



LDRA

Automated unit testing
with TBrun™

Windows Embedded CE ターゲット上での動的テストを自動化

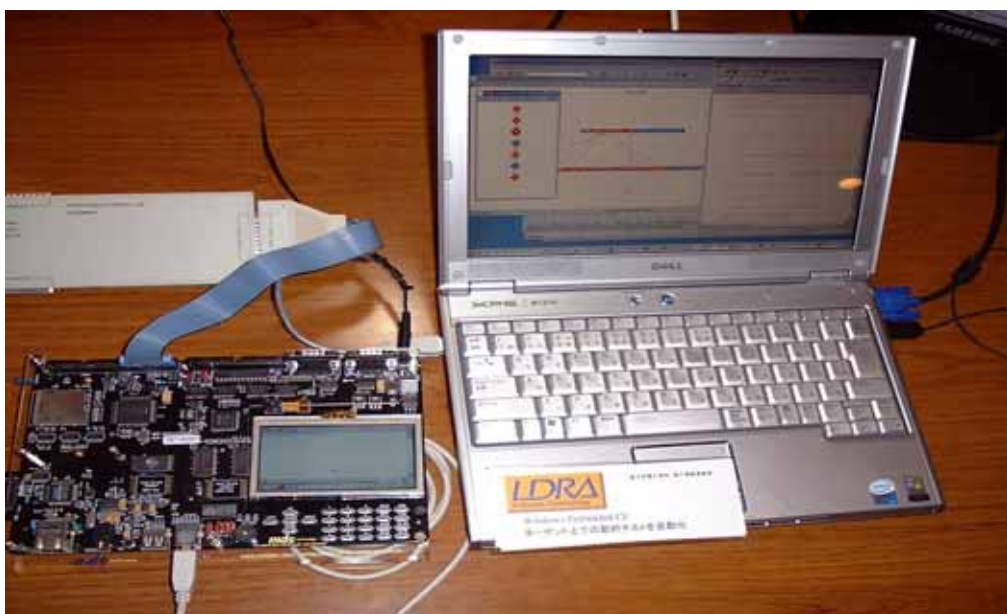
LDRA テスト自動化ツール インテグレーション for Windows Embedded CE ターゲット

本資料では、LDRA テストツールを用いて Windows Embedded CE 6.0 ターゲット上でアプリケーションのテストを自動実行し、結果判定とテストのカバレッジ測定の自動化を行う方法を紹介しています。

【ターゲット】トライポッドワークス(株) : TriBOARD EVA9 評価ボード

Windows Embedded CE ターゲットでテストを自動化する上で、以下のツールを統合し活用しました。

- LDRA 社 Testbed/TBrun テスト自動化ツール
 - ・ タグ付け手法により、あらゆるターゲットに対応できる
 - ・ 独自のタグ付けは、最小限の負荷で測定
 - ・ リグレッションテストの自動化により工数を削減
- ローターバッハ社 JTAG デバッガ
 - ・ 高速、高い安定性。組込みデバッガにありがちな不安定要素を排除
 - ・ ターゲット上にデバッグ用モニタが不要で、直ぐにデバッグ・テストができる
 - ・ 実行前のプロセスにブレークを設定（仮想空間のアプリを扱うターゲットのテストに有効）
- T-VEC 社 テストベクタ自動生成ツール
 - ・ 要求モデル検証から入力と期待値を自動生成
 - ・ 要件通りにコードが正しく動作することを検証できる
 - ・ 要件からテストまでのトレーサビリティをサポート



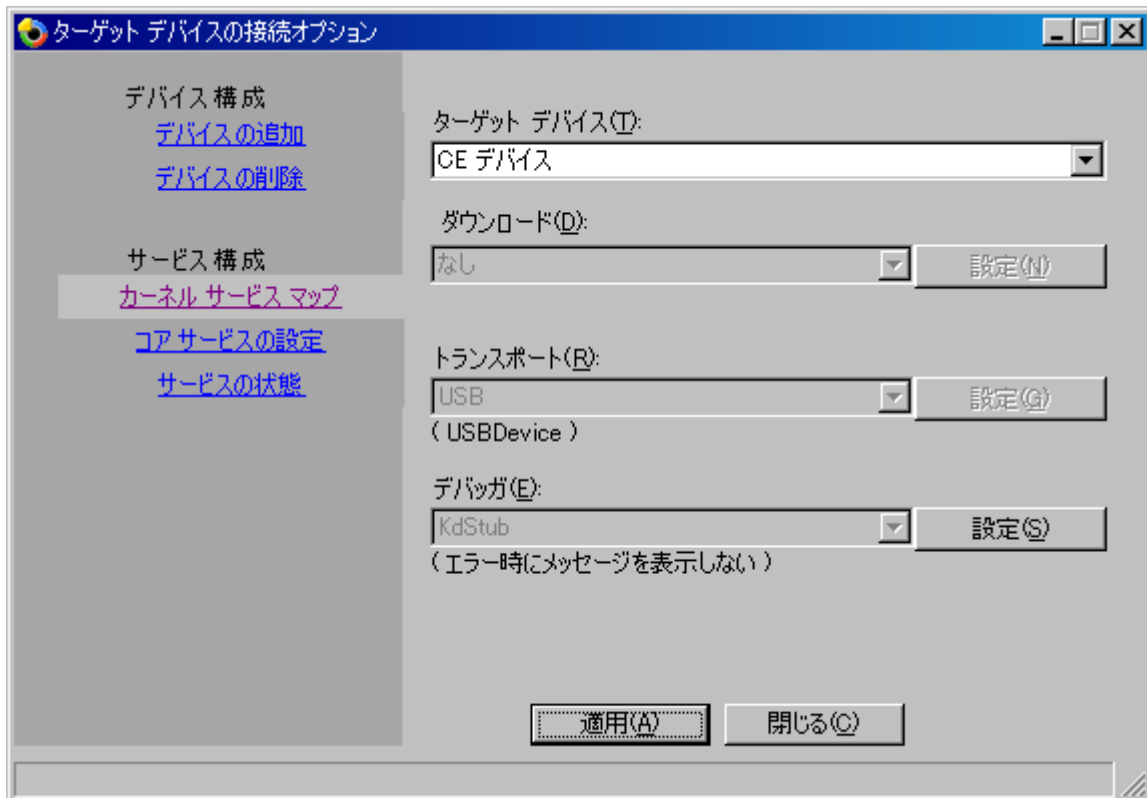
【全体の流れ】

Visual Studio 2005 とターゲットを接続し、ターゲット上で Windows CE が起動していることを前提とします。テスト対象ソースコードの静的解析・カバレッジ測定用のタグ付けを、ホスト側の LDRA テスト自動化ツール (Testbed) で実行。テストベクタ自動生成ツール (T-VEC) で要件から生成されたテストベクタを LDRA/TBrun に読み込み。LDRA/TBrun からテストドライバを生成させ、コードと合わせてコンパイル実行。JTAG デバッガ起動、ターゲットにアプリケーションをダウンロードし、テスト実行。得られるテスト結果と実行履歴をホスト側に収集し (JTAG デバッガから) LDRA/Testbed ツールによって結果判定、カバレッジ解析を行う。

(タグ付けとはソースをコピーし、カバレッジを測定するためのプローブ (関数) を挿入すること)

【手順】

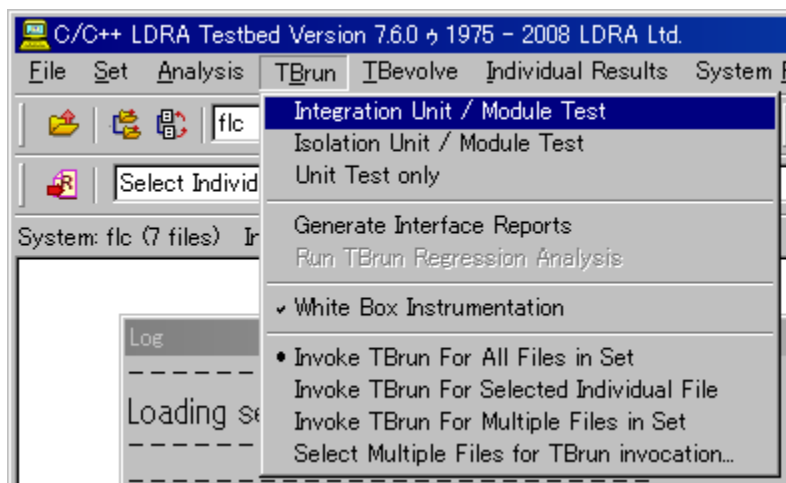
Visual Studio 2005 において、以下のオプションでターゲットデバイスと接続し、Windows CE を起動。
(ここでは、USB 接続)

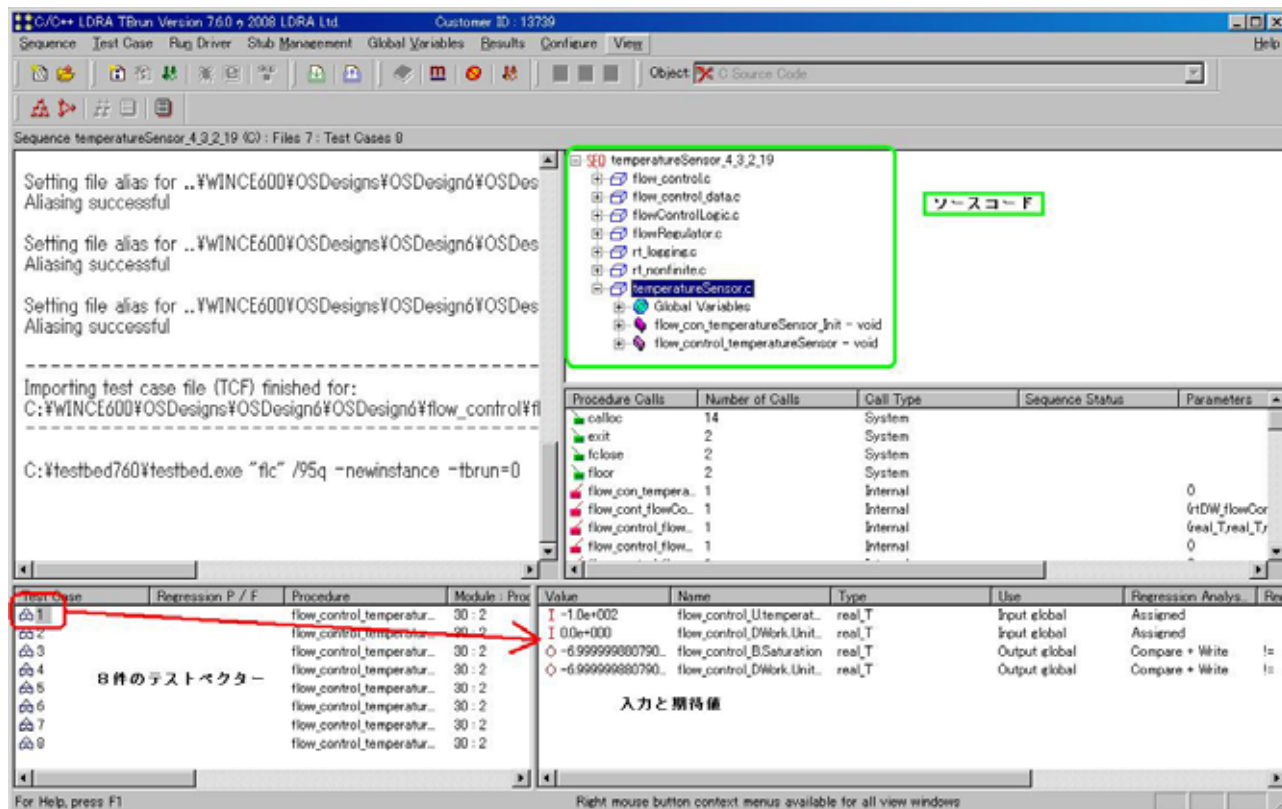


- 1 . アプリケーションの作成。Visual Studio 2005 を使って、Windows Embedded CE 6.0 用のアプリケーション用プロジェクトを作成 (今回は、独自のサンプルコードを使用)。
- 2 . LDRA/ Testbed を起動し、[File]メニューから[Select File or Set from TCF]を選択して、事前にテストベクタ生成ツール(T-VEC)で作成した *.tcf ファイルをツールにインポートすることで、テスト対象ソースコード、テストベクタなど読み込ませます。

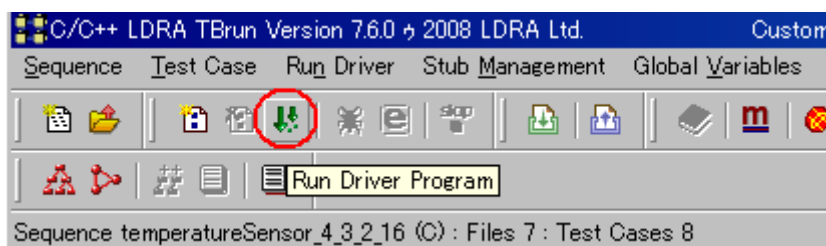


- 3 . LDRA/ TBrun テストドライバー生成・実行ツールの起動。(Integration Unit / Module Test)



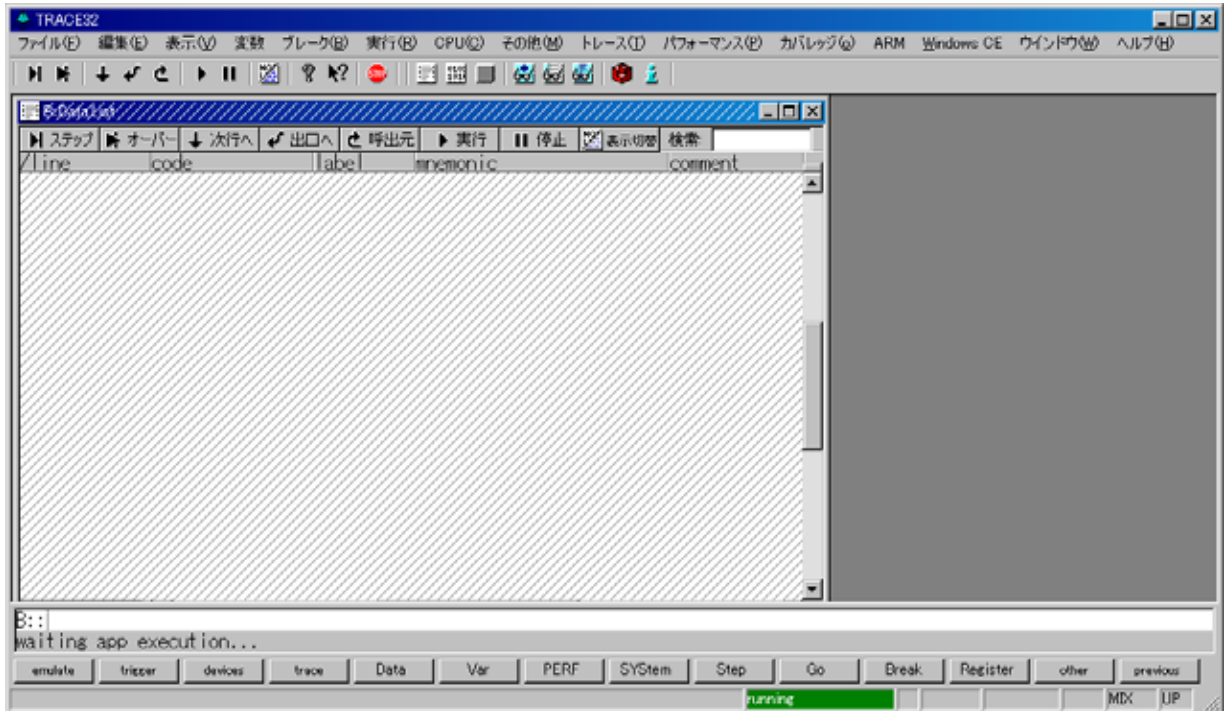


- 4 . ビルドの設定。LDRA/TBrun から、テスト対象コード・テストドライバーに対して、バッチファイルを用いてビルドするための準備。
- 5 . 実行設定。LDRA/TBrun から、バッチファイルを使ってテストを実行するための準備。バッチファイルには、JTAG デバッガ起動処理などを記述している。
- 6 . カバレッジの測定レベルを設定。
- 7 . Run Driver Program ボタンを押下することで、テストドライバー生成、コンパイル、JTAG デバッガ起動、アプリケーションのダウンロード、実行前のプロセスにブレークポイント設定といった作業をバックグラウンドでスクリプト実行。(上述で設定したバッチファイルや JTAG デバッガのコマンドファイルが実行される)



アプリケーションのダウンロードは、ホストの Release ディレクトリをターゲットのルートディレクトリにマウントすることで、最新の実行ファイルをターゲットで実行できます。

8 . 以下、ローターバッハのデバッガ画面で、ブレーク待ちの状態です。



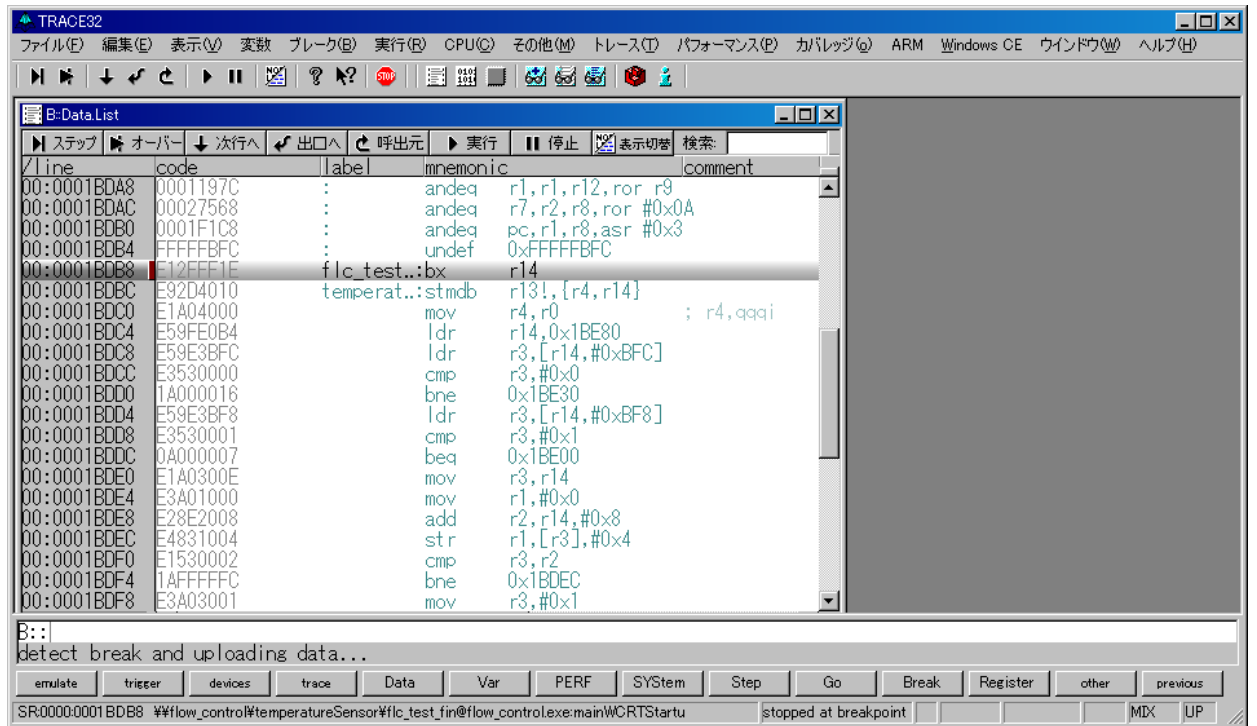
Visual Studio 2005 のターゲットコントロールから、Windows CE リモートコンソールを立ち上げて、テスト対象アプリケーションを実行。

```
Windows CE Command Prompt
<command>: Shell commands ('?' for shell help)
'.<command>': Debugger commands ('.?' for help)
'!<command>': Debugger extension commands
Ctrl-Q: Abort pending command
Ctrl-L: Clear all
Ctrl-A: Select all
Ctrl-F: Find (F4: Search forward, Shift-F4: Search backwards)

Windows CE>s flow_control
```

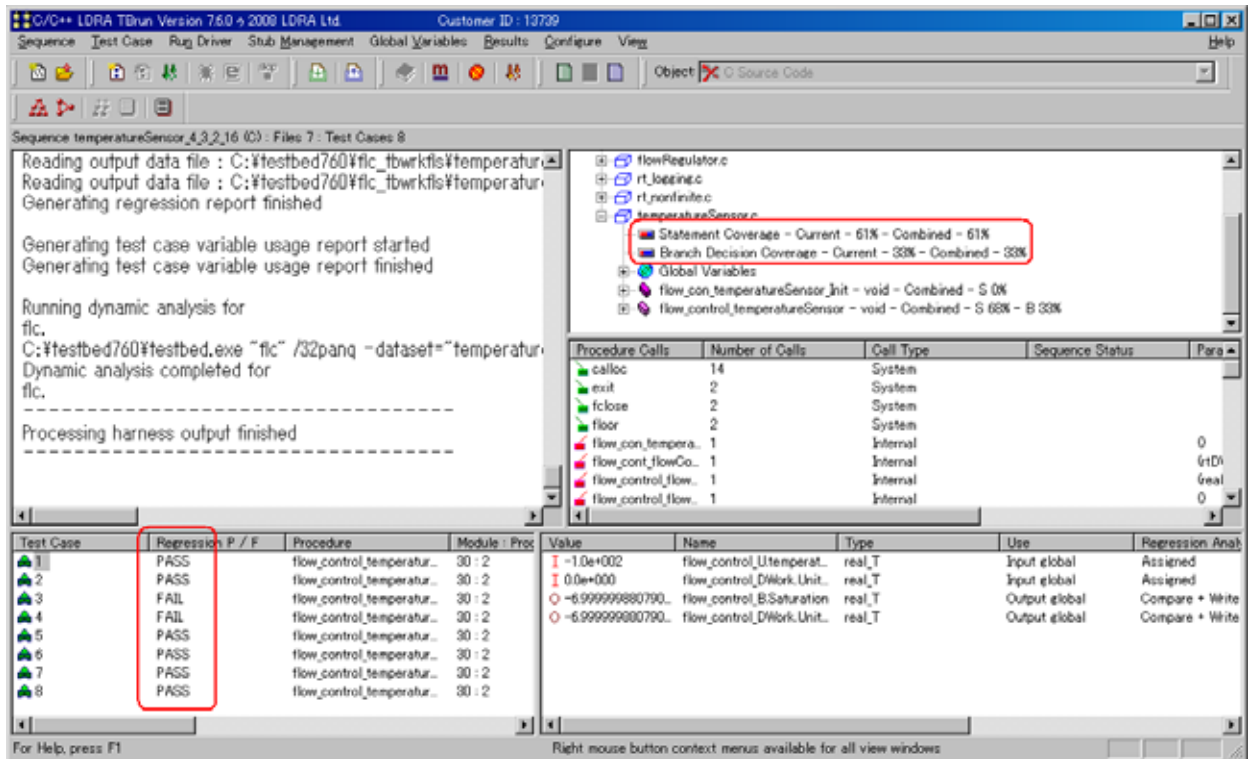
(シリアルを使って Windows CE のコンソールを起動させるバッチファイルを定義することで、テスト対象コードの自動実行もバッチ処理内に組み込み可能)

事前に設定したブレークポイントでプログラムがブレーク。



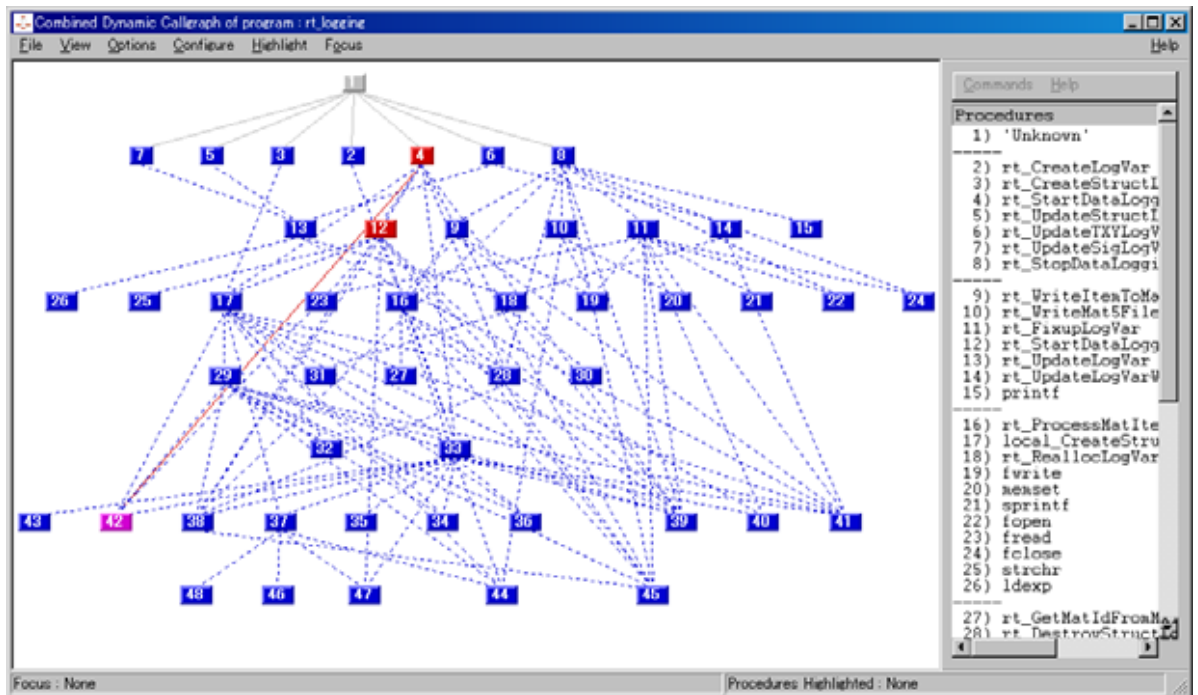
ブレーク後、タグデータ・テスト結果収集、判定・カバレッジ解析といった残りの作業もスクリプト実行されます。

9. テスト結果判定・カバレッジ解析結果を参照。



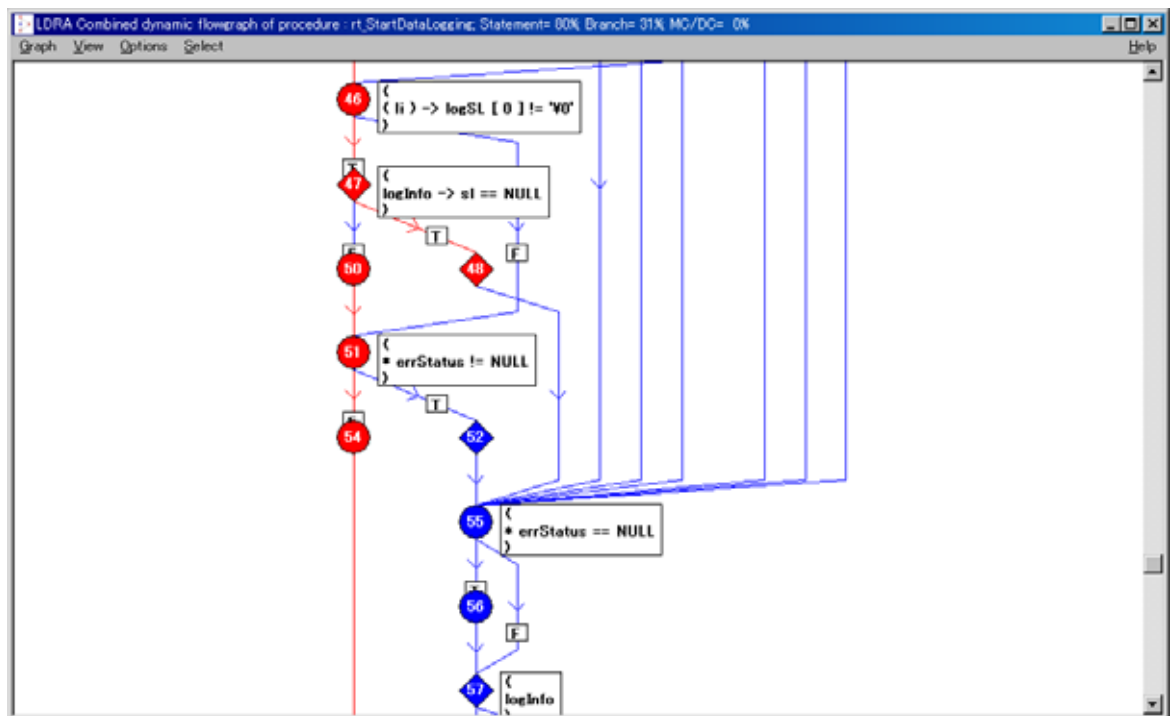
システム全体のコールグラフを表示。赤い部分が実行された関数、青い部分が未実行の関数。

[コールグラフ]



上記コールグラフの4番の関数をクリックすると、その関数内のフローグラフが表示されます。赤色が実行部分、青色が未実行部分です。分岐箇所の条件を表示することもできます。また、カバレッジ結果レポートを見ることもできます。

[フローグラフ]



フローグラフのステートメントをクリックすると、ソースコードが表示されます。実行、未実行がソース表示でも確認できます。

[ソース表示]

```
Line Block Statement
4727 51 /* outputs */
4728 51 * errStatus =
4729 51 rt_StartDataLoggingForOutput (
4730 51 li, finalTime, stepSize, errStatus );
4731 51 if
4732 51 {
4733 51 * errStatus != NULL
4734 51 }

Current Coverage : Block 51 has been executed once
Combined Coverage : Block 51 has been executed 3 times

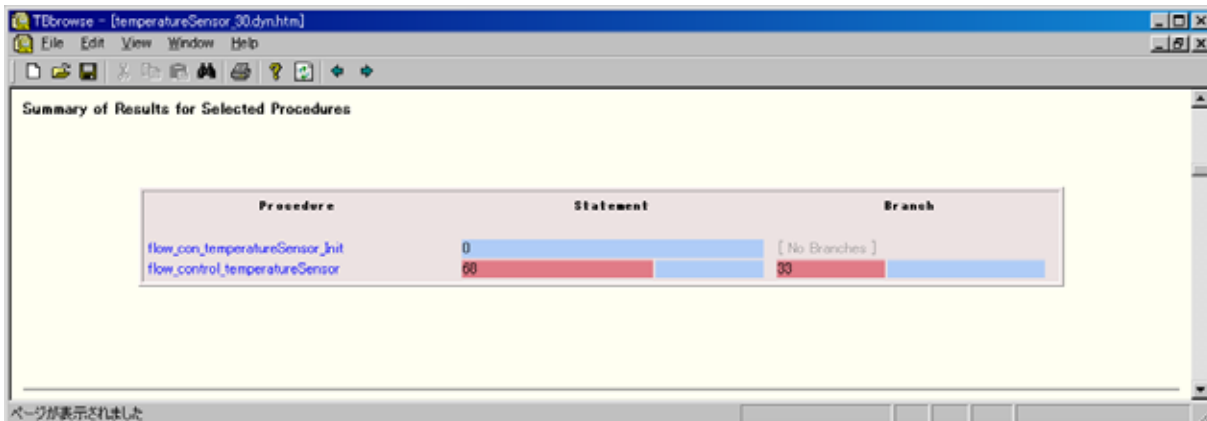
4735 52 (
4736 52 goto ERROR_EXIT ;

Current Coverage : Block 52 has not been executed
Combined Coverage : Block 52 has not been executed
```

RT_LOGGING_27 : rt_StartDataLogging : Showing blocks 51 to 52 Standards Model:
For Help, press F1

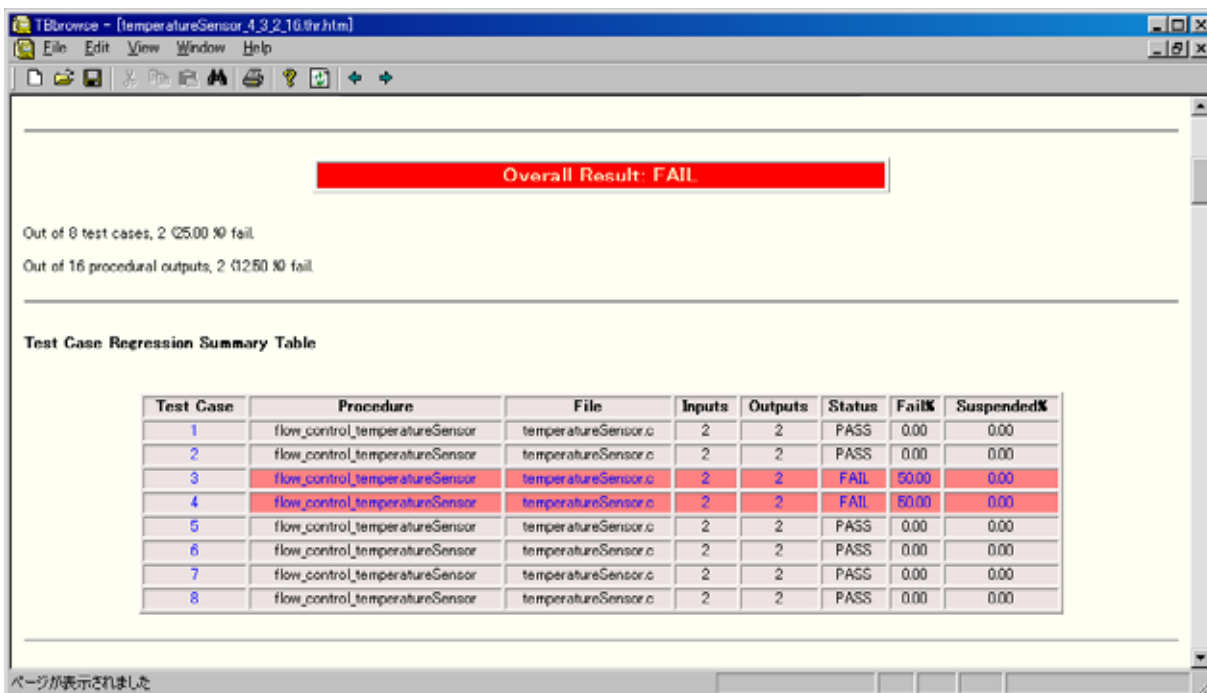
関数、ファイル単位など、テストカバレッジ結果の%表示も確認。

[カバレッジレポート]



テスト結果も確認。

[テスト結果レポート]



各テストケース、結果、の詳細を確認。

[テストケースレポート]

The screenshot shows a web browser window titled 'TBrowser - [temperatureSensor_4_3_2_10.ftr.htm]'. The main content area displays the following information:

Test Case 3 : Procedure flow_control_temperatureSensor (temperatureSensor.c) - FAIL

Number of Input Parameters : 0
Number of Input Globals : 2
Number of Output Parameters : 0
Number of Output Globals : 2

Procedure Returns : void

Input Globals

Name	Type	Value	UsrG
flow_control_U.temperature	real_T	-1.0e+002	
flow_control_DWork.UnitDelay_DSTATE_k	real_T	-3.3333333333333333e+010	

Output Globals

Name	Type	Expected Value	Actual Value	Status	UsrG
flow_control_B.Saturation	real_T	-1.0e+002	-1.000000e+002	PASS	
flow_control_DWork.UnitDelay_DSTATE_k	real_T	-1.000000007e+010	-1.000000e+010	FAIL	

以上のように LDRA Testbed/TBrun テスト自動化ツールの特徴は、あらゆるターゲットハードウェア、OSバージョン、コンパイラなどの環境に対しても設定が容易に行えることです。

富士設備、LDRA社では、このような設定に対するサービスも行っていますので、ご興味いただける場合はご相談くださると幸いです。

以上



富士設備工業株式会社 電子機器事業部

〒591-8025 大阪府堺市北区長曾根町1928-1

Tel: 072-252-2128 www.fuji-setsu.co.jp