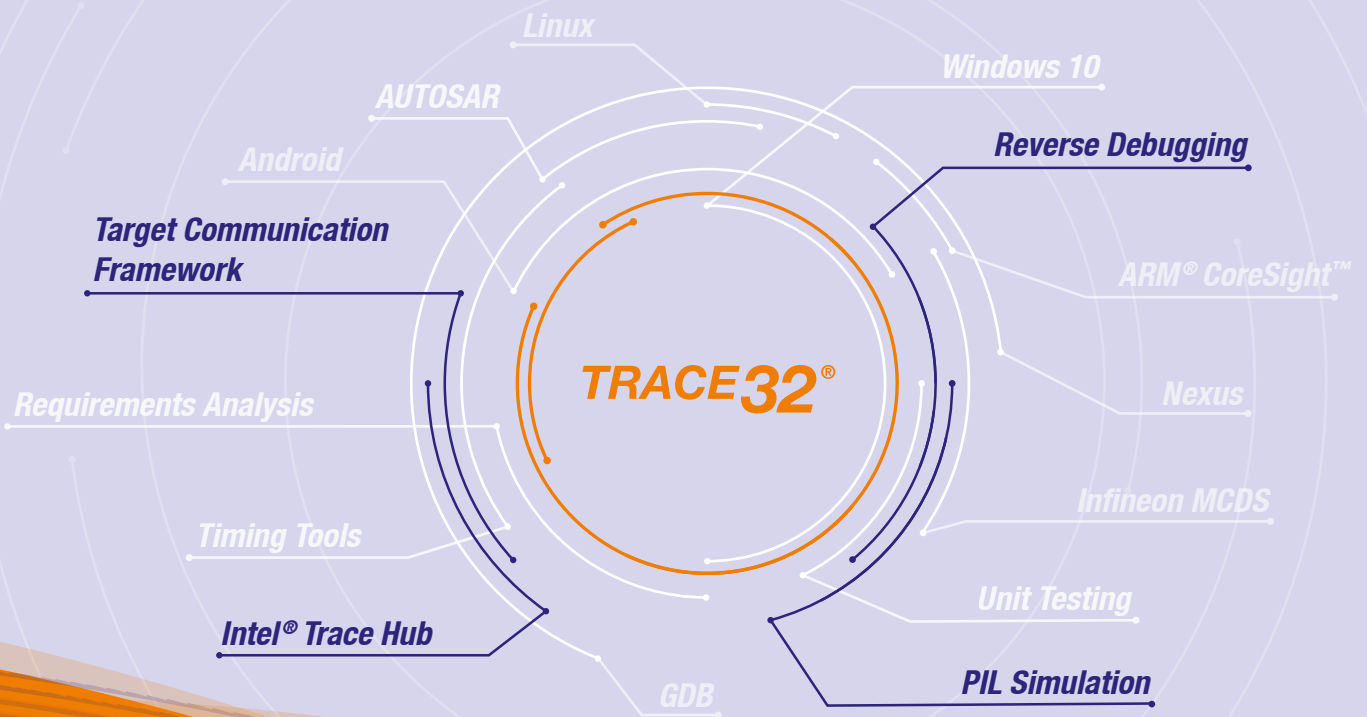


# NEWS 2016



## コンテンツ

TRACE32を使用したPILシミュレーション	2
新型TRACE32ベースモジュール	4
TRACE32 PowerTrace Serial	5
Intel® Trace Hubサポート	6
TRACE32 TCFエージェント Wind River Workbench Eclipse	7
TRACE32 UndoDBフロントエンド	8

# TRACE32を使用したPILシミュレーション

2016年3月にローターバッハは、PIL（プロセッサインザループ）シミュレーション用 Simulink プラグインの提供を開始しました。この新しいプラグインの利用により、モデリング環境は TRACE32 デバッガを経由して直接ターゲットと通信できるようになります。

ここ数年の間、ソフトウェア開発におけるモデルベースメソッドの重要性が高まっています。モデルベースメソッドのメリットは、ソフトウェア設計の継続的検証が可能なこと。プロセッサインザループシミュレーションは、設計検証における重要なステップと位置付けられています。

## PIL シミュレーション

PIL シミュレーションは、開発したアルゴリズムがターゲット環境において正しい機能を提供していることを確認するために行われます。この検証ステップは以下のターゲットシステムの何れか1つで実施されます。

- 最終ハードウェア / 評価ボード
- 仮想ターゲット / コアシミュレータ
- TRACE32 インストラクションセットシミュレータ

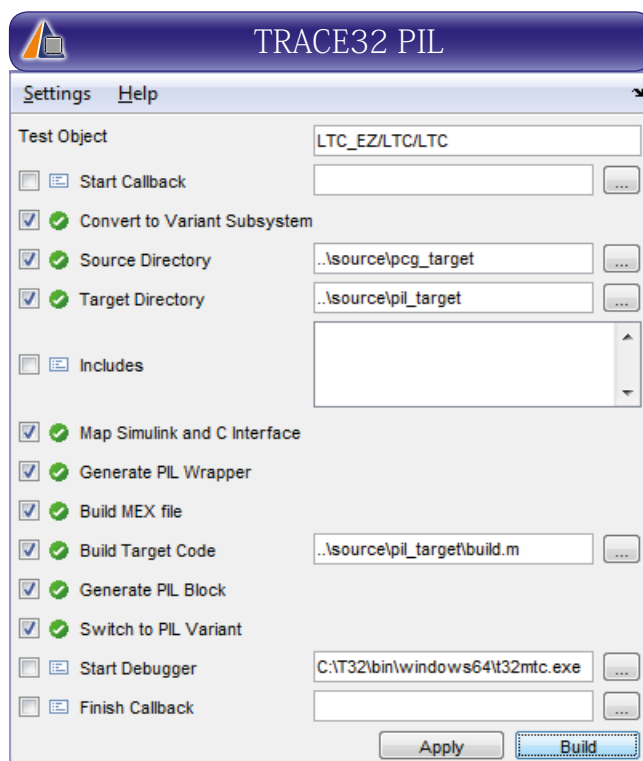
PIL シミュレーションを実行するために、テスト対象のアルゴリズムはモデリング環境で PIL ブロックに置き換えられます。

## TRACE32 PIL

TRACE32 PIL プラグイン（右の画像参照）は PIL シミュレーションの設定に使用します。このプラグインの重要な設定項目を次に概説します。

### Checkbox: Map Simulink and C Interface

PIL ブロックを生成する前に、Simulink とターゲットアプリケーション間のインターフェースを設定する必要があります。S 関数（ユーザ定義ブロック）のコールバックを該当する C 関数にマッピングする設定が、マッピング



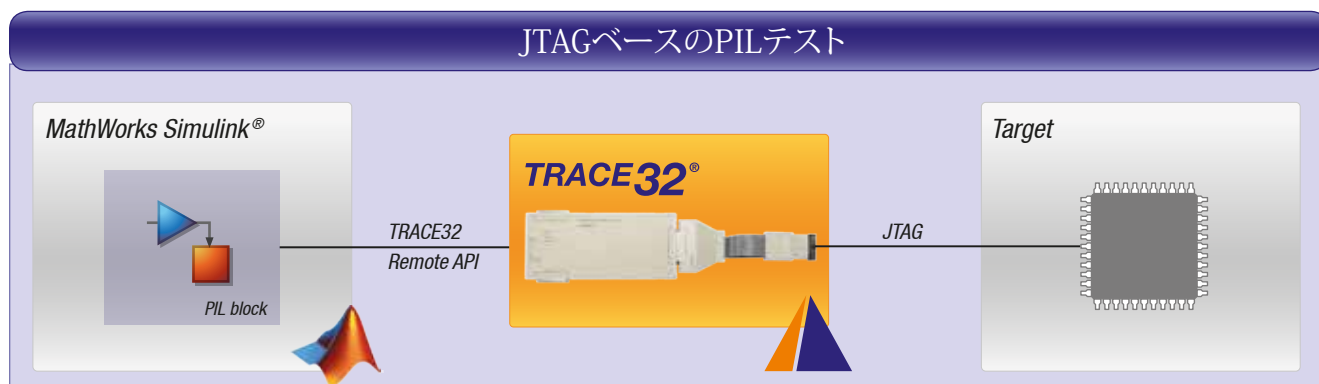
GUI ダイアログ（p.3 画像参照）に、自動的に用意されます。ダイアログで確認の上、修正が可能です。モデルパラメータと C 変数間でも同様にマッピングを実施しなければなりません。

### Checkbox: Generate PIL Wrapper

コールバックとモデルパラメータのマッピングに基づき、この設定項目では以下のインターフェースが自動的に作成されます。

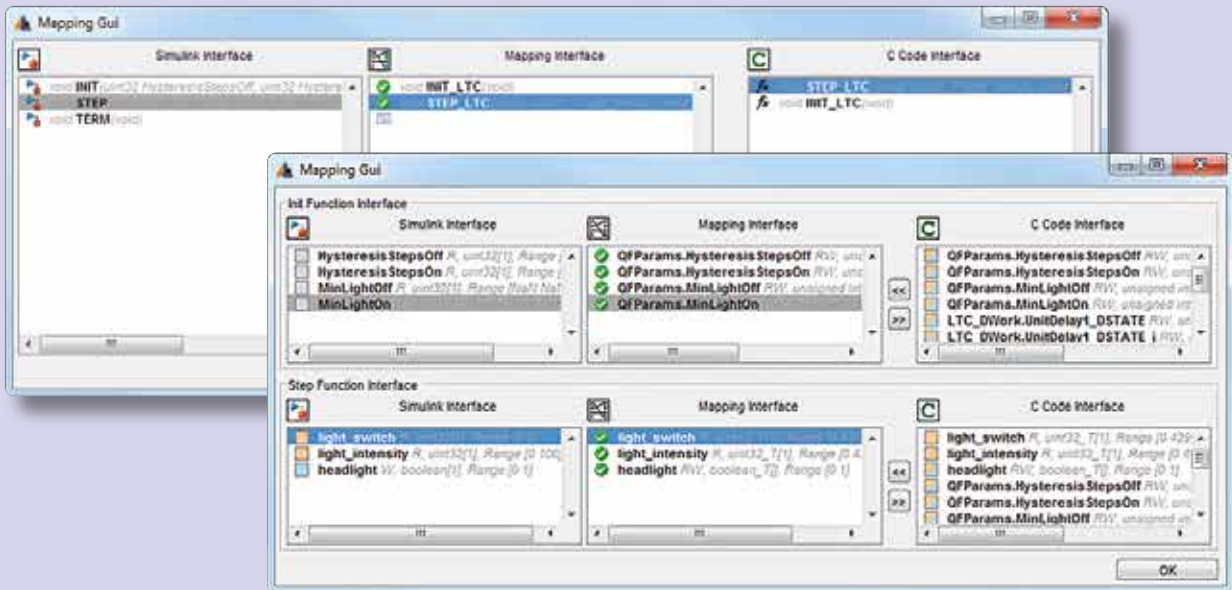
- ターゲットシステム上のコールバックの実装
- TRACE32 リモート API に対するモデル側インターフェース

その後、実行コードが生成されます。



## SimulinkとC インターフェースのマッピング

Mapping Simulink callbacks → C functions



Mapping Simulink parameters → C variables

### Checkboxes: Generate PIL Block and Switch to PIL Variant

モデルと TRACE32 リモート API 間の関数インターフェースが作成された後、PIL ブロックを作成し、シミュレーションモデルで表示することができます。TRACE32 が起動するとすぐに PIL シミュレーションの設定が完了します。

## メリット

### 1. 各種コード生成方法に対応

コールバックのインターフェースをダイナミックに設定する機能が装備されているため、ローターバツハのソリューションには自動生成されたコードだけではなく、手書きのコードも使用できます。

### 2. 新しいターゲットシステムに直接接続

TRACE32 はプロセッサアーキテクチャおよびコンパイラを幅広くサポートしています。開発者はターゲットシステムのスタートアップスクリプトを直接作成することができます。

ツールプロバイダによるソフトウェアアップデートを待たずとも、デバッガがターゲットと通信可能になり次第、PIL シミュレーションの準備が完了します。

### 3. 迅速なデバッグ

シミュレーション結果が要求仕様から逸脱している場合、C 関数を直接デバッグできます。

### 4. ライセンスポリシー

ハードウェアベースの TRACE32 デバッガや TRACE32 フローティングライセンスで仮想ターゲット用デバッガを既に利用している場合、TRACE32 PIL シミュレーションライセンスのみが必要です。TRACE32 インストラクションセットシミュレータをシミュレーションに利用する場合は、新たに TRACE32 シミュレータライセンスが必要になります。

## 2015年に新型ベースモジュールへの転換を完了

全ての TRACE32 PowerDebug モジュールに USB3.0 インターフェイスが装備されました。PowerDebug PRO モジュールにはギガビットイーサネットインターフェイスだけでなく、TRACE32 PowerTrace PX（新製品）や

TRACE32 PowerTRACE II 等のトレースモジュールの接続に使用する PodBus Express インターフェイスも装備されています。前世代のベースモジュールは今後も新しいチップ/プロセッサに使用できます。

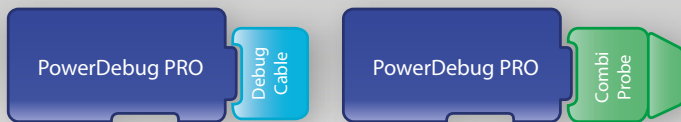
### PowerDebug USB 3

Previous generation PowerDebug USB 2



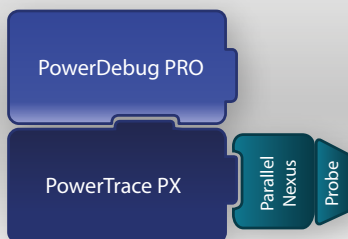
### PowerDebug PRO

Previous generation PowerDebug Ethernet or PowerDebug II



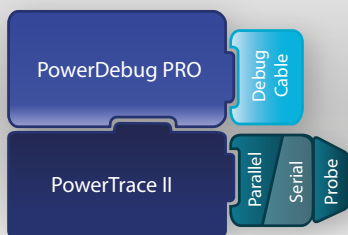
### PowerDebug PRO + PowerTrace PX

Previous generation PowerTrace Ethernet

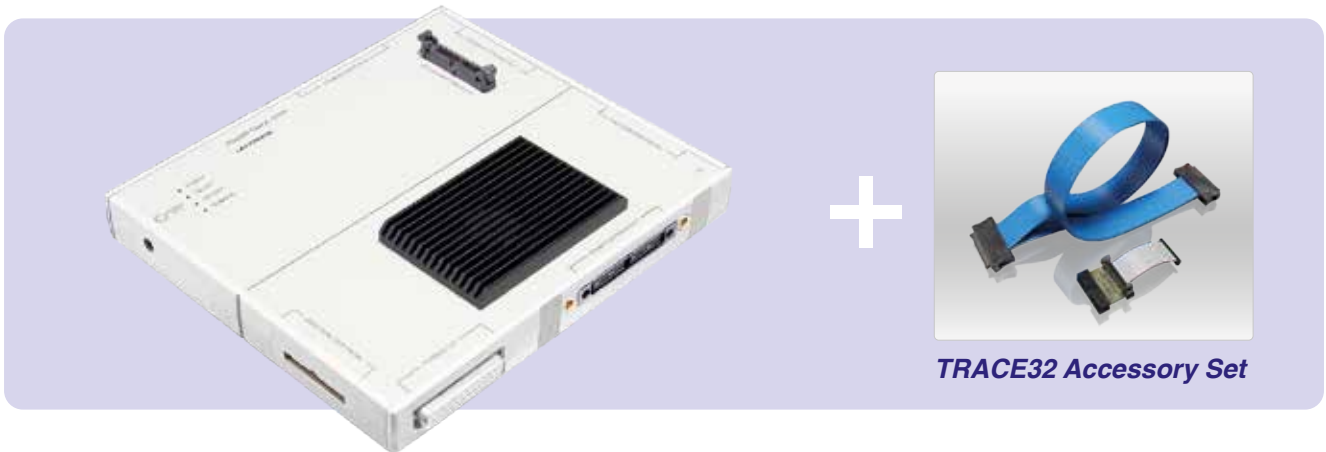


### PowerDebug PRO + PowerTrace II

Previous generation PowerDebug II + PowerTrace II



## PowerTrace Serial



TRACE32 Accessory Set

2016年の第2四半期から新型 TRACE32 ベースモジュール、PowerTrace Serial の提供を開始します。

ローターバッチは 2008 年以来、シリアルトレースプローブにより、シリアルトレースインターフェースおよび様々なプロセッサアーキテクチャをサポートしてきました。プローブの機能は以下の通りです。

- 最大 4RX チャンネル
- 最大 3 チャンネル、各チャンネル :6.25 Gbit/s
- 最大 4 チャンネル、各チャンネル : 3.125 Gbit/s
- Aurora を採用しているトレースプロトコル対応

複雑なマルチコアシステムの内部動作を完全に視覚化するには、現在のプロセッサに実装されているシリアルトレースインターフェースでは帯域幅が不足することもあります。そのため、データ転送速度とチャンネル数を増やしたインターフェースの開発がプロセッサメーカーで始まっています。さらに、PCI Express がトレースエクスポートインターフェースとして急速に注目を集めるようになっていきます。

### PowerTrace Serial の機能

Trace32 PowerTrace Serial ベールモジュールは、トレース規格の最新要件を満たすように設計されています。

- 最大 8 チャンネル
- 各チャンネル : 最大 12.5 Gbit/s
- Xilinx Aurora、PCI Express、その他プロトコル
- トレースメモリ : 4 ギガバイト

上記機能の実装には大容量かつ高性能な FPGA の搭載が必要なため、最新の PowerTrace Serial はオールインワンソリューションとして設計されており、これまでは別売りであったトレースプローブは PowerTrace Serial に統合されています。ローターバッチは PowerTrace Serial のターゲットへの接続に必要なアクセサリセットを提供しています。通常のアクセサリセットには、そ

れぞれのターゲット環境に適したフレックスケーブルと必要なアダプタが含まれます。PowerTrace Serial には導入時に 1 種類のプロセッサアーキテクチャのコアトレースデコーディングのためのライセンスが付与されています。その他のアーキテクチャのトレースに対応する場合でも、ライセンスを追加することで後日、簡単に対応できます。

### PowerTrace Serial コネクタ

PowerTrace Serial には以下のインターフェースが装備されています。

#### シリアルトレースポート 0 (Samtec ERF8, 40pin)

- Aurora を採用しているトレースプロトコル対応
- 6 RX チャンネル
- リファレンスクロック 0.325-6.25 GHz

#### ターゲットデバッグポート (34pin MIPI コネクタ)

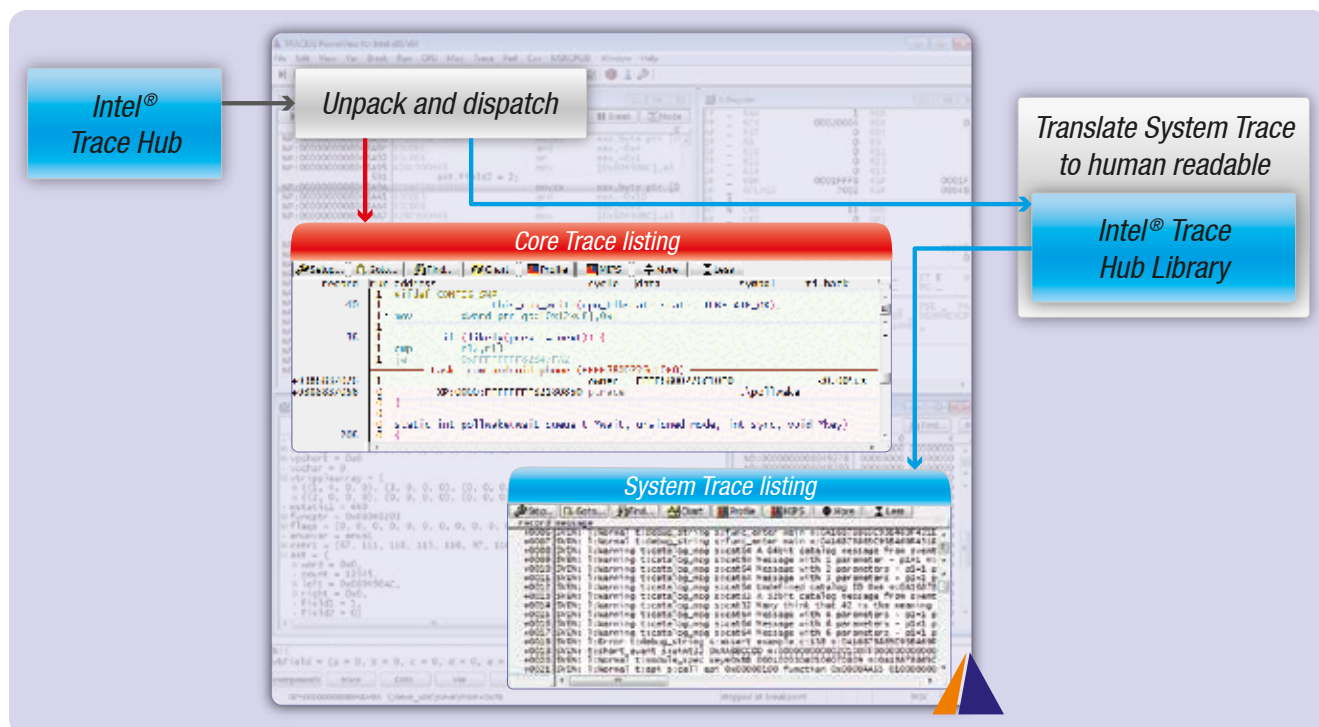
デバッグシグナルが 40pin Samtec コネクタ (JTAG/SWD/cJTAG) にも配線されている場合、TRACE32 デバッグケーブルもこのポートに接続することができます。

#### シリアルトレースポート 1 (Samtec ERM-ERF, 80pin)

- 各種プロトコル対応
- 8RX/TX チャンネル
- リファレンスクロック 0.325-6.25 GHz/10-500 MHz

両方のトレースポートが装備されているため、現在の設計だけでなく、今後のプロジェクトにも PowerTrace Serial を使用できます。

# Intel® Trace Hubサポート



2016年5月、TRACE32はIntel® Trace Hubと関連フレームワークのサポートを開始します。

## Intel® Trace Hub

Intel® Trace Hub (Intel® TH)は、Intelの新しいハードウェアプラットフォーム上のトレースインフラストラクチャの名称です。Intel® Trace Hubにより以下の機能が可能となります。

1. 各コアからのIntel® プロセッサトレースデータと多種ソースからのシステムトレース情報に共通タイムスタンプを提供
2. MIPI STPv2.1 プロトコルを使用してシングルトレースストリームに全てのデータをマージ
3. 選択したトレースの宛先にストリームを転送

TRACE32などのデバッグ&トレースツールがIntel® Trace Hubを簡単にサポートできるように、Intelはソフトウェアフレームワークも提供しています。

## Intel® Trace Hub 設定 API

Intel® Trace Hub 設定 APIは、デバッグツールを通してトレースインフラストラクチャの設定の簡素化を可能にします。TRACE32は、プラットフォーム特有のプログラ

ミングシーケンスを認識する必要はなく、Intel® TH 設定 API に設定リクエストを送信することができます。その後、API が適切なプログラミングシーケンスを提供するため、TRACE32 は JTAG インターフェースを経由し、このシーケンスを制御レジスタに書き込みます。

## Intel® Trace Hub Library

Intel® TH Library はシステムトレースパケットを可読性の高いトレースメッセージに変換します。Intel® TH は全てのトレース情報を MIPI STPv2.1 に圧縮しているため、最初に解凍した後、デコーディングのために転送します。

- Intel® Processor Trace パケットは TRACE32 で表示や評価のために直接デコードされます。
- システムトレースパケットは Intel® TH Library に転送されます。デコードされ、可読性の高い形式に変換されたトレースメッセージは、TRACE32 内の専用のコマンドグループで表示し、評価することができます。

TRACE32 内の各種トレース情報の時間相関ビューの使用により、プラットフォーム全体の動作一覧を素早く確認できます。

## TRACE32 TCFエージェント



2015年10月以降、TRACE32はTCFエージェント機能も提供しています。Wind River WorkbenchまたはEclipseデバッガをIDEとして、TRACE32をデバッグバックエンドとして使用することができます。

### TCF

Target Communication Framework (TCF) は、IDEとターゲットシステム間の統一デバッグ通信プロトコルを規定することを目的に、プロトコルフレームワークとしてEclipse Foundationにより開発されました。

TCF通信モデルはサービスという考え方に基づいており、そのサービスは関連するコマンド、イベント、該当するセマンティクスの集合になります。例えば、メモリサービスはメモリを読み書きするコマンドとイベントの集合を定義します。

TCFは一連の標準サービスを定義する一方、フレームワークには独自のサービス定義を行うことが可能です。

### TRACE32 TCF

TRACE32ソフトウェアは、TCFエージェントとして起動するとTCP/IPを経由してWind River WorkbenchまたはEclipseデバッガにサービスを提供します。

リクエストされたサービスは、ターゲットに接続されているデバッガとTRACE32によって操作されます。JTAG経由でターゲットシステムに接続するハードウェアベースのデバッガであるか、仮想ターゲットと通信する純粋なソフトウェアデバッガであるかは、関係ありません。現在TRACE32は、デバッグと関連する従来の各種サービスを提供しています。お客様からのフィードバックやご要望に応じて、特殊なTRACE32サービスの開発が将来的に計画されています。ローターバツハが提供するTRACE32 TCFプラグインを使用すると、Wind River

WorkbenchやEclipseデバッガからTRACE32を設定、起動することができます。

### Wind River Workbench

ローターバツハは、Wind River Workbenchを使用する開発者向けに統合されたデバッグソリューションの提供を今年から開始します。

### Eclipse デバッガ

GDBベースの接続では、GDBがサポートするプロセッサアーキテクチャおよびコンパイラにより利用できる環境が限定されていました。TCFサービスを使用することでTRACE32がサポートする各種プロセッサアーキテクチャ、コンパイラをEclipseでデバッグするためのオープン通信インターフェースを提供できるようになりました。

### TRACE32: Wind River向けサポート

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 Undo DBフロントエンド

2015 年半ば以来、UndoDB リバーシブルデバッガ用のフロントエンドとして TRACE32 を使用することができます。ARM/Cortex だけでなく、Intel® x86/x64 アーキテクチャもサポートされています。

UndoDB ターゲットサーバを使うことで、Linux の開発者はアプリケーションプロセスのデバッグだけでなく、その実行の詳細な記録もできます。

TRACE32 フロントエンドはデバッグプロセスの制御に加えて、UndoDB ターゲットサーバの記録済みデータを TRACE32 の GUI で表示するタスクも引き継ぎます。トレース記録と同様に、開発者は再実行、逆実行の両方を使ってアプリケーションをデバッグすることができます。(リバーシブルデバッグ)

この機能を利用すると、アプリケーションプロセスのエラーを迅速かつ容易に探し出すことができます。

記録済みデータのデバッグが有効であることを視覚的に強調するために、TRACE32 のステータスラインに表示される記録時間は反転表示されます。また、ソースリストのデバッグボタンは黄色で表示されます。TRACE32 GUI は自動的に以下の表示モードに切り替わります。

- ソースリストの命令ポインタは、表示される記録時点の値にリセットされます。
- 記録時点のメモリ内容、変数が表示されます。

The screenshot shows the TRACE32 GUI with three callouts: 'Forward & backward debugging' pointing to the execution control buttons, 'Variable reconstruction' pointing to the variable window, and 'Recording point displayed' pointing to a yellow highlight in the source code. To the right, a diagram shows a 'Target' (represented by a chip and a penguin) connected to an 'UndoDB server' via 'Protocol: GDB' and 'Link: TCP/IP'.

アドレスの変更またはメーリングリストからの退会をご希望の場合には下記へご連絡ください。

[info@lauterbach.co.jp](mailto:info@lauterbach.co.jp)



**LEADING** through Technology