

SOLID IDEのイベントトラッカ機能を 使ってみよう!

2019.07.08 京都マイクロコンピュータ

SOLIDイベントトラッカ機能



1

イベントトラッカの概要

 タスク、割り込み、タイマハンドラの実行時間や遷移を、この 画面のように時間軸に沿って記録・表示する機能です。





イベントトラッカの概要

- タスクや割り込み間の遷移が設計通りになっているか?
- ・タスクや割り込みの処理時間が想定通りになっているか?
- ・タスクや割り込みの処理間隔が想定通りになっているか?

等の確認や、

・実際の処理時間のざっくりした評価

等に使える機能です。



SOLIDイベントトラッカを 使うための準備

使うために前もって設定すべき内容

enjoy **D**evelopment

イベントトラッカを使うための準備

- 既にイベントトラッカが使用するタイマーの実装が済んでいる
 ことを前提として説明します。
- タイマー以外の下準備は必要ありません。

※新しい環境でタイマーを実装する必要が有る場合は、本資料末 尾の付録を参照して実装してください。



SOLIDイベントトラッカの 使い方



6

イベントトラッカの使い方

- ソリューションのプロパティファイル(.props)の
 <LibEventTracker> というプロパティを "true" に設定。
 無効にする場合には "false"に変更します
- ・プロパティ<LibEventTracker>が 無い場合には巻末の付録の 説明に従い追加して下さい

※変更後は、一旦 SOLID-IDE を 終了し、ソリューションを 読み直す必要があります。 [~~.propsの変更内容]

```
<PropertyGroup Label="UserMacros">
```

<LibEventTracker>true</LibEventTracker>

</PropertyGroup>

※※スターターキット(BSP 2.0.1 以降必須)では常時有効なので、この設定は不要です。

イベントトラッカの使い方

- 1. 全ページの設定をしてビルド&実行
- 2. 適当なタイミングで、[デバッグ]-[すべて中断]

🛟 DB51903PF-demo-AXELL-usecase (実行中) - SOLID-IDE				
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B)	デル	ッグ(D) ツール(T) KMC(SOLID_V7A_A	RM) ウィンドウ(W)	ヘルプ(H)
🦉 G.〇 階・ 🗇 - 🔄 💾 🔐 米 🗗 台 ウ・		ウィンドウ(W)	•	- Debu
プロセス [10732] demo-AXELL-usecase.out 🔹 💽 ライフサイ	\gg	続行(C)	F5	🏹 🚧 スタック フレー。
	н	すべて中断(K)	Ctrl+Alt+Break	* I X
7574	-	デバッグの停止(E)	Shift+F5	
デバッガーの実行中は利用できません。	×	すべてデタッチ(D)		
		すべて中止(M)		
	ð	再起動(R)	Ctrl+Shift+F5	
	e. B	プロセスにアタッチ(P)		-
		その他のテバックターゲット	•	
		例外(X)	Ctrl+Alt+E	
	ς.	ステップ イン(I)	F11	
	Ċ,	ステップ オーバー(O)	F10	
	G	ステップ アウト(T)	Shift+F11	
	⇔	クイック ウォッチ(Q)	Shift+F9	
		ブレークポイントの設定/解除(G)	F9	
		ブレークポイントの作成(B)	•	
	8 9	すべてのブレークポイントの削除(D)	Ctrl+Shift+F9	
		すべてのデータヒントをクリア(A)		
		データヒントのエクスポート(X)		
		データヒントのインポート(I)		8

イベントトラッカの使い方

3. メニューの[デバッグ]-[ウインドウ]-[イベントトラッカ]を選択

	~					
۲ ^۲ (B)	デバ	ッグ(D) ツール(T) KMC(SOLID_V7A_AF	RM) ウィンドウ(W)	୍ଧା	_/ プ(H)	
5.		ウィンドウ(W)	•	•	ブレークポイント(B)	Alt+F9
イフサイ	•	続行(C)	F5	Ľ	デバッグ 例外	
	Ш	すべて中断(K)	Ctrl+Alt+Break	K	出力(O)	
	•	デバッグの停止(E)	Shift+F5	Ô	タスク(S)	Ctrl+Shift+D, K
>> >>	Х	すべてデタッチ(D)		(≣	並列スタック(K)	Ctrl+Shift+D, S
22 <u>22</u>		すべて中止(M)			並列ウォッチ(R)	· · ·
·· ··	ð	再起動(R)	Ctrl+Shift+F5		ウォッチ(W)	•
22 22 11 11	¢ [©]	プロセスにアタッチ(P)		F	自動変数(A)	Ctrl+Alt+V, A
		その他のデバッグターゲット	•	[*;]	ローカル(L)	Alt+4
		例外(X)	Ctrl+Alt+E	Þ	イミディエイト(I)	Ctrl+Alt+I
MODI	ς.	ステップ イン(1)	F11	>	PARTNER コマンド ウィンドウ	
	C,	ステップ オーバー(O)	F10	Œ	呼び出し履歴(C)	Alt+7
EL_MOI	G	ステップ アウト(T)	Shift+F11	F	コードカバレッジ	
	⇔	クイック ウォッチ(Q)	Shift+F9	▣	קב	
		ブレークポイントの設定/解除(G)	F9	<i>70</i> .	スレッド(H)	Ctrl+Alt+H
		ブレークポイントの作成(B)	•	E	モジュール(O)	Ctrl+Alt+U
÷.	ک و	すべてのブレークポイントの削除(D)	Ctrl+Shift+F9	e [®]	プロセス(P)	Ctrl+Shift+Alt+P
(すべてのデータヒントをクリア(A)		Q∎	RTOS ビューアー	
		データヒントのエクスポート(X)		1	イベント トラッカー	X
		データヒントのインポート(I)			メモリ(M)	· ·
スタ		名前を付けてダンプを保存(V)		5	逆アセンブル(D)	Alt+8
を呼び		オプションと設定(G)		08	レジスタ(G)	Alt+5
	ų	demo-AXELL-usecase のプロパティ				kemer_impi.c

[DATA RANGE]	-300	-200	-100	0	100	200	30
[VIEW RANGE] S: -316 E: 316				 			
[DATA INFO]							
RTOS La-7-	- 2)						
	これ	がイハ 表示	ベント 同面	トラッです	ッカの		

イベントトラッカの使い方

4. イベントトラッカの画面にマウスカーソルを移動し、右ク リックで表示されたメニューから[DLLのロード(D)]を選ぶ

イベントトラッカー 😐 🗙					core_support.S @	* × •
[DATA RANGE]	-300 -200	-100	0	100	200	300
[VIEW RANGE] S: -316 E: 316			<u> </u>			
[DATA INFO]	-88		DLLのロード(D) データのロード(L) データのセーブ(S) 解析(P) 検索(F) プロファイル(P) モジュールの編集(M)			

イベントトラッカの使い方

5. EvtFilterSOLID_TOPPERS.dll を選ぶ (所在はP.13)



選択後 OK をクリック。

開く)
← → × ↑ 📙 <u>C:¥P</u>	rogramData¥KMC¥SOLID¥tools¥eventtracker	~ Ū	eventtracker	の検索	م ر
整理 ▼ 新しいフォルダ-					
👝 OneDrive	名前 ^	∨ 更新日	诗	種類	
DeskTop	EvtFilterSOLID_TOPPERS.dll	2018/04	4/13 15:55	アプリケーショ	ョン拡張
DisneyGames					
PHOTO SHARE					
お気に入り					
ドキュメント					
モバイルの写真					
,画像					
共有のお気に入り					
公開					
自宅に送信					
📙 電子メールの添付 🗸	<				
ファイ	ル名(N): EvtEilterSQLID TOPPERS dll	~	DLL (*.dll)		~
		-			
			開く(O)	4	Fヤンセル

TOPPERS/FMP の場合はコア数指定必要

-		···· K.		
	DLL: C:¥DriveD¥	MASTER¥tools¥eventtracker¥dII¥I 👻 🛄		
	CPU: エンディアン:	ARM	DLL を選択した後、	
	コア数:	1 1 2 3 4 5	[コア数]のリストボックスから SOLID-OSを実行しているコアの 数を選択してから、	ら、 の
			[OK] を押します	



イベントトラッカの使い方

EvtFilterSOLID_TOPPERS.dll の所在

StarterKitの環境
 C:¥ProgramData¥KMC¥SOLID¥tools¥eventtracker¥

・製品版環境 (インストール先フォルダ)¥tools¥eventtracker¥dll¥

イベントトラッカの使い方

6. イベントトラッカの画面にマウスカーソルを移動し、右ク リックで表示されたメニューから[解析(P)]を選ぶ

イベントトラッカー 👳 🗙				core_support.S	≝ × -	<u> </u>	· · · · · · · · · · · · · · · · · · ·
[DATA RANGE] S: O F: O	30us 20us	Oms 10us	10us	20us	30us		×
[VIEW RANGE] S: -316 E: 316		 		1	.0usi K	Start address:	val(_kmc_toppers_event_data) -
CPU [DATA INFO]	-7.3us	DLLのロード(D) データのロード(L) データのセーブ(S) 解析(P) 検索(F) プロファイル(P) モジュールの編集(N イベントの編集(E)	A)			Size 範囲指定ダ では[OK]を/	val(_kmc_toppers_event_data_size) ・ OK ++ンセル イアログ フリック
		モジュールを隠す(H)		•		

イベントトラッカの結果を表示

機能が豊富にあるのですが、ここでは基本的な操作のみ説明します。



イベントトラッカの結果を表示(FMP)

機能が豊富にあるのですが、ここでは基本的な操作のみ説明し



イベントトラッカの結果を表示

•マウスのホイールを回すと、時間軸が Zoomin, Zoomout します。



イベントトラッカの結果を表示

タイムバーを表示し、経過時間を計測できます







タイムライン上の線と四角の色の意味

状態
CPU実行状態 (Running) ※FMPの場合 Core0実行状態
CPU実行権獲得可能 (Ready)
待ち状態 (Wait, Suspend, Suspend-wait)
FMPのみ Core1 実行状態 (Running)
FMPのみ Core2 実行状態 (Running)
FMPのみ Core3 実行状態 (Running)

四角の色	イベントの意味
青四角	待ち状態イベント (Wait に遷移)
赤四角	強制待ちイベント (Suspend に遷移)
紫四角	二重待ちイベント (Suspend-wait に遷移)
赤四角	メールボックス送信イベント



メモリ使用量、処理時間 オーバーヘッド

enjoy **D**evelopment

イベントトラッカのオーバーヘッド

- ・メモリ使用量
 - .data セクションに (24byte * 最大イベント数) + 32byte の領域を 使用します。 注意: 最大イベント数は 2のn乗の数字である必要があります
- 処理時間オーバヘッド イベントを記録するための処理量は、全体の処理量と比較して 十分に少ない(ディスパッチが起きる回数は十分少ない)ので、 処理時間のオーバーヘッドは無視できます。





必要なタイマの実装

・システム横断で一意なカウンタ値が読める64bit長タイマを選択し、以下の2つの関数を実装する必要があります。

```
#include "impl_globaltick.h"
void IMPL_GLOBAL_TICK_Init(IMPL_GLOBAL_TICK_INFO *pInfo);
ί
      // 最低限必要な処理
      pInfo->tickspersec = (64ビットカウンタが1秒にカウントするカウント数);
      // 64bitカウンタの初期設定・起動処理が必要なら追加
}
unsigned long long IMPL GLOBAL TICK GetCurrent()
Ł
      return (64bitカウンタの現在の値);
}
```



記録するイベントの個数の変更

記録イベント数の変更は、bsp プロジェクトの impl_evttrk.cの
 以下のマクロを変更してください

#define MAX_EVENT_NUM 0x400

 このマクロで保存できるイベント数を指定しますが、処理効率 化の為、MAX_EVENT_NUMの値は、必ず20n乗の数にして下 さい。守らない場合は、イベントが正しく記録されません

.propsに <LibEventTracker> が無い場合

前述のエントリを一行追加します。

```
...
<PropertyGroup Label="UserMacros">
...
<LibEventTracker>true</LibEventTracker>
...
</PropertyGroup>
...
```

•加えて、以下の変更も追加します。

```
...
<ItemGroup>
```

<BuildMacro Include="LibEventTracker"> <Value>\$(LibEventTracker)</Value> </BuildMacro>

```
. . .
</ItemGroup>
```

• • •

以上です





日付	更新内容
2019.07.08	 FMP向け記述の追加 制限事項や設定内容について記述を追加 現状の実装とマッチしない記述内容を修正
2018.06.08	 初期リリース