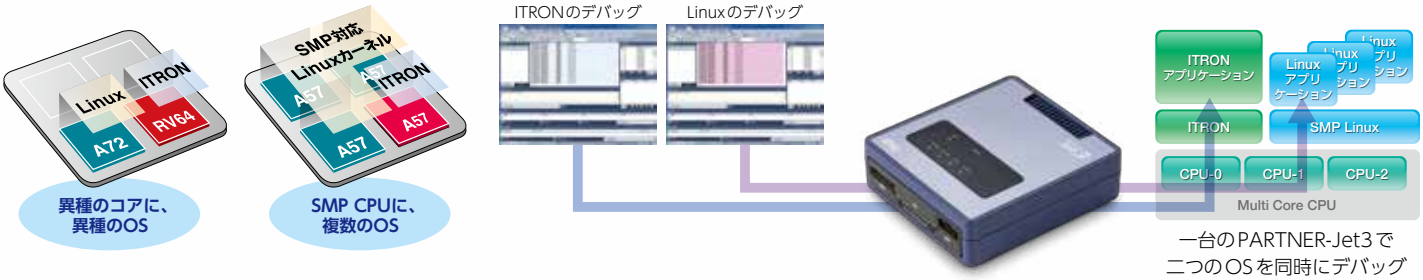


SMPデバッグ

SMP Linuxなど、OSが複数のプロセッサーをコントロールする場合には、1つのデバッガウィンドウで、対象の複数プロセッサーをまとめてデバッグできます。こうすることで、SMP OS上で動作するプログラムがプロセッサー間を移動しても、ユーザは、それを意識せずにデバッグができます。その他に、SMPを構成する複数のプロセッサーと別のプロセッサーをAMP構成としたシステムや、複数のSMPから構成されるシステムのデバッグも可能です。

異なるコア/異なるOSに対応

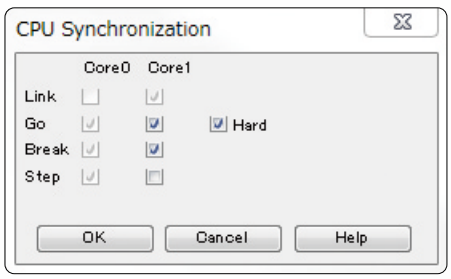
AMP構成のシステムにおいては、Cortex-A53とCortex-R5のように、種類が異なりかつAArch64/AArch32のようにアーキテクチャも異なる複数のプロセッサーを同時にデバッグできます。
また、LinuxとITRONのように異なるOSや、異なるコンパイラで作成されたプログラムも同時にデバッグできます。
さらに、Cortex-A53 QuadコアのSMP Linuxと、Cortex-R5のリアルタイムOSの様に、SMPグループとAMPのプロセッサーとの組み合わせも、同期・非同期でのデバッグができます。



一台のPARTNER-Jet3で
二つのOSを同時にデバッグ

同期デバッグ

AMP構成の場合、各プロセッサーを完全に独立して動作させる、もしくはgo/breakを同期させた状態でのデバッグが選択できます。(SMP設定のプロセッサーでは、常に同期設定で動作します)
PARTNER デバッガではCoreSight CTI などのコア間同期機能に対応しているので、クロックレベルでの同期実行・ブレークが可能です。
デバッグ対象のハードウェアに同期機能が無い場合は、ソフトウェアでの同期が可能です。



同期設定ダイアログ

ブレークポイント

ソフトウェアブレークポイントは、各プロセッサーの設定数の合計が最大63点まで使用できます。複数のデバッガから、同じ物理アドレスにソフトウェアブレークポイントを設定した場合には、プロセッサー間でのブレークポイントの競合をPARTNER-Jet3が自動的に解決します。
また、1つのプロセッサー(デバッガ)に対して指定したソフトウェアブレークポイントと同じアドレスの命令を、他のプロセッサーが実行した場合にはデバッガを停止させないといった排他処理も、PARTNERデバッガが高速に制御しています。
ハードウェアブレークポイントはプロセッサーごとにブレーク条件を設定できるだけでなく、SMP構成の場合には、複数のプロセッサーに同時に同じ条件でブレークポイントの設定が可能です。

リアルタイムトレース機能

マルチコアシステムでは、Cortex-A53とCortex-R7のように異なる種類のプロセッサーであっても、同時にトレース情報を取得し、時間軸に同期した参照ができます。
しかし、同時に複数のプロセッサーのトレース情報を取得すると、トレース情報が増えるため転送されるトレースパケットがオーバーフローする可能性があります。このような場合には、特定のプロセッサだけを指定してトレース有効とすることで、トレースパケットのオーバーフローの抑制が可能です。

QProbe

PARTNER-Jet3 を使用して取得したトレースデータを利用して、ソフトウェア全体を鳥瞰しながら性能解析や品質改善を支援する動的解析ツールです。

＊QProbe は PARTNER-Jet3/ デバッガとは別売の、専用解析ソフトウェア製品です。

大規模化・複雑化するソフトウェアは、論理バグが取れてからが評価作業の本番がスタート

組込みシステムの高機能化に伴い、ソフトウェアはますます大規模化や複雑化しています。外部から導入したOSやミドルウェアを利用して多拠点で開発したソフトウェアは、結合テストで不具合を取り除いた後にソフトウェア全体を鳥瞰しながら性能のボトルネックの解析や品質評価作業の本番がスタートします。

動的解析ツールであるQProbeでは、デバッグツールであるPARTNER-Jet3で取得したトレース情報をそのまま性能解析に利用しており特別なハードウェアを用意する必要がありません。同じ環境でデバッグ作業と評価作業ができるので、評価環境構築の準備に要するエンジニアの負担を大きく減らすことができます。

統計的手法を用いた解析と、時系列で変化する情報を解析

関数の実行回数やカバレッジ、関数の呼び出し参照の関連性表示といった統計的な解析結果と、関数実行履歴や実行タイミングのばらつきといった時系列で変化する情報を組み合わせて分析することで、性能のボトルネックとなっているソフトウェアに潜んだ問題を解決に導きます。

統 計 & 時系列

大容量トレースをフル活用

トレース情報はPARTNER-Jet3本体内のメモリに保存できるだけでなく、PARTNER-Jet3上のUSB 3.0を介してPCや外部機器への保存が可能です。外部機器で保存した大容量（長時間）のトレース情報もQProbeを使えば統計的な分析データとしてすぐに活用できます。
取得したトレース情報を保存して、後にデバッガやターゲット実機を接続することなくQProbeで再度分析できるので、プログラムの改訂前後の分析結果の比較も容易です。

2種類のトレース情報を利用し、関数やコンテキスト単位でソフトウェアのふるまいを可視化

命令レベルトレース

ETMトレースといった、プロセッサから直接出力される実行履歴を使用します。取得した情報を元にソースコードと照らし合わせながら関数の実行履歴を分析します。情報取得のオーバーヘッドが無く、ツールチェーン依存性もありません。

関数レベルトレース（コンテキスト集計）

ソースコードの関数の入口／出口にトレース用のフックデータを出力するルーチンを埋め込み、実行時に出力されるデータをトレース情報として取得し分析します。タスクやスレッド単位で実行履歴の分析ができ、機能単位での問題絞り込みに有効です。埋め込みトレースでは、QProbeがフック用のコードを埋め込むため、GCCで作成されたソースコード向けの機能となります。

プロファイル表示

関数やコンテキストごとの実行時間、実行ステップ数などの情報をグラフィカルに表示することにより、実行スピードに対するボトルネックとなっている関数の特定を容易にします。

時系列的分析の例

関数やコンテキスト単位に実行状況を時系列で表示することにより、注目している関数が、実際にどのように実行されているかを見ることができます。

呼び出し参照表示

全実行履歴より、指定された関数がどの関数から呼び出されているか、またどの関数を呼び出しているかの制御関係を、呼び出し回数を含めて表示します。制御関係を把握することで、そのモジュールの影響範囲を把握することや、テスト範囲の絞り込みを行うことが可能です。

統計機能

関数やコンテキストについて、連続して実行した単位毎での、実行時間のばらつきや、また終了から実行までの間隔のばらつきなどを表示します。周期的に実行されなければならないタスクについて、周期が外れているかどうか、またどの程度はずれているかなど確認ができます。

その他のQProbeの仕様

- ソースカバレッジ表示が可能です。
- QProbeはマルチコアシステムに対応しています。
- コンテキスト解析が可能なOSは、LinuxとITRON系リアルタイムOSです。
- ETMトレースが取得できない（端子が無い）マイコン/SoCでも、埋め込みトレースによるコンテキスト解析機能が利用可能です。
- QProbeで分析した結果をレポートとして利用できるよう、全ての解析結果はcsv形式でエクスポート可能です。

Linuxデバッグ

Linuxを隅々までデバッグする

64bit Linux、LPAE Linux、thumb2ビルドカーネルに対応するとともに、“デバッグパッチを適用したカーネル”、“デバッグパッチ無しのカーネル”のいずれにおいても、アプリケーション／ローダブル・モジュールのデバッグが可能です。

■デバッガから見たLinuxのプログラムの種別と特徴

プログラム種別	空間	アドレス	ページング	デバッグ
ブートローダ	非MMU空間	固定番地	なし	既存の組み込みデバッグに同じ
カーネル	MMU上の単一カーネル空間	固定番地	(ほぼ)なし	既存の組み込みデバッグに(ほぼ)同じ
ローダブル・モジュール	MMU上の単一カーネル空間	リロケータブル	デマンドページング	リロケーション解決、ページングへの対応が必要
アプリケーション	MMU上の論理多重仮想空間	固定番地	デマンドページング	論理多重空間、ページングへの対応が必要
共有ライブラリ	MMU上の論理多重仮想空間	リロケータブル	デマンドページング	リロケーション解決、論理多重空間、ページングへの対応が必要

ブートローダ

一部のブートローダは、実行を開始してから外部メモリ等に移動して動作します。PARTNERデバッガでは、シンボルを読み込む際にオフセットを指定することで、1つのオブジェクトを移動前と移動後の両方のアドレスに対応させてソースデバッグができます。

カーネルデバッグ

カーネル空間は単一の固定アドレスで動作するので、通常は特別な設定をしなくてもデバッグが可能です。カーネルがSMP構成の場合には、複数のプロセッサーに対して個別にデバッガを接続し同時にその状態を見ることができるため、カーネルのブート時のデバッグが容易に行えます。
また4.x以降のカーネルでは、“idle時にコアの電源を落とす”、“デバッグ中に一定時間ブレーク状態が続くと、警告のメッセージをコンソールに出力する”というように、デバッグをしづらくする機能が多数追加されています。このような問題に対処するため、カーネルのコンフィグレーションの変更やコマンドラインの追加で問題を回避・抑制する方法を具体的にマニュアルでガイドしています。

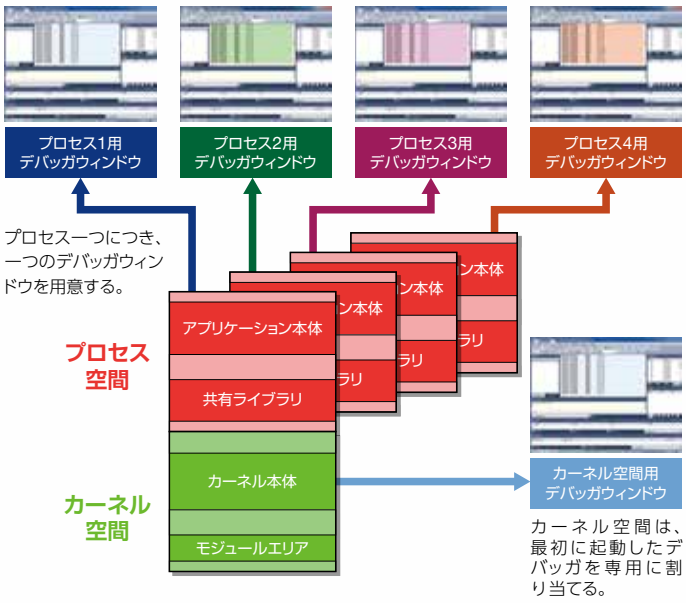
ローダブル・モジュール

Linuxのローダブル・モジュールは、アドレス情報を持たないリロケータブルなオブジェクトで、insmodされた時にアドレスが確定します。PARTNERデバッガは、insmodされた時にカーネル内のモジュールのメモリマップを調べてシンボル情報を読み込むことで、ローダブル・モジュールのソースデバッグを実現しています。
ユーザは、デバッグしたいローダブル・モジュールを登録してinsmodするだけで、モジュールの先頭からデバッグできます。

アプリケーション

Linuxのアプリケーションは多重仮想空間になっており、全てのアプリケーションは、同じアドレス範囲で動作します。そのため、単一の空間で動作するプログラムのようにアドレス情報だけでアプリケーションを識別できません。PARTNER デバッガは、Linuxアプリケーション個別の空間を認識することでアプリケーションのデバッグを実現していています。
複数のアプリケーションをデバッグする際は、デバッガウィンドウを複数起動し、それぞれのアプリの専用デバッガとして対象のアプリのシンボルをロードしてソースコードデバッグができます。
main()からのデバッグ、実行中のプロセスのデバッグにも対応できます。
またマルチプロセス、マルチスレッドのデバッグにも対応しています。

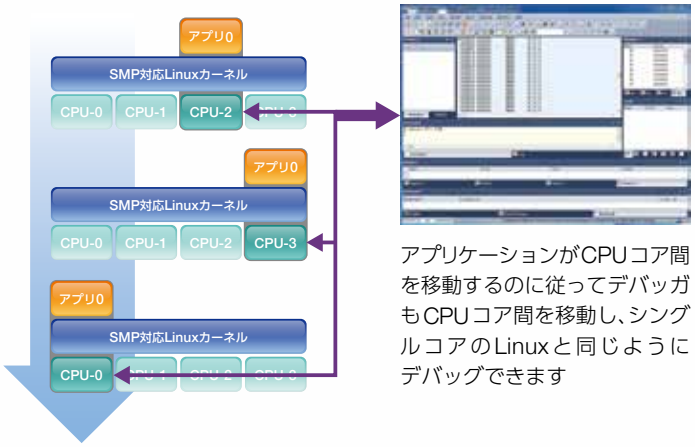
■デバッガと空間



共有ライブラリ

デバッグ中のアプリケーションにリンクされている共有ライブラリをデバッグする場合は、リンクされている共有ライブラリ名やマップ情報をPARTNERデバッガが自動認識してシンボル情報を読み込むので、ソースコードデバッグが可能です。

SMP Linuxのデバッグ



アプリケーションがCPUコア間を移動するのに従ってデバッガもCPUコア間を移動し、シングルコアのLinuxと同じようにデバッグできます

デバッグ対象は、ブートローダ／カーネル／ローダブル・モジュールを含む Linux の全てのプログラムです。カーネル／ローダブル・モジュール／アプリケーションの同時デバッグが可能

＊Linux のローダブル・モジュール、アプリケーションのデバッグには、Linux オプションデバッガが必要です。

Linuxデバッグ機能別対応リスト

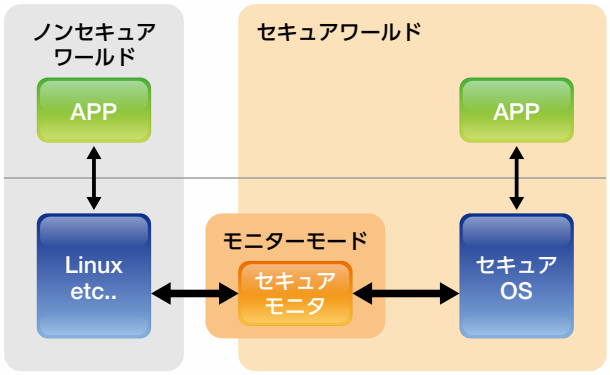
デバッグ対象	カーネル
	ローダブル・モジュール
	アプリケーション
	特別なアプリ（init=プロセスID1番）
	マルチプロセス
	マルチスレッド
デバッグ機能（カーネル空間）	ソフトウェアブレーク
	ハードウェアブレーク（命令実行）
	ハードウェアブレーク（データアクセス）
	実行トレース（分岐トレース）
	ソースレベルデバッグ
	ローダブル・モジュールの自動リロケーション
	デマンドページングの解決
	カーネルブレーク中のプロセスデバッグ
デバッグ機能（ユーザ空間）	ソフトウェアブレーク
	ハードウェアブレーク（命令実行）
	ハードウェアブレーク（データアクセス）
	実行トレース（分岐トレース）
	ソースレベルデバッグ
	デマンドページングの解決
	共有ライブラリの自動リロケーション
	実行中プロセスへのアタッチ
	プロセスブレーク中のカーネルデバッグ
	プロセスブレーク中のカーネルや他のプロセスの実行 ※

※ 仮想ICE機能で実現します

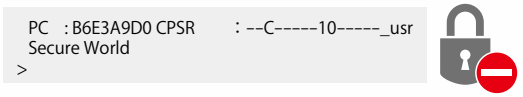
TrustZoneのデバッグ

Arm向け応用機能

トラストゾーンは、主にCortex-Aのプロセッサ向けの拡張機能で、アクセス保護が必要なデータやソフトウェアが動作する「セキュアワールド」と、それ以外の「ノンセキュアワールド」にメモリ空間を分離することができます。
ノンセキュアワールドとセキュアワールド間の切り替えは、モニターモード（mon）で動作するセキュアモニタを通して行われます。
PARTNERでは、これらの状態を、ステータスバーや実行停止時のコマンドウィンドウの表示から知ることができます。また、ハードウェアブレークやベクタキャッチのオプションで、有効となる状態を指定すれば、ノンセキュアワールドでのみ停止するようなブレークポイントや、セキュアモニタに切り替わった直後に停止するベクタキャッチなどの作成が可能です。
ただし、ハードウェアでセキュアワールドのデバッグが禁止されている場合は、セキュアワールドやセキュアモニタの実行中に停止させることはできません。セキュアワールドでのプログラム実行中にブレーク要求が発生した場合はそのブレーク要求が保留され、セキュアワールドからノンセキュアワールドに分岐した瞬間にブレークします。



トラストゾーンの使用例



セキュアワールドで停止した際の表示

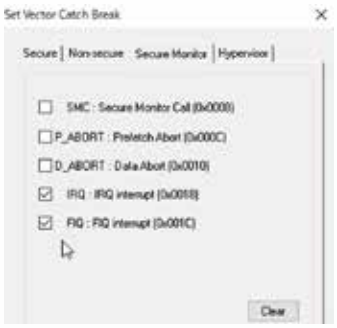
Linuxデバッグをより快適に軽快にする独自機能（PARTNER-Jet3標準機能）

teraterm連携機能

teraterm連携機能は、u-bootやLinuxのコンソール（teraterm）入出力を、PARTNER デバッガのコマンドウィンドウ経由で入出力する機能です。
デバッグ開始時にPARTNERデバッガのマクロで、カーネルシンボルロードや、ブレークポイント設定を自動化することに加えて、teraterm連携機能を使うことにより、Linuxへのログインやアプリケーション実行までも、Linuxのコンソールを操作することなくマクロを使った自動化が可能です。 ※ホストPCにteratermがインストールされている必要があります。

デバッグ情報キャッシュ機能

組込みLinuxの64bit対応により、vmlinuxのシンボル情報が更に巨大化し、シンボル情報の解析に時間がかかるようになりました。
PARTNERデバッガはこのストレスを軽減するため、カーネルのシンボル情報をホストPCにキャッシュする機能を実装しています。10秒以上かかっていたシンボル情報解析が1秒以下になりました。
※デバッグ情報キャッシュ機能はLinux カーネル専用ではありません。



セキュア空間ごとのベクタキャッチ設定

セキュア空間を指定したハードウェアブレーク

■PARTNER-Jet3機能仕様一覧

		Model10	Model15
JTAGクロック		70MHz	
トレースソース	ETB/ITM/ETR(トレース端子不要)	○	○
	TPIU(トレース端子使用 4/8bit *1)	–	○ (200MHz DDR)
トレースメモリ	PARTNER-Jet3内蔵トレースメモリ	–	512Mバイト
ホストPC	接続インタフェース	USB3.2Gen1(推奨)/USB2.0	
	OS	Windows 10/11以降	
電源仕様	USBバスパワー	○	○
	ターゲット信号電圧(JTAG信号およびトレース信号)	1.6V ～ 3.6V	1.6V ～ 3.6V
ACアダプタ		IN：AC100-240V OUT：DC5V	
本体外形サイズ(突起部を除く)		105*115*33 mm	
動作温度		5 ～ 35℃　湿度85%以下(結露なきこと)	

*1：Mictor-38 pinプローブ(8/4bit)，Halfpitch-10/20 pinプローブ(4 bit)を接続して使用

■製品構成

PARTNER Jet3 外観（Model10/Model15 共通）

■品名一覧(品名の一部です)

品名	機能	仕様	型式
JTAGエミュレータ PARTNER-Jet3	ターゲットボードと接続するJTAGエミュレータボックスです	JTAGデバッグ機能を搭載	Model10
		トレース機能を強化	Model15
JTAGプローブ	PARTNER-Jet3と ターゲットボードを接続するためのプローブです	Standard-20pinプローブ	STD20P-01
		HalfPitch-10/20pinプローブ	HP10/20P-01
		Mictor-38pinプローブ	MICTOR38P-01
PARTNER-Jet3デバッガ	Armプロセッサ用のデバッガです	32bit、シングルコア対応	PDARM3-S
		32bit/64bit対応、シングル/マルチコア対応	PDARM3
	RISC-Vプロセッサ用のデバッガです	32bit/64bit対応、シングル/マルチコア対応	PDRV3
PARTNER Linuxデバッガ	PARTNERデバッガ上で、Linuxアプリケーションを デバッグ有効にするためのオプションライセンスです	Arm 32bit/64bit 用共通 Linux デバッガ	PDLINUX3-XX

保守サービスについて

PARTNER-Jet3のご利用にあたっては、ご購入の際に保守サービスへの加入を推奨しております。保守サービスに加入いただきますと、以下のようなサービスを受けられます。

- **最新デバッガソフトウェアの入手**
  バージョンアップ、新規対応デバイスの追加 などを含みます。
- **テクニカルサポート(問い合わせ)**
  電子メールまたはFAXでご対応いたします。
  出張サポートや、ユーザー様環境を使ってのサポートは別途有償です。
- **ユーザーサポート専用回線のご利用**
- **ハードウェアの修理及び検査の基本費用(税別)の免除**

コンパイラやPC環境(Windows OS)がアップデートした場合、デバッガソフトウェアが正しく利用できないといった支障が発生する可能性がありますので、保守サービスにご加入のうえで常に最新のデバッガソフトウェアでPARTNER-Jet3をご利用ください。

なお、デバッガソフトウェアはご購入後3ヶ月間の無償サポート期間を、ハードウェアは1年間の保証期間を設けております。

詳しくは、https://jet.kmckk.com/partner-jet3-toppage/jet3-anualsupport/をご覧ください。

KMC Member of eSOL
京都マイクロコンピュータ株式会社
本　　社：〒610-1104 京都市西京区大枝中山町2-44　Tel.075-335-1050
東京オフィス：〒170-0013 東京都豊島区東池袋 2-5-9 加瀬ビル122 4F Tel.03-5957-1001
お問い合わせメールアドレス：jp-info@kmckk.co.jp
https://jet.kmckk.com/partner-jet3/

取扱店

2026.03