

TRQerS 導入概要

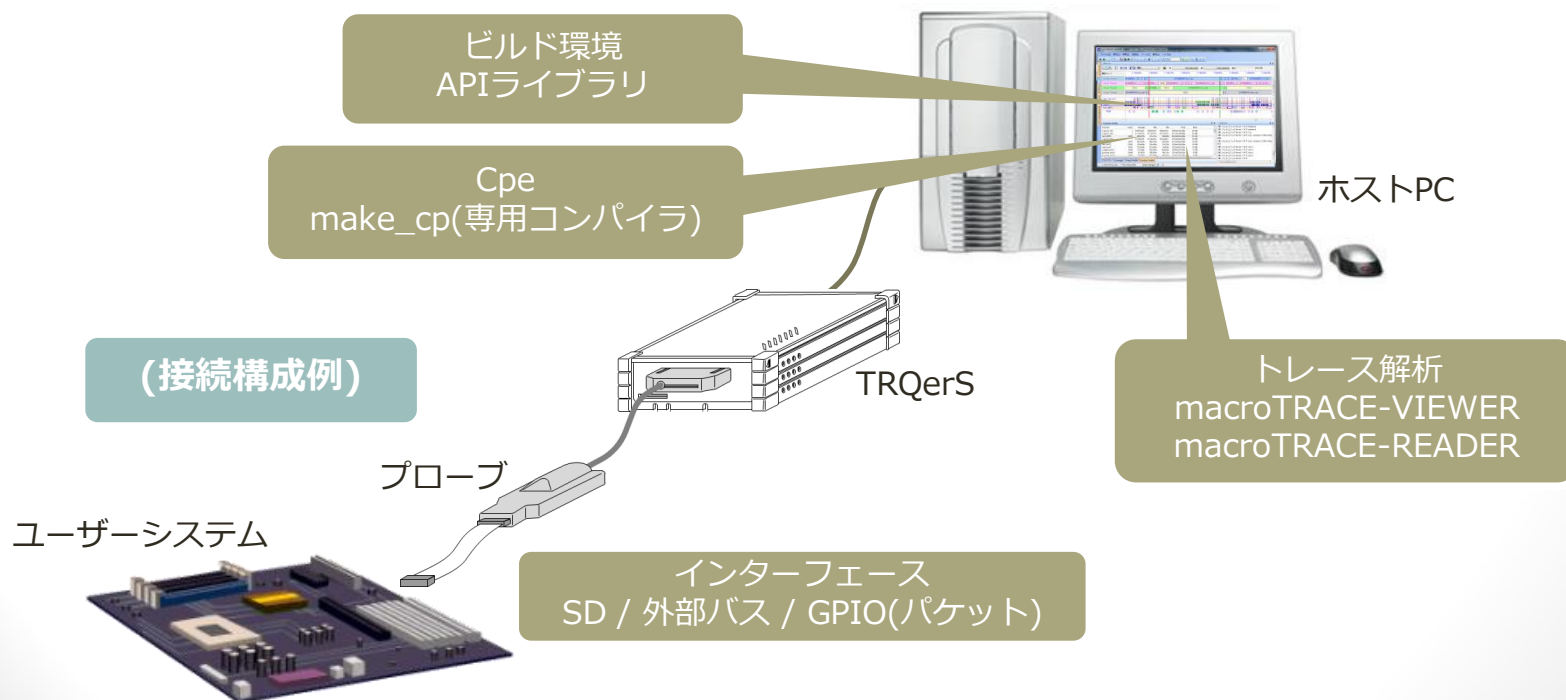
株式会社DTSインサイト
(ESC-APN-035-03 Apl,01,2017)

システムマクロトレースの特徴、 製品構成と導入フロー

システムマクロトレースの特徴

- システムマクロトレース

- printfログ出力の発展形 (printfより高速。文字列/タグ情報/タスク遷移/関数遷移)
- ハードウェアインターフェース(SD,外部バス(独自),GPIO(独自))、
または、ソフトウェアモデル(ユーザーシステムのストレージにトレースデータ保存)
- ユーザープログラムに、トレース出力用のAPIを組み込んで実現
- トレース / プロファイル / 関数カバレッジ / ヒストグラム / ポートモニタ



製品構成(ソフトウェア)

- インストールキット (有償)

Model	Description
TLA000	- TLA000/V : 標準ソフトウェア (macroTRACE-VIEWER, macroTRACE-READER, Cpe, make_cp) - TLA000/R : リーダーソフトウェア (macroTRACE-READERのみ、※トレース取得不可) - ノードロックライセンス(/N)、または、フローティングライセンス(/F)
TLA002/S	- 標準ソフトウェア (ソフトウェアモデル ※ハード不要、1年ターム) (macroTRACE-VIEWER, macroTRACE-READER, Cpe, make_cp) - ノードロックライセンス(/N)、または、フローティングライセンス(/F)

- APIライブラリ (無償ダウンロード)

対応ターゲットOS	インターフェース
Android	SD、外部バス、GPIO(パケット)、USB(Device)
	ソフトウェアモデル
Linux	SD、外部バス、GPIO(パケット)、USB(Device)
	ソフトウェアモデル
RTOS	SD、外部バス、GPIO(パケット)
Windows	SD

※ターゲットOSごとにAPIライブラリとリファレンスマニュアル、ターゲットOSのインターフェースごとにポーティングガイドを提供。

製品構成(ハードウェア)

- 本体

Model	Description
AP512/4G	TRQerS本体 (ACアダプタ、USBケーブル付属)

- プローブ

Model	Description
AQ720	SDインターフェースプローブ (フレキケーブル含む)
AQ721 + AQ730	パケットインターフェースプローブ (AQ730: パケットインターフェース対応フレキケーブル)
AQ721 + AQ731	外部バスインターフェースプローブ (AQ731: 外部バスインターフェース対応フレキケーブル)

※TRQerS本体と、ユーザーシステムインターフェースに合わせたプローブの組み合わせで使用する。
※ソフトウェアモデル(TLA002)は、トレースハードウェア不要。(ターゲットのメモリ/ストレージ上にトレースデータを保存)

導入フロー

インターフェース 選択

- ユーザーシステムインターフェース選択
(SD / 外部バス(8bit/16bit)) / GPIO(Packet)、または、ソフトウェアモデル)
- 該当するAPIライブラリのダウンロード取得

ポーティング

- APIライブラリのビルド ~ APIライブラリの組み込み
- SMTデバイスドライバの実装(ターゲットOSによって実装手順が異なります)
- タスクスイッチ情報出力の組み込み
- API関数組み込み後のビルド

ここまでの組み込みで、
print文、タスクスイッチ情報の取得が可能

チェックポイント 挿入

- Cpeを使用して、トレース対象関数の選択(チェックポイント挿入)
- CP挿入後のビルド (専用コンパイラmake_cp or 汎用コンパイラ)
- ユーザーシステムのプログラムの更新 (ダウンロード)



Cpe / make_cp

ここまでの組み込みで、
タスクスイッチ情報等のほか、関数トレースが可能

トレース

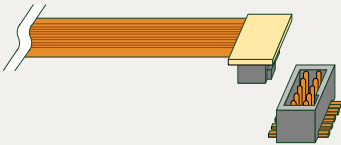
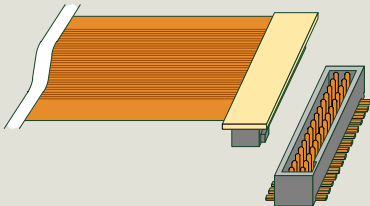
- TRQerSとユーザーシステムの接続
- macroTRACE-VIEWER (トレース取得、解析、トレースログ保存)
- macroTRACE-READER (取得済みログファイルの読み込み~解析)



macroTRACE-
VIEWER/READER

各フローでおこなう作業について

ユーザーシステムインターフェースの選択

イメージ	インターフェース	備考
	SD (AQ720)	• 一般のmicroSD仕様 • 最大50MHz
	GPIO (パケット) (AQ721+AQ730) ※独自I/F	• 1.27mmピッチ10pin • CLK, xSYNC, D[3:0], TVCC, GND • 最大100MHz
別途提供するパケット出力論理(Verilog)をFPGA/SoCに実装し、専用I/F化も可能		
	外部バス (AQ721+AQ731) ※独自I/F	• 1.27mmピッチ30pin • 8bitバス A[2:0], D[7:0], xCS, xWR, xTRESET, TVCC, GND • 16bitバス A[2:0], D[15:0], xCS, xWR, xTRESET, TVCC, GND • オーバーヘッド小

ソフトウェアの準備

- ソフトウェアインストールキット TLA00x (有償)

Application	Description	Install
macroTRACE-VIEWER	トレース取得/解析ソフトウェア	ライセンス取得時に発行されるID,PWでログインし、最新版インストーラをダウンロード。 インストーラを使用してインストール。
macroTRACE-READER	トレースデータ解析ソフトウェア	
Cpe	チェックポイント挿入ツール	
make_cp	チェックポイント挿入用コンパイラ (プリプロセッサ)	

- APIライブラリ(無償DL)

インターフェースに合わせて
必要なソース(ドキュメント)を
ダウンロード。
ビルド環境(ソース環境)に展開。

https://www.dts-insight.co.jp/support/support_advice/?m=AdviceSmtapilib&item=1

Linux環境 APIライブラリ

ターゲットOS	APIライブラリ サンプル・ソース、リファレンス・マニュアル	バージョン	
Linux	APIライブラリ、ドライバサンプル・ソース	3.10	ダウンロード
	APIライブラリ リリース・ノート	3.10	readme.txt
	APIライブラリ リファレンス・マニュアル	第2版	ダウンロード
	SDカードインターフェース	第7版	ダウンロード
	外部バス インターフェース	第7版	ダウンロード
	GPIO インターフェース	第8版	ダウンロード
	ポーティング・ガイド		
	USB(Device) インターフェース	第7版	ダウンロード
	アプリケーション対象 SDカードインターフェース	第6版	ダウンロード
	ソフトウェアモデル	第6版	ダウンロード
	Linux用APIライブラリコピースクリプト (B版)	rev1	ダウンロード
	Linux用APIライブラリコピースクリプトについて	rev1	ダウンロード

例: Linux環境 外部バスI/Fでご利用頂く場合、下記の組み合わせでダウンロードしてください。
APIライブラリ、ドライバサンプル・ソース (APILib_Linux_310.tar.gz)
APIライブラリ リファレンス・マニュアル (Linux-Android_APILibrary_Reference_jpn.pdf)
外部バスインターフェース ポーティング・ガイド (Linux_ExternalBus_SMT_Porting_Guide_jpn.pdf)

ポータリング

- APIライブラリと一緒に提供される、ターゲットOS/インターフェースごとの「ポータリングガイド」にそって作業を行ないます。

例: Linux / 外部バス

1. APIライブラリのインストール(展開)
2. カーネルランドAPIライブラリソース組み込み ~ カーネルビルド
(プロセス遷移情報を取得する場合、コンテキストスイッチ関数を挿入)
3. ユーザーランド(アプリケーション)APIライブラリ(libsmt.a)ビルド
~ アプリケーションにリンク
4. SMTデバイスドライバ(smtModule.ko)の実装 ~ ドライバのビルド
5. デバイスファイルの作成
6. アプリケーションに、APIライブラリ(libsmt.a)をリンク

※H/W上で、パケット論理を実装している(FPGAなど)場合、外部バスI/FのAPIライブラリを使用する。

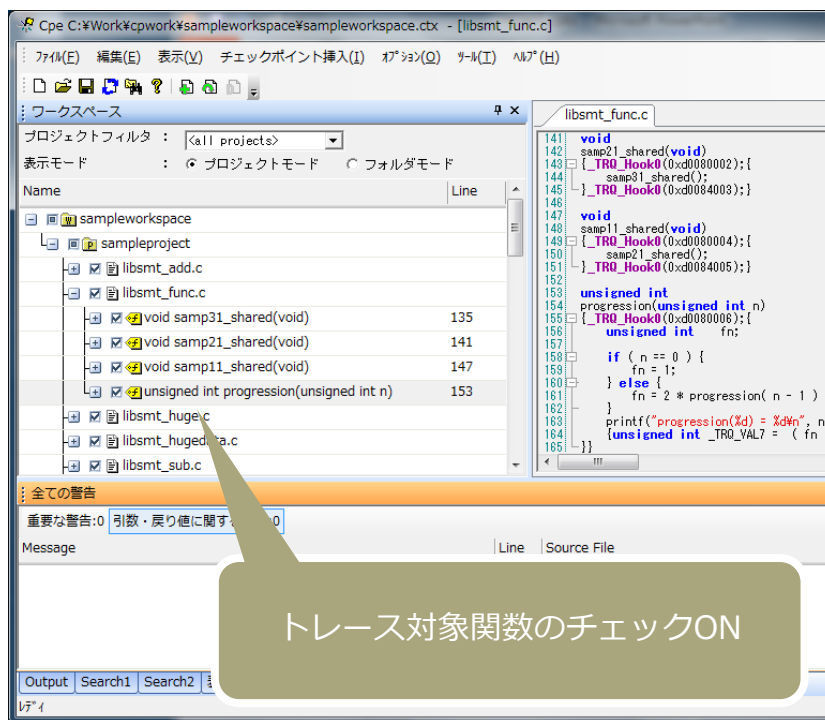
例: RTOS / GPIO

1. APIライブラリのインストール(展開)
2. SMTデータ出力関数の実装 (初期化、SMTデータ出力中の割り込み禁止)
3. APIライブラリのビルド
4. APIライブラリをユーザープログラムにリンク

※OSレスシステムの場合、RTOS用のAPIライブラリを使用する。

チェックポイントの挿入

- Cpe (Check Point Editor)



ワークスペースファイルの選択

専用コンパイラ版 (make_cpを使用)

- ソースを直接編集しない
- defineマクロも解析
- コンパイラコマンドを変更可能なIDEであること
- 対応コンパイラ依存あり
 - ARM社C/C++コンパイラ
 - GNU Compiler Collection
 - Microsoft社Platform Builder
 - Renesas社 SuperH RISCコンパイラ

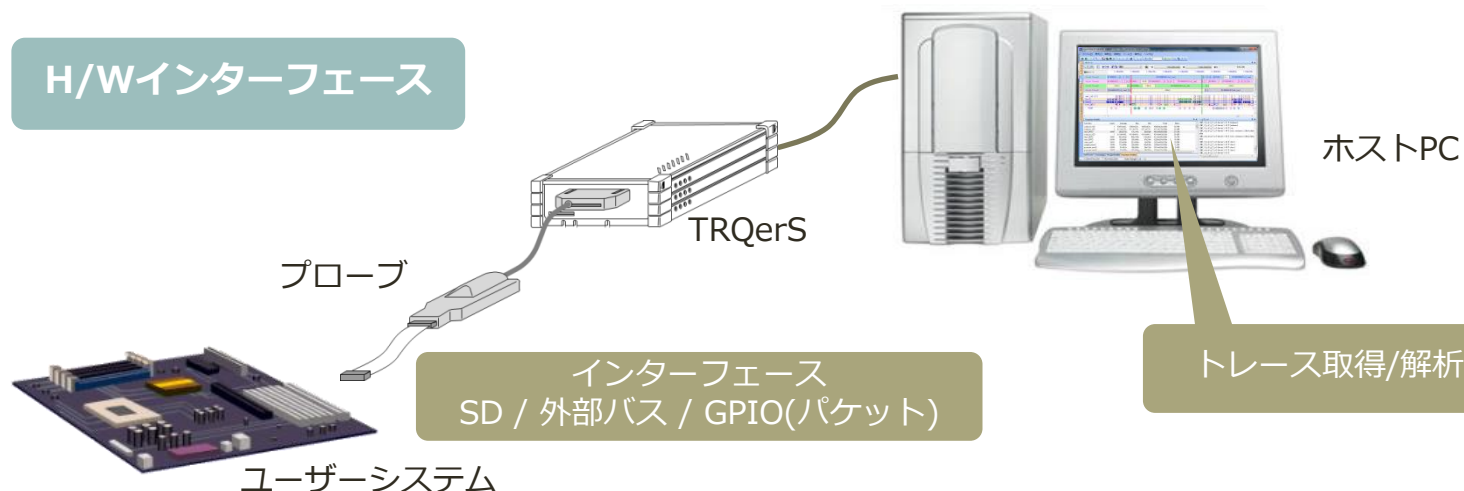
汎用コンパイラ版

- ソースファイルに直接CP挿入
- コンパイラやIDEに依存しない
- 対応コンパイラ依存なし
- defineマクロの解析できない(手動で設定)

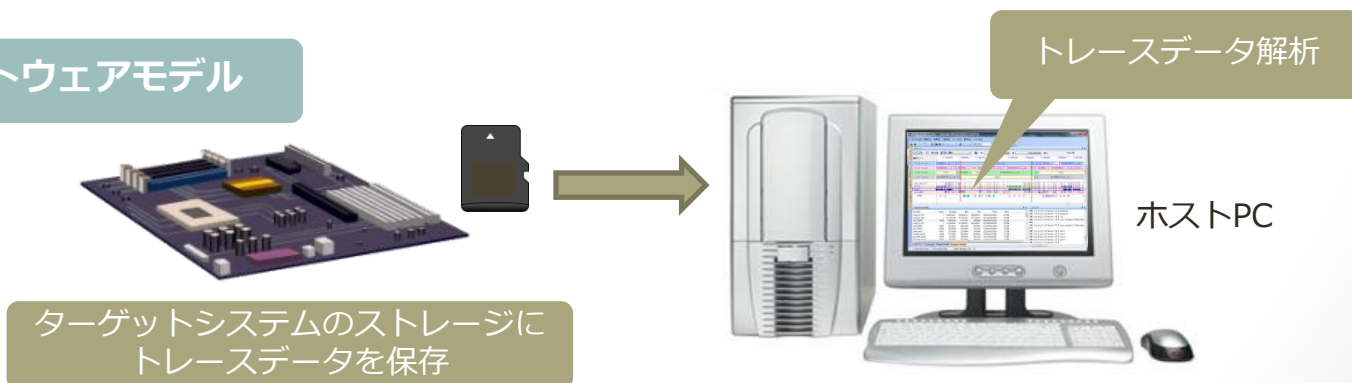
※チェックポイント挿入後、ビルド → ターゲットシステムにロード → 関数トレースが可能
※専用コンパイラ版は、MakefileのCCを'make_cp'に置き換えてビルド
※ワークスペースファイルは設定保存だけでなく、Viewerに登録して関数名の解決を行なう

ユーザーシステムとの接続

H/Wインターフェース



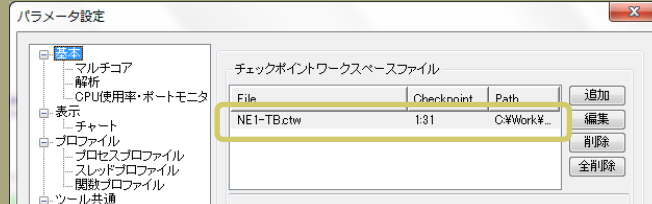
ソフトウェアモデル



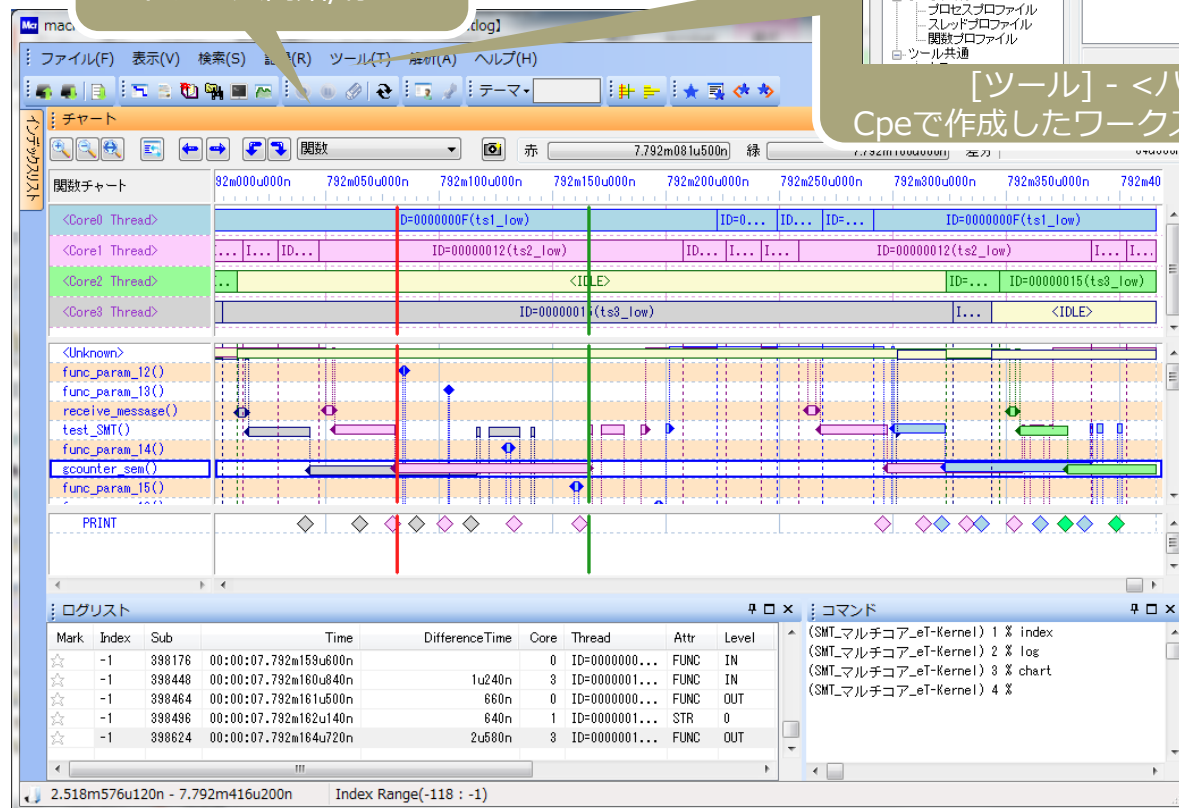
トレースデータ取得



トレース開始/停止



[ツール] - <パラメータ設定>
Cpeで作成したワークスペースファイルを指定



タスク/関数遷移の可視化

- ・タスク遷移
- ・関数遷移

print出現箇所

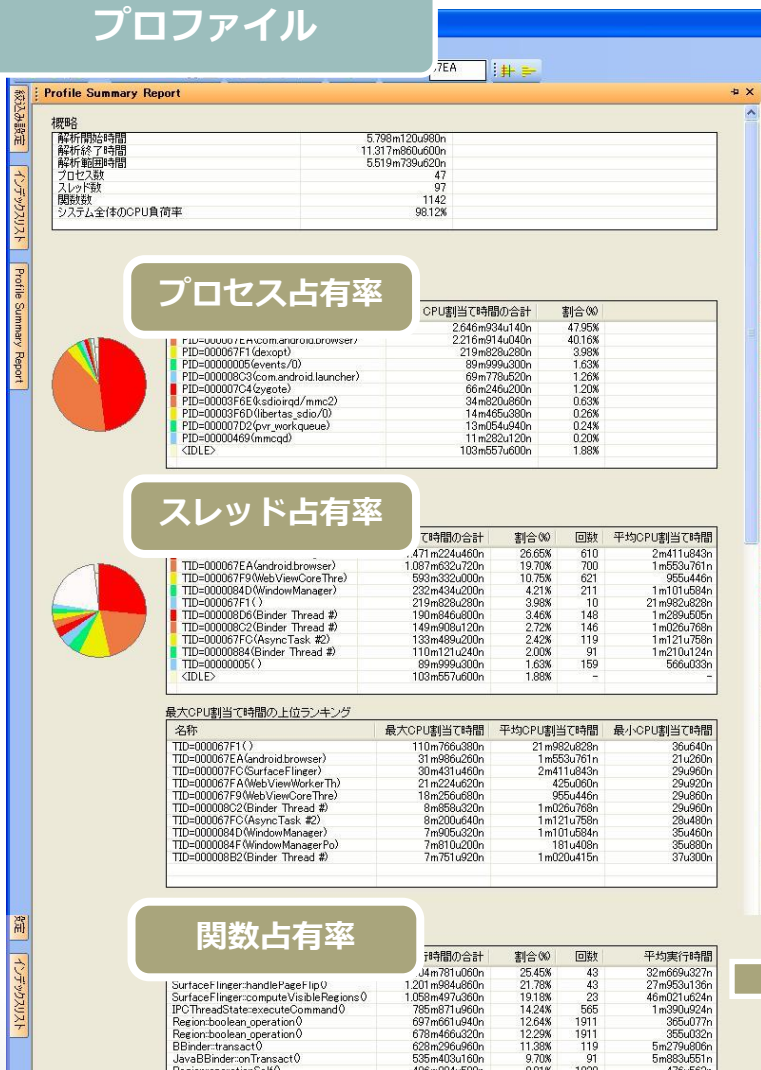
リスト表示

実行時間の可視化

※検索/絞り込み/フィルタ/強調/コアごと等の表示 → 自分が着目したい関数を絞り込んで解析
※トレースログを保存 → 別の担当者様が解析(要Readerライセンス)

トレースデータ解析

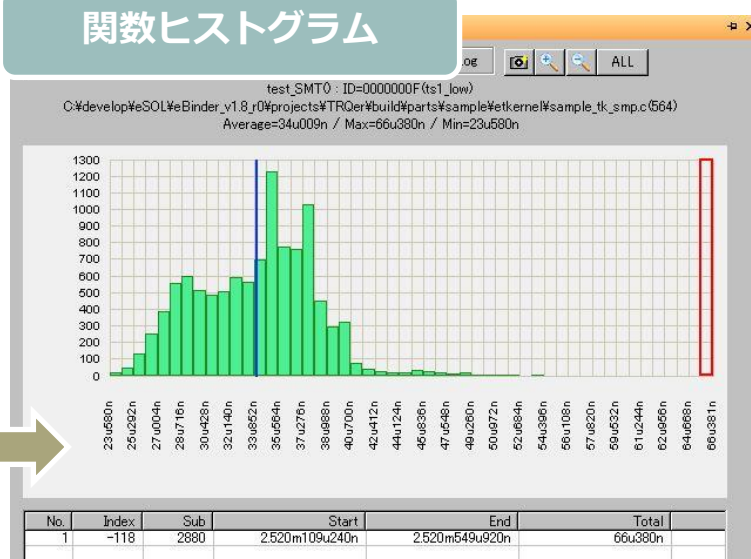
プロファイル



関数カバレッジ

Function	Detail
Project	File
File	Func
Level	Count
DDR_COM.c	sampleApp main.c int_abort() IN 0
DDR_ZYNQ_GPIO.c	sampleApp main.c int_abort() OUT(229) 0
DDR_ZYNQ_TIMER.c	sampleApp main.c Fibo() IN 46981
DDR_ZYNQ_QUART.c	sampleApp main.c Fibo() OUT(244) 46981
hw_init.c	sampleApp main.c SndTask() IN 0
main.c	sampleApp main.c SndTask() OUT(286) 0
void int_abort(void)	sampleApp main.c RcvTask() IN 0
IN	sampleApp main.c RcvTask() OUT(315) 0
line 229	sampleApp main.c MainTask() IN 0
unsigned long Fibo	sampleApp main.c MainTask() OUT(334) 0
IN	sampleApp main.c undefined_instru... IN 0
OUT	sampleApp main.c undefined_instru... OUT(350) 0
line 244	sampleApp main.c initpr() IN 0
void SndTask(VP_	sampleApp main.c initpr() OUT(372) 0
void RcvTask(VP_	sampleApp main.c main() IN 0
void undefined_ins	sampleApp main.c main() OUT(429) 0
void initpr(void)	
int main(void)	

関数ヒストグラム



- 改訂履歴

Rev.	Date	Description
1.0	2015/10/26	新規作成
2.0	2015/12/25	細部修正、公開
3.0	2017/04/01	新商号対応