

# Helio ボードで ARM<sup>®</sup> Streamline<sup>™</sup> を使う方法

ver.13

## Helio ボードで ARM Streamline を使う方法

### 目次

1. はじめに .....	3
2. 概要: Streamline とは? .....	3
3. 準備: 利用するための条件 .....	4
4. 手順 1: ターゲット側の起動～設定 .....	5
5. 手順 2: 開発マシン( PC )側の準備 .....	5
6. 手順 3: キャプチャと解析オプション .....	6
7. 手順 4: カウンタのコンフィギュレーション .....	7
8. 手順 5: キャプチャ開始 .....	8
9. 手順 6: キャプチャ停止 .....	8
10. キャプチャ結果の解析 .....	9
11. その他の結果表示(例) .....	10
改版履歴 .....	11

## 1. はじめに

この資料は、Helio ボード用の GSRD を使って ARM Streamline を用いた動的プロファイルを行う手順について記載します。ARM Streamline の使い方については、ARM 社発行のドキュメントにて詳述されるため、本書では画面キャプチャを多用して簡潔に判り易く使い方を解説します。

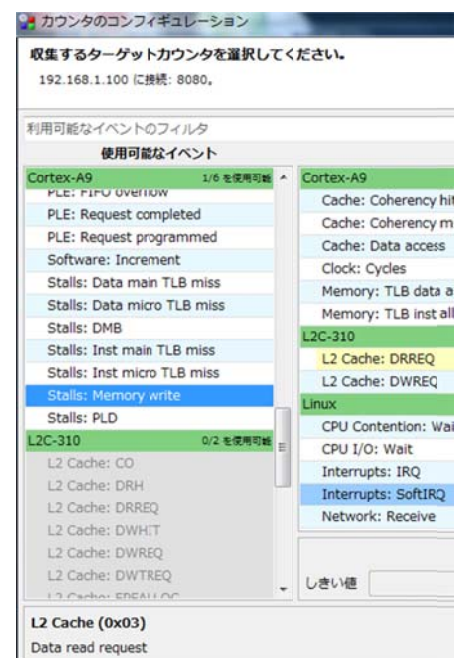
尚、本書に記載する説明内容および画面キャプチャは SoC EDS v13.1 に含まれる ARM Development Studio™5 (DS-5™) Altera Edition ( 5.15.0 ) で実施した場合の内容となります。

## 2. 概要: Streamline とは?

- CPU スケジューリング、コールトレース他、Linux ターゲットの動作情報をモニタすることができます。
- パフォーマンス問題の解析に有用です。
- 選択したイベント情報のモニタも可能です。

下記のカテゴリからイベントを選択できます。

- Cortex®-A9 :モニタイベントを 6 個 まで選択できます。  
Branch, Cache, Clock, Core, Exception, External, Instruction, Intrinsic, Java, Jazelle®, Memory, Pipeline, PLE (PreLoad Engine), Software, Stalls
- L2C-310 :モニタイベントを 2 個 まで選択できます。  
L2 Cache
- Linux :モニタイベントを個数制限なしで選択できます。  
CPU Contention, CPU I/O, Disk IO, Interrupts, Memory, Network, Scheduler



詳細は、以下のメーカー資料をご参照ください。

「ARM Streamline パフォーマンス アナライザ」

URL: <http://www.arm.com/ja/products/tools/software-tools/ds-5/streamline.php>

### 3. 準備: 利用するための条件

➤ Streamline を使うための条件

- ✓ ターゲットで Linux が動作していること(カーネル v2.6.32 以降)。
- ✓ 必要なカーネルコンフィグオプションが有効であること。  
( CONFIG\_DEBUG\_KERNEL=y 、CONFIG\_DEBUG\_INFO=y )
- ✓ gator ドライバデーモンがインストールされていること。
- ✓ Ethernet でホスト PC とネットワーク接続できること。

※RocketBoards.org で配布されている Linux Kernel 3.9 の Helio ボード用 GSRD は、Streamline が利用できる状態になっています。

ターゲット環境の設定方法は以下の ARM 社のドキュメントに記載されていますのでご参考ください。

「ARM DS-5 ARM Streamline の使用」

URL: <http://infocenter.arm.com/help/topic/com.arm.doc.dui0482ej/BABEJAAI.html>

➤ 実行環境(機器接続)

- ✓ ターゲットボードとの接続: UART、Ethernet ケーブルを接続してください。必要に応じて USB-Blaster™ II も接続してください。
- ✓ SD カードスロット: Linux の SD カードイメージが書き込まれたカードをセットしてください。

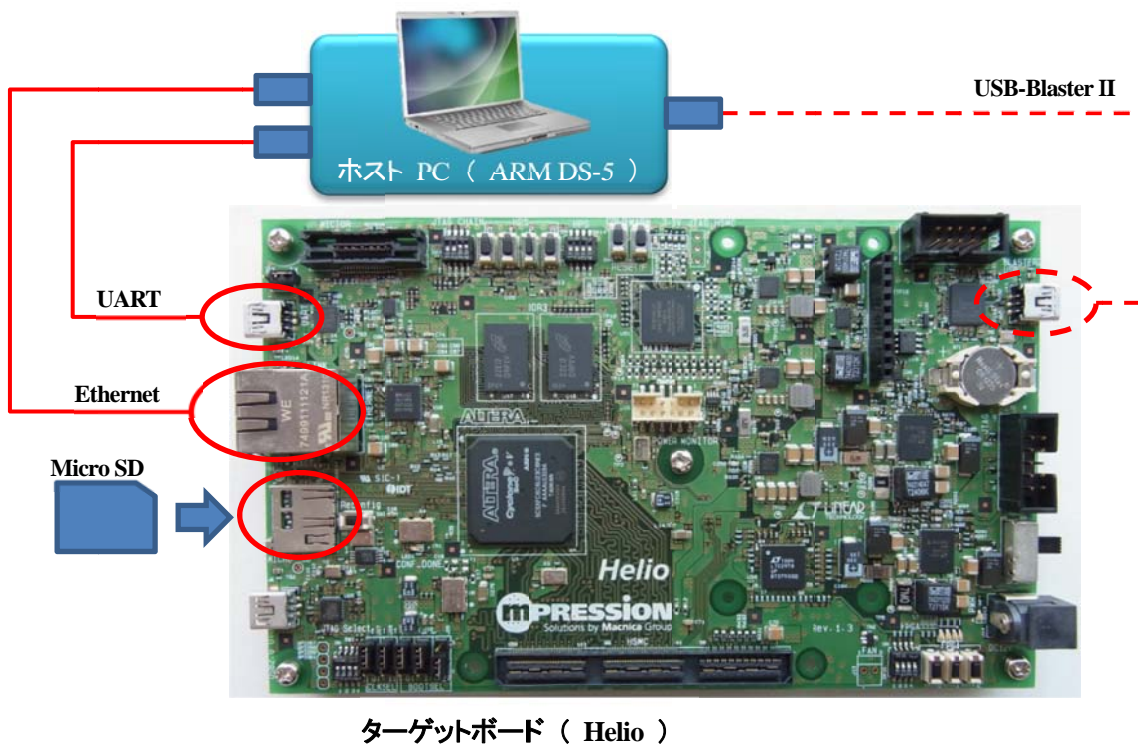


図 1 接続イメージ

#### 4. 手順 1:ターゲット側の起動～設定

1. ターゲットと PC が Ethernet および UART ケーブルで接続された状態でターゲットを起動します。

2. ターゲット上で gator ドライバ/デーモンを起動します。

ターミナルから下記のコマンドを順番に実行してください。

```
# insmod /etc/init.d/gator.ko
```

```
#/etc/init.d/gatord &
```

3. ターゲットの IP アドレスを確認し、必要に応じて固定 IP アドレスを設定します。

ターミナルから下記のコマンドを実行すると IP アドレスを確認できます。

```
# ifconfig
```

ターミナルから下記のコマンドを実行すると IP アドレスを変更できます。

```
# ifconfig eth0 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255 up
```

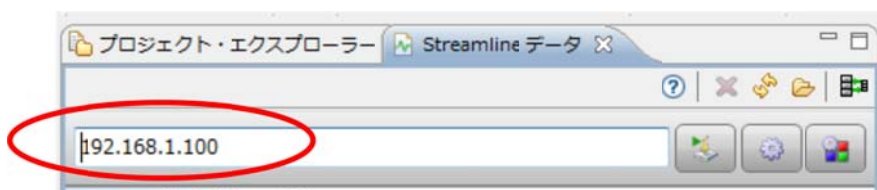
#### 5. 手順 2:開発マシン( PC )側の準備

1. DS-5 を起動して、Streamline データ ビューを表示します。

メニュー: [ウィンドウ(W)] > [ビューの表示(V)] > [その他(O)...] から

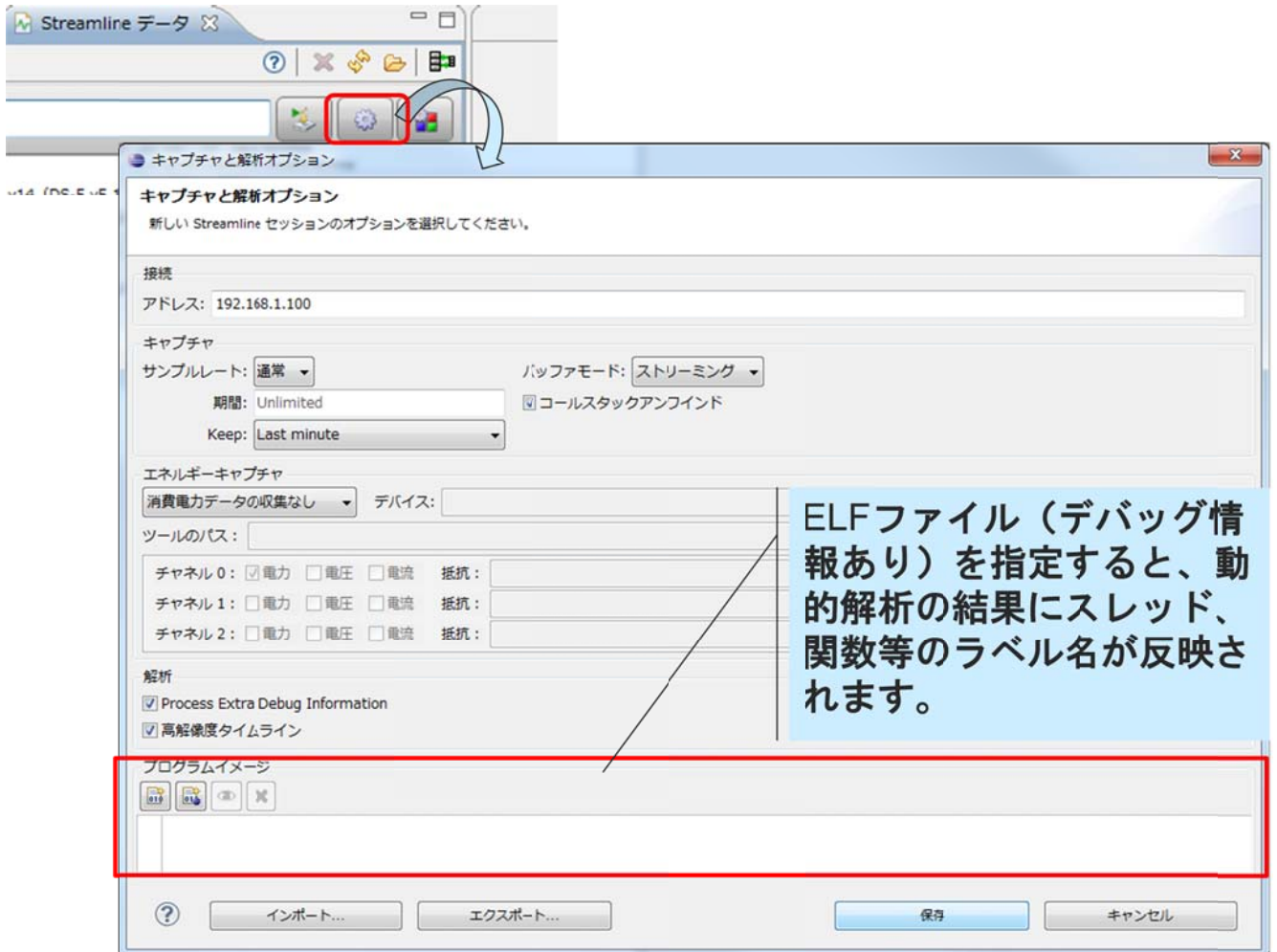
ビューの表示ウィンドウ上で [ARM Streamline] > Streamline データ を選択してください。

2. Streamline データ ビューにターゲットの IP アドレスもしくはマシン名を入力してください。



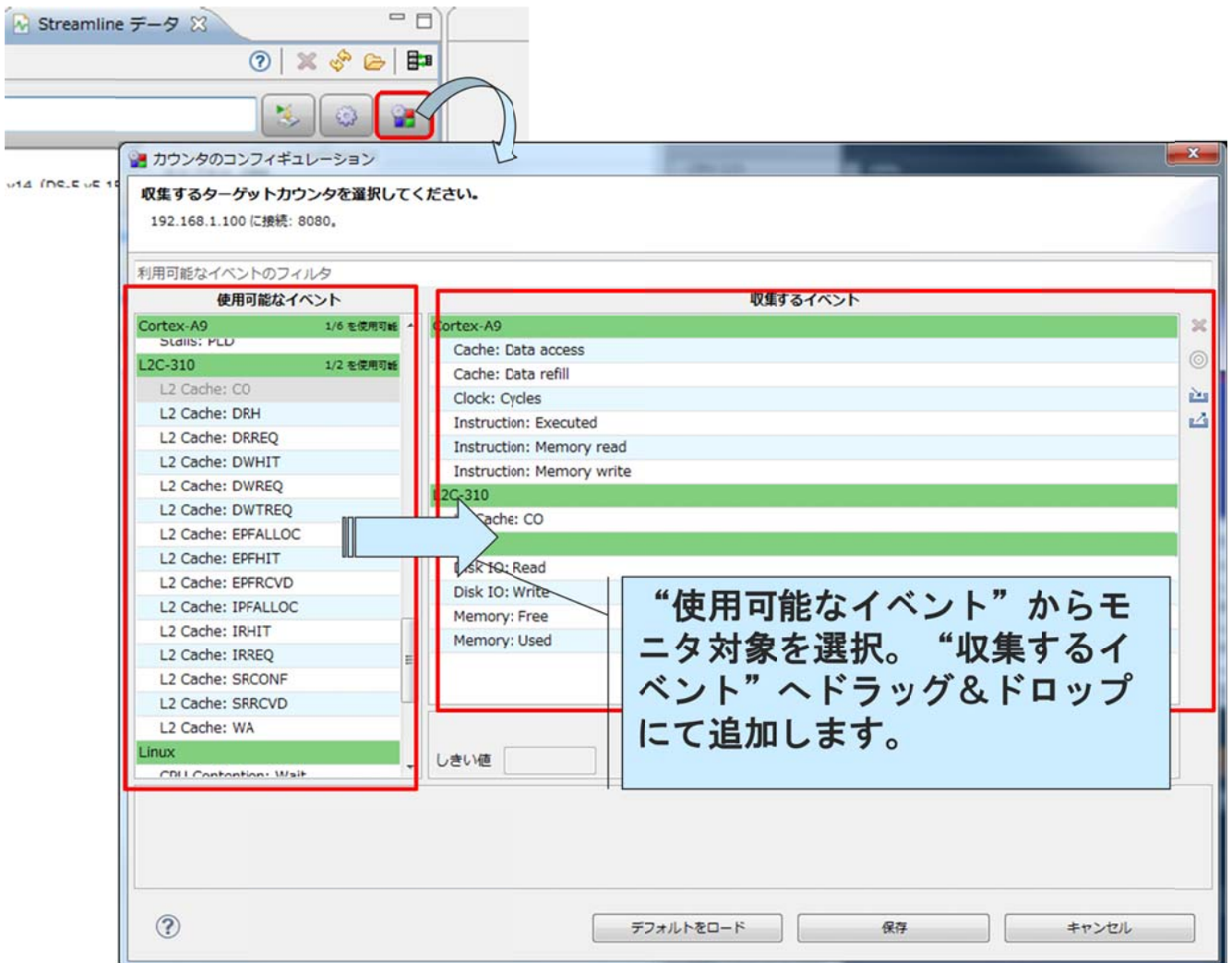
## 6. 手順 3: キャプチャと解析オプション

プログラム情報をモニタしたい場合はプログラムイメージ欄に ELF ファイルを指定します。



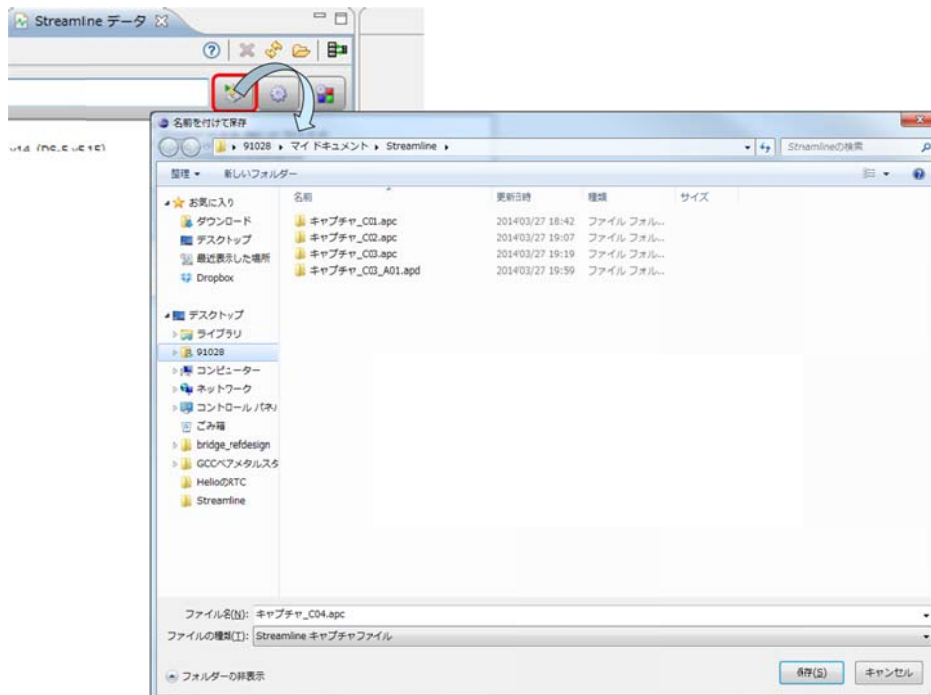
## 7. 手順 4: カウンタのコンフィギュレーション

モニタしたいイベント情報を選択します。

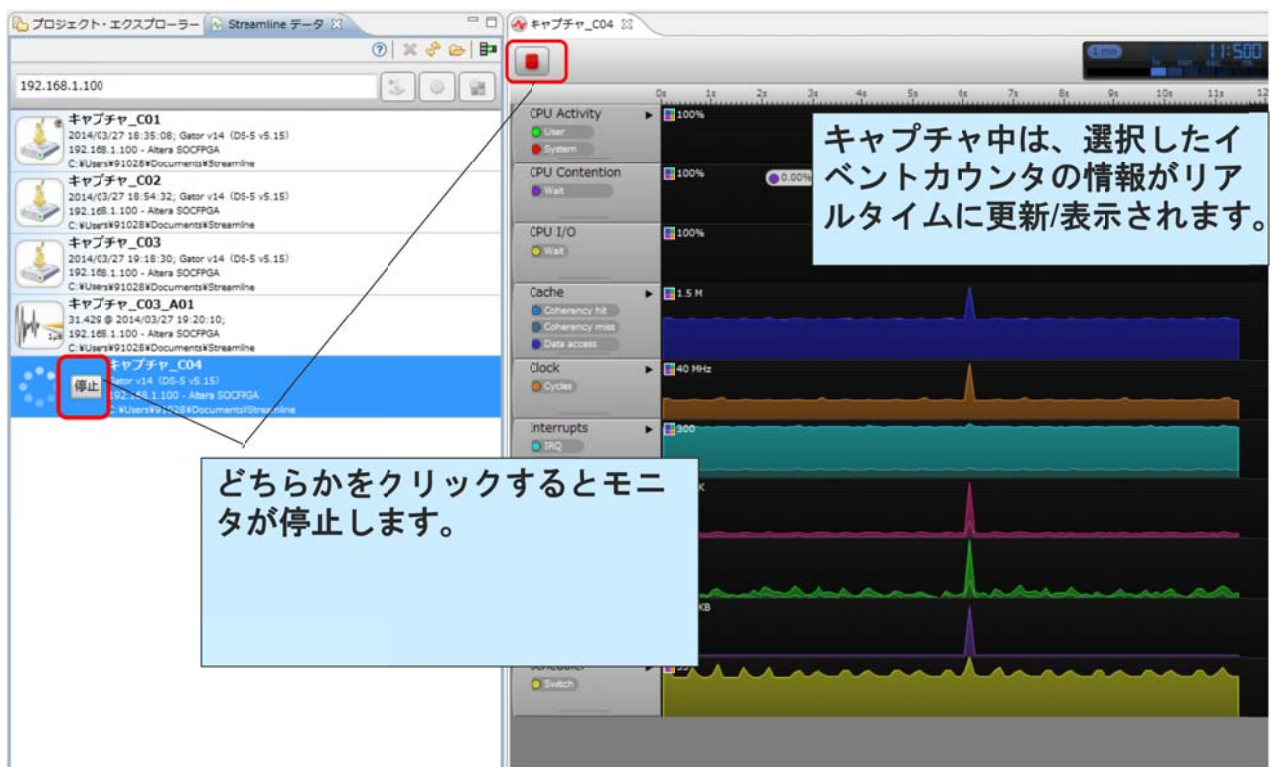


## 8. 手順 5: キャプチャ開始

キャプチャ開始ボタンをクリックして、ファイル名を指定するとモニタを開始します。



## 9. 手順 6: キャプチャ停止





## 10. キャプチャ結果の解析

自動的に収集したトレースデータの分析処理が動作します。

分析終了 ( 100% ) になったデータを選択するとプログラムのトレース情報が参照できます。

The screenshot displays the Streamline software interface. On the left, a 'プロジェクト・エクスプローラー' (Project Explorer) window shows a list of capture results. The entry 'キャプチャ\_C04\_A01' is highlighted with a blue selection bar and a red box. A red arrow points from this entry to the main analysis window. The main window shows a detailed performance analysis graph with multiple tracks: CPU Activity, L2 Cache, Cache, Memory, Lock, Interrupts, Network, and DMC. The graph shows various performance metrics over time, with a 'サンプル' (Sample) window open at the bottom right.

## 11. その他の結果表示(例)

画面左上のタブ・ボタンを操作する事で色々な分析結果が確認できます。

The screenshot displays the ALTIMA software interface. At the top, there are navigation tabs: タイムライン (Timeline), コールバス (Call Bus), 関数 (Functions), コード (Code), コールグラフ (Call Graph), スタック (Stack), and ログ (Log). Below these is a CPU Activity graph showing 100% activity for the 'User' process. The main window is divided into several panes:

- Left Pane:** Lists capture sessions (キャプチャ) for '192.168.1.100'. It shows sessions for 'キャプチャ\_C01', 'キャプチャ\_C02', 'キャプチャ\_C03', and 'キャプチャ\_C03\_A01'. The 'キャプチャ\_C03\_A01' session is selected.
- Top Middle Pane:** A table showing process statistics. The columns are: セルフ (Self), セルフの割合 (Self Ratio), プロセス (Process), % プロセス (% Process), 合計 (Total), スタック (Stack), [プロセス](スレッド)/コード ([Process](Thread)/Code), and ログセッション (Log Session).
- Bottom Middle Pane:** A detailed view of the selected process, showing its call stack and associated code.

セルフ	セルフの割合	プロセス	% プロセス	合計	スタック	[プロセス](スレッド)/コード	ログセッション
62,681	100.00%	[idle]	100.00%	99.72%	0	[idle]	-
62,681	100.00%	{swapper/1 #0}	100.00%	99.72%	0	{swapper/1 #0}	-
126	100.00%	{-unknown code in kernel-}	100.00%	0.20%	0	{-unknown code in kernel-}	<unknown>
126	100.00%	{busybox #11285}	100.00%	0.20%	0	{busybox #11285}	-
112	88.89%	{top #11285}	88.89%	0.18%	0	{top #11285}	-
10	7.94%	{-unknown code in libc-2.15.so-}	7.94%	0.02%	0	{-unknown code in libc-2.15.so-}	<unknown>
4	3.17%	{-unknown code in busybox-}	3.17%	< 0.01%	0	{-unknown code in busybox-}	<unknown>
25	100.00%	{scroll_server #660}	100.00%	0.04%	0	{scroll_server #660}	-
25	100.00%	{scroll_server #660}	100.00%	0.04%	0	{scroll_server #660}	-
21	84.00%	{-unknown code in kernel-}	84.00%	0.03%	0	{-unknown code in kernel-}	<unknown>
4	16.00%	{-unknown code in libc-2.15.so-}	16.00%	< 0.01%	0	{-unknown code in libc-2.15.so-}	<unknown>
19	100.00%	{gattord #12635}	100.00%	0.03%	0	{gattord #12635}	-
16	84.21%	{gattord-sender #12637}	84.21%	0.03%	0	{gattord-sender #12637}	-
16	84.21%	{-unknown code in kernel-}	84.21%	0.03%	0	{-unknown code in kernel-}	<unknown>
2	10.53%	{gattord-stopper #12636}	10.53%	< 0.01%	0	{gattord-stopper #12636}	-
2	10.53%	{-unknown code in kernel-}	10.53%	< 0.01%	0	{-unknown code in kernel-}	<unknown>
1	5.26%	{gattord-child #12635}	5.26%	< 0.01%	0	{gattord-child #12635}	-
1	5.26%	{-unknown code in kernel-}	5.26%	< 0.01%	0	{-unknown code in kernel-}	<unknown>
7	100.00%	{kernel}	100.00%	0.01%	0	{kernel}	-
3	42.86%	{kworker/u:0 #6}	42.86%	< 0.01%	0	{kworker/u:0 #6}	-
3	42.86%	{-unknown code in kernel-}	42.86%	< 0.01%	0	{-unknown code in kernel-}	<unknown>
1	14.29%	{kworker/u:0 #12655}	14.29%	< 0.01%	0	{kworker/u:0 #12655}	-
1	14.29%	{-unknown code in kernel-}	14.29%	< 0.01%	0	{-unknown code in kernel-}	<unknown>
1	14.29%	{kworker/u:0 #12675}	14.29%	< 0.01%	0	{kworker/u:0 #12675}	-
1	14.29%	{-unknown code in kernel-}	14.29%	< 0.01%	0	{-unknown code in kernel-}	<unknown>
1	14.29%	{kworker/u:0 #12684}	14.29%	< 0.01%	0	{kworker/u:0 #12684}	-
1	14.29%	{-unknown code in kernel-}	14.29%	< 0.01%	0	{-unknown code in kernel-}	<unknown>

上の 1 つのコールバスのリンクをクリックしてください。そのリンクとその子リンクの関数呼び出し情報が表示されます。

## 改版履歴

Revision	年月	概要
1	2014 年 7 月	新規作成

### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。  
 株式会社アルティマ ホームページ: <http://www.altima.co.jp>      技術情報サイト EDISON: <https://www.altima.jp/members/index.cfm>  
 株式会社エルセナ ホームページ: <http://www.elsena.co.jp>      技術情報サイト ETS : <https://www.elsena.co.jp/elspear/members/index.cfm>
4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカー発行の英語版の資料もあわせてご利用ください。