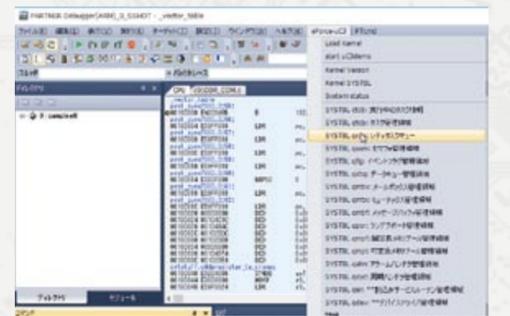


PARTNER-Jet2はLinuxやRTOSのデバッグに対応しており、以下のようなOSデバッグ支援機能があります。

Linux対応、RTOSの資源表示対応

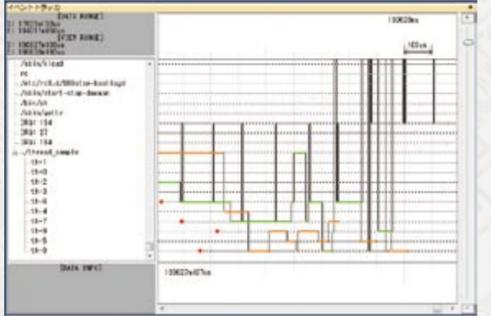
PARTNER-Jet2は、PARTNER-Jetでも好評だったLinuxデバッグ機能を踏襲しています。共有ライブラリを含むユーザー空間のデバッグから、カーネルモジュールまで、JTAGデバッガでCPUをブレークしながらソースコードデバッグする事が可能です。また、RTOSについては、代表的なRTOSに対応した資源表示スクリプトを用意しています。これを使えば、タスクやセマフォの状態を表示する事ができ、RTOSを使ったデバッグをしやすくします。



※対応している内容やOSについては、お問い合わせください。

イベントトラッカーによるタスクやスレッドの遷移表示

LinuxもRTOSもイベントトラッカーによるスレッドやタスクの遷移状況の表示が可能で、これを使えば動作の状況を把握しやすくなります。また、イベントトラッカーを使って、スレッドやタスクを実行時間でプロファイリングする事もできます。



PARTNER-Jet2が提供する各種トレース機能は動的解析ツールQProbeと連携することができます。

PARTNER-Jet2のModel20/Model30は、大容量トレースメモリや、ソースレベルトレース、命令レベルトレース、拡張プローブ(Model30のみ)対応、ストレージへの大容量保存など、多彩なトレース支援機能があります。これにより、ソフトウェアの動作を俯瞰的に動的解析するツールに活用する事ができます。

例えば、以下のように長時間のソースレベルトレースを取得する事ができます。

- 関数の入出力にトレースポイントを埋め込む
- トレースポイントによるソースレベルトレースデータを非同期バスコネクタから取り込む
- 取り込んだトレースデータを、USB3.0経由でPC上のSSDに連続保存

取り込んだ長時間のトレースデータについては、動的解析ツールQProbeを使って、実行遷移や実行時間の分散状況、またプロファイルなど各種の解析が可能になります。

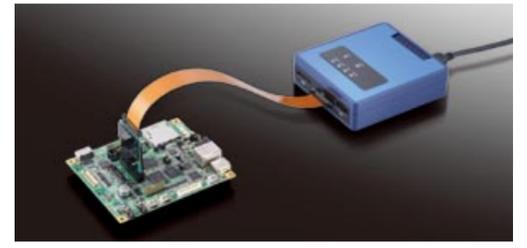


| | |
|-------------------------|--|
| 長時間のトレースデータ保存に対応 | <input checked="" type="checkbox"/> USB3.0を使ってPCのストレージに連続保存 (Model20, Model30) |
| | <input checked="" type="checkbox"/> 本体に直接接続したSSDに連続保存 (Model30) |

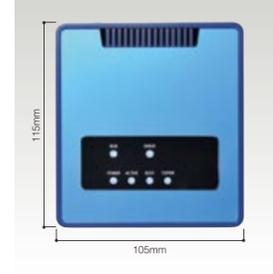
PARTNER-Jet2では、長時間の動的解析を実現するべく、トレースデータを本体に接続したSSDに保存したり、USB3.0経由でPC上に保存する事ができます。ストレージに空き容量がある限り保存できるので、トレース保存できる時間が飛躍的に長くなります。

※機能の組み合わせや利用条件など、詳細については別途資料をご確認ください。

PARTNER Jet2 Model10



Model10/20/30



Model10



Model20



Model30



| 機能仕様 | | Model10 | Model20 | Model30 | |
|---------------------|----------------|---------------------|---------|----------------|----------|
| USB/バスパワー | | ○ | × | × | |
| JTAGクロック上限 | | 70MHz | 100MHz | 100MHz | |
| トレース種別 | CPUトレース | ETB | ○ | ○ | |
| | | ETM/PTM/AUD | × | ○ | |
| | トレースポイント | ITM | ○ | ○ | |
| | | メモリ | × | ○ | |
| トレース保存仕様 | トレースメモリ容量 | なし | 4G/バイト | 8G/バイト | |
| | | トレースクロック | なし | 500MHz以上 | 500MHz以上 |
| | テラトレース | LVDS (ETM HSSTP) 対応 | × | × | ○ |
| | | 外部バス/SD | × | × | ○ |
| 右の機能は、全てのモデルで利用可能です | USB3.0からPC | × | ○ | ○ | |
| | Jet2からSSD | × | × | ○ | |
| ブレイクポイント | ソフトウェアブレイクポイント | 最大64点 | | | |
| | ハードウェアブレイクポイント | CPU内蔵H/Wブレイク機能に対応 | | | |
| スタンダオン実行 | | × | × | ○ | |
| 物理仕様 | | Model10 | Model20 | Model30 | |
| ACアダプタ定格 | IN | AC100~240V,0.3A | | | |
| | OUT | DC5V,2A | | | |
| サイズ | | 105 x 115 x 33 | | 105 x 115 x 56 | |
| 使用環境 | | 温度5~35、湿度85%以下 | | | |

▶保守サービスについて

PARTNER-Jet2は保守サービスに加入されていなくても、利用する事は可能です。しかし、保守契約終了後に、弊社が新しく対応したプロセッサやコンパイラなどを使って利用する場合、正しく利用できないなどの支障が発生する可能性があります。機能対応について、常に最新を維持できるよう、保守サービス契約の継続加入を推奨します。保守サービス契約をしていただく事により、右の事が可能になります。

<http://www.kmckk.co.jp/jet2support.html>

- 最新デバッグソフトウェアの入手
- テクニカルサポート (電子メール or FAX)
- ハードウェア修理の基本費用免除
- 来社、または出張によるサポート (別途追加費用必要)
- ユーザー様環境を提供いただいた際のデバッグ動作に関するサポート (別途追加費用必要)



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: 〒105-0004 東京都港区新橋2-14-4 Rビル5F Tel.03-5157-4530

<http://www.kmckk.co.jp/jet2/>
 お問い合わせメールアドレス: jp-info@kmckk.co.jp

ARM, SH, Intel 対応 JTAGエミュレータ

PARTNER Jet2



Model30
登場

USB3.0 対応 64bit 対応

デバッグ、そして解析を極める

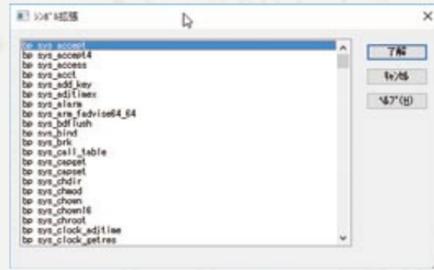
PARTNER Jet 2 デバッガの基本機能

PARTNERは、デバッガとして30年間進化し続けたプロ仕様ソースコードデバッガです。プログラマがソースコードデバッグを行う時に必要な細かなデバッグ支援機能を豊富に備え、GUIとCUIを統合することで「使い込みやすい」ユーザーインターフェイスに仕上がっています。そして、JTAGデバッガとしての基本機能も、前モデルのPARTNER-Jetを引き継ぎ、さらに進化して実装されています。

基本機能 ソースコードデバッガ

一般的なソースコードレベルデバッガが備える機能(インスペクト表示やソース行クリックでのブレークポイントなど)の他、プログラマが便利に使えるよう、以下の機能などを備えています。

- 複数のデバッグ情報ファイルの読み込み機能
- シンボル名補完機能



"BP sys_" と入力して、シンボル補完した様子

- プリプロセッサマクロのインスペクト機能 (対応コンパイラのみ)
- 各ウィンドウやコマンドの結果を、ファンクションでテキスト保存
- ステップ実行履歴機能
- ソース上にブックマーク指定
- 関数呼び出し履歴表示とローカル変数表示の連動

基本機能 ソフトウェアブレークとハードウェアブレーク

ソフトウェアブレークポイントは、実行を停止させるだけでなく、多彩な処理が付加的に行えます。実行時にマクロを含む特定のコマンドを実行したり、printライクなメッセージ出力ができます。ハードウェアブレークポイントは、CPUのオンチップデバッグ機能を使って実現しています。そのため、CPUによってできる事が異なりますが、例えばメモリアccessを検出してブレークさせる事ができます。

基本機能 実行時間簡易測定機能

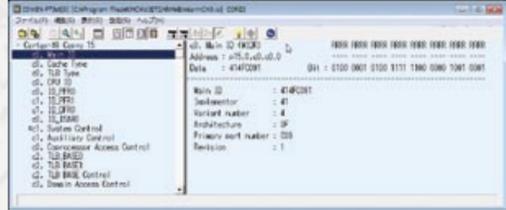
プログラムがブレークポイントで停止したとき、実行開始から停止までの時間について自動的に測定し、表示しています。

基本機能 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フラッシュ上に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で、標準サポートの並列接続のNORフラッシュと同様に書き込みを行う事ができます。

基本機能 I/Oレジスタ機能

周辺系I/Oのレジスタについて、レジスタの名前やビットアサイン、意味などを分かりやすいGUIで表示する機能があります。特定のビット単位での変更も可能です。



基本機能 USBバスパワー動作対応

Model 10

Model10については、USB2.0/3.0でのバスパワー動作を実現しました。

応用的なデバッグ機能

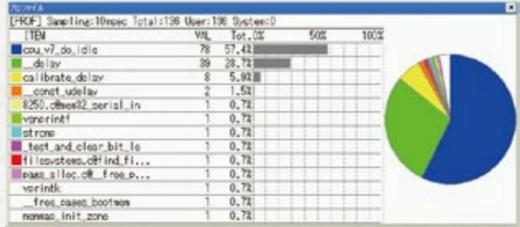
PARTNER-Jetで培ったマルチコアデバッグをはじめとする高機能 JTAGデバッグの機能を、PARTNER-Jet2ではさらに進化させました。これにより、複雑化する近年の組み込み機器開発で求められる様々なデバッグシーンに対応できます。

応用機能 マルチコアデバッグ

マルチコアデバッグは、SMP/AMP どちらにも対応しています。AMPでは、Cortex-A5とM3など、異なる種類のプロセッサコアが混在していてもデバッグが可能です。また、CoreSight CTIなどのコア間同期に対応しているので、クロックレベルでの同期実行・ブレークをすることもできます。

応用機能 プロファイル機能

プログラムカウンタをサンプリングする事で、C/C++関数単位でのプロファイリングが可能です。ソースコードの改変などは不要で、デバッグ情報さえあれば簡単にプロファイリング可能です。結果はテキストだけでなく、円グラフや棒グラフでの表示が可能です。



応用機能 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、Model20以上ではリアルタイムトレースと連携したり、ソースレベルトレースと連携したりすることができます。

応用機能 電源監視機能

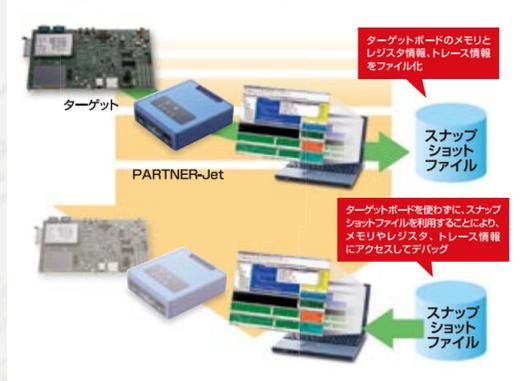
JTAGデバッグ中にターゲットボードの電源が落ちても、それを認識して、電源再投入時に自動的にデバッガを再接続する事ができます。再接続時には、設定しておいたハードウェアブレークなどを復元するので、電源断から再投入時のデバッグなどに応用する事ができます。

応用機能 マクロ機能

デバッガのコマンドライン操作を、パッチ的に自動化することができるマクロ機能を搭載しています。マクロ機能では、コマンドだけでなく、シンボルを扱ったり値を評価したり、またそれらを簡単な演算式や制御構文で処理をすることができます。

応用機能 スナップショットデバッガ

ある瞬間のターゲットシステムの状態である、CPU(汎用レジスタや制御レジスタ)と物理メモリをPARTNER-Jet2がファイルに保存し、あとからターゲットハードウェアがなくてもデバッガ上にCPUの状態やメモリを復元して静的にデバッグする機能です。障害発生時などの情報を保存しておき、あとから複数人で解析したりするなど、様々な応用が可能です。リアルタイムトレースやソースレベルトレースの情報も保存する事が可能で、これらも一緒に保存しておけば、障害にいたる動作もある程度確認する事ができます。



応用機能 デバッグスクリプト

マクロ機能より強力な制御構文を有するスクリプト機能を搭載しています。オープンソースのSquirrel言語という、C/C++ライクなスクリプト言語なので、C/C++プログラマの方が利用しやすくなっています。デバッグ情報にアクセスするAPIを使って、シンボルのアドレスを取得したり構造体のメンバにアクセスしたりする事ができます。またメモリやCPUにアクセスするAPIも存在します。forやifの制御構文もあるので、リストされた構造体を追跡したりする事もできます。これらを使えば、RTOSの内部のタスクなど資源をリスト表示したりする事も可能です。

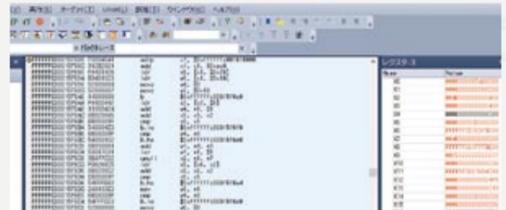
```
32
33
34 local task_size = SearchSymbolNameSize("init_task");
35 local task_top = GetSymbolAddress("init_task");
36 local offs_tasks = OffsetOf("task_struct", "tasks");
37 local offs_com = OffsetOf("task_struct", "comm");
38 local size_com = SearchSymbolNameSize("init_task.com");
39 local offs_pid = OffsetOf("task_struct", "pid");
40 local offs_thread_group = OffsetOf("task_struct", "thread_group");
41 local offs_w = OffsetOf("task_struct", "w");
42
43 local task = task_top;
44
45 do {
46     local pid = Read32(task + offs_pid);
47     local com = ReadMemoryAsString(task + offs_com, size_com);
48     local w = Read32(task + offs_w);
49     local thread_top = task + offs_thread_group;
50     local thread = Read32(thread_top);
51     print(forcat(" %05 %s [%s] [%s] %s", pid, com, w == 0 ? "kni" : "usr", get
52         while (thread_top != thread) {
53             local thread_task = thread - offs_thread_group;
54             pid = Read32(thread_task + offs_pid);
55             com = ReadMemoryAsString(thread_task + offs_com, size_com);
56             print(forcat(" %05 %s [%s] %s", pid, com, getTaskState(thread_task
57                 thread = Read32(thread_task + offs_thread_group);
58         }
59     }
60     task = Read32(task + offs_tasks) - offs_tasks;
61     while (task_top != task);
62 }
```

ARMプロセッサ向け応用機能

近年の組み込みシステムで標準的に利用されているARMプロセッサについて、PARTNER-Jet2はARMプロセッサの様々なデバッグ機能に対応します。

ARM向け応用機能 64bit ARMv8対応

PARTNER-Jet2は最新のARMv8アーキテクチャ、64bitプロセッサに対応します(Cortex-A57/A53など)。64bitモードのAArch64モード、32bitモードのAArch32モードの両方に対応し、また、各モードの混在実行にも対応しています。



ARM向け応用機能 プローブホットプラグ対応

ターゲット動作中にプローブを取り外したり接続したりする事ができる、プローブのホットプラグに対応しています。ターゲットシステムの異常を認識したその時に、プローブを接続することで、異常と思われる状態のプロセッサやメモリ、プログラムの状態を確認する事ができます。ソースレベルトレースのメモリ保存機能と併用すると、デバッガ接続前の実行履歴もわかります。

ARM向け応用機能 ベクタキャッチブレーク対応

プロセッサで発生するIRQやData Abortなど、例外発生をキャッチして発生するハードウェアブレークに対応します。これらの例外は、セキュアモードや非セキュアモードなど、各種実行モード別の例外ベクタ毎に設定可能です。

>> 他ツール連携

ツール連携 exeGCC & VLINK対応

KMCが提供するGCCコンパイラexeGCCでビルドしたプログラムをPARTNER-Jet2からダウンロードして利用すると、専用ライブラリを介してWindows上のファイル資源にアクセスしたりする事が可能です。VLINKの詳細については、exeGCCのカタログをご確認ください。

ツール連携 Eclipseからの利用

デバッガフロントエンドにEclipse/CDTを利用可能です。エディット、ビルド、CDTのデバッガでPARTNER-Jet2を使ったデバッグができます。ARM社のEclipse for DS-5、ルネサス社のe2studio、Xilinx社の、Xilinx SDKなど、各社から提供されるEclipseベースの統合開発環境にも、プラグイン追加で対応できます。

ARM向け応用機能 AXI-AP/AHB-AP機能

通常のJTAGデバッガでのメモリアccessは、CPUのロード・ストア命令を使ってメモリを読み書きします。そのため、メモリアccess中はCPUが(一時的に)ブレークした状態で行います。このAXI-AP(もしくはAHB-AP)機能を使うと、CPUを介さずにAXIバスなどにアクセスできるため、プログラム実行中にCPUの動作を妨げないでメモリなどの読み書きができます。たとえば、CPUが不正なメモリアccessによりCPUバスがロックしてしまっている時でも、メモリを読み出す事ができる場合があり、問題調査に役立てる事ができます。また、このアクセスはCPUのキャッシュを介さないため、キャッシュ回りのデバッグにも役立ちます。

ARM向け応用機能 ITMトレース

CoreSightのITMに対応しており、ターゲットシステムから送られるログメッセージをデバッガに表示する事ができます。SWOにも対応し、SWDデバッグ接続環境下では、SWCLK、SWDIO、SWOの三線信号でデバッグ+ITMトレースの利用が可能です。

ARM向け応用機能 MMU対応

ARMプロセッサのMMUに対応し、MMUの状態表示やON/OFF機能、また論理アドレスから物理アドレスへの変換機能を有しています。

ARM向け応用機能 TrustZone,HyperViser対応

TrustZoneのセキュアモニタモード、またCortex-A15などがサポートしているHyperViserでのモニタモードなど、特殊な動作モードのデバッグも可能で、セキュアモニタや仮想化モニタの開発でのデバッグを可能にします。

ツール連携 VisualStudioからの利用

デバッガフロントエンドにVisualStudioを利用可能です。exeGCCコンパイラでビルドし、VisualStudioのデバッガがU/IでPARTNER-Jet2を使ってデバッグができます。VisualStudioを使って、コンパイラ、デバッガを含めた組み込みシステム向けのオールインワンソリューションを実現できます。

(※特定プラットフォーム向け専用機能です。VisualStudioに拡張機能を組み込むため、Visual Studio Professional Edition 以上での対応となります。)