

# SystemView

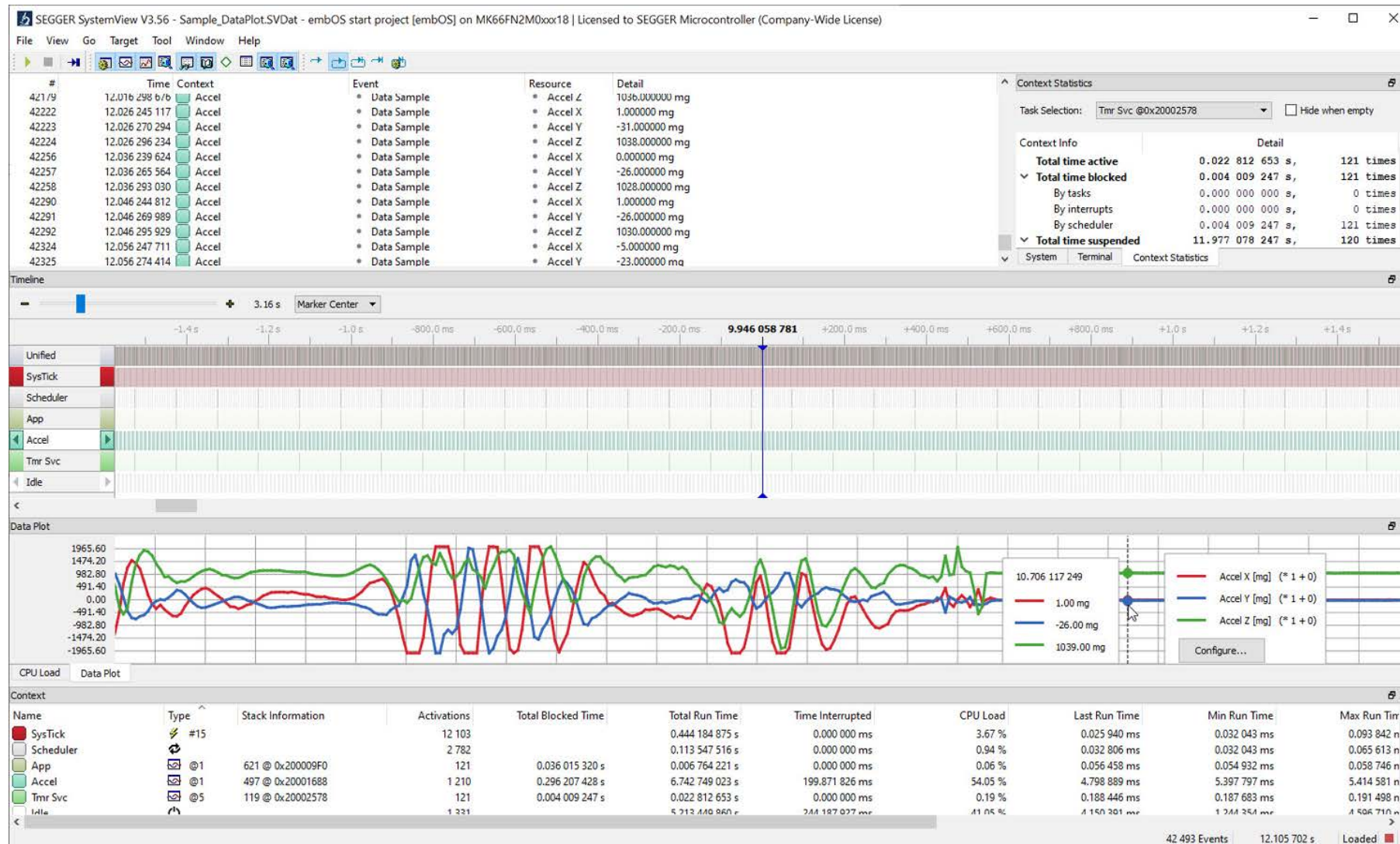
アプリケーション可視化・記録ツール



WWAZ	▲	111.52	▲	-59.31
TYRZ	▲	50.21	▲	-57.88
TTAW	▲	47.34	▲	-57.54
CCAD	▲	42.04	▲	-52.68
HAEW	▲	14.68	▼	-52.14
JAS	▲	14.68	▲	-51.14
RRAP	▲	17.74	▼	-59.26

## DataPlot ウィンドウを追加

## カスタムデータサンプルをタイムラインに合わせて記録・可視化

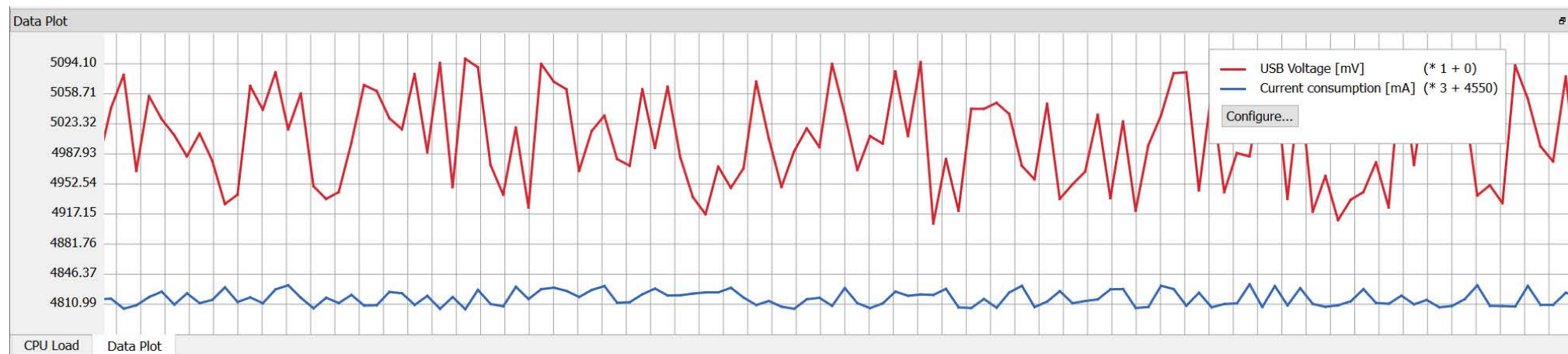


新たに追加されたデータ記録表示機能「DataPlot ウィンドウ」では、実行イベントと合わせて変数データを記録、表示が可能になりました。SystemViewを利用して記録されたデータの詳細な分析が容易になります。



## DataPlot ウィンドウを利用し、消費電力の監視

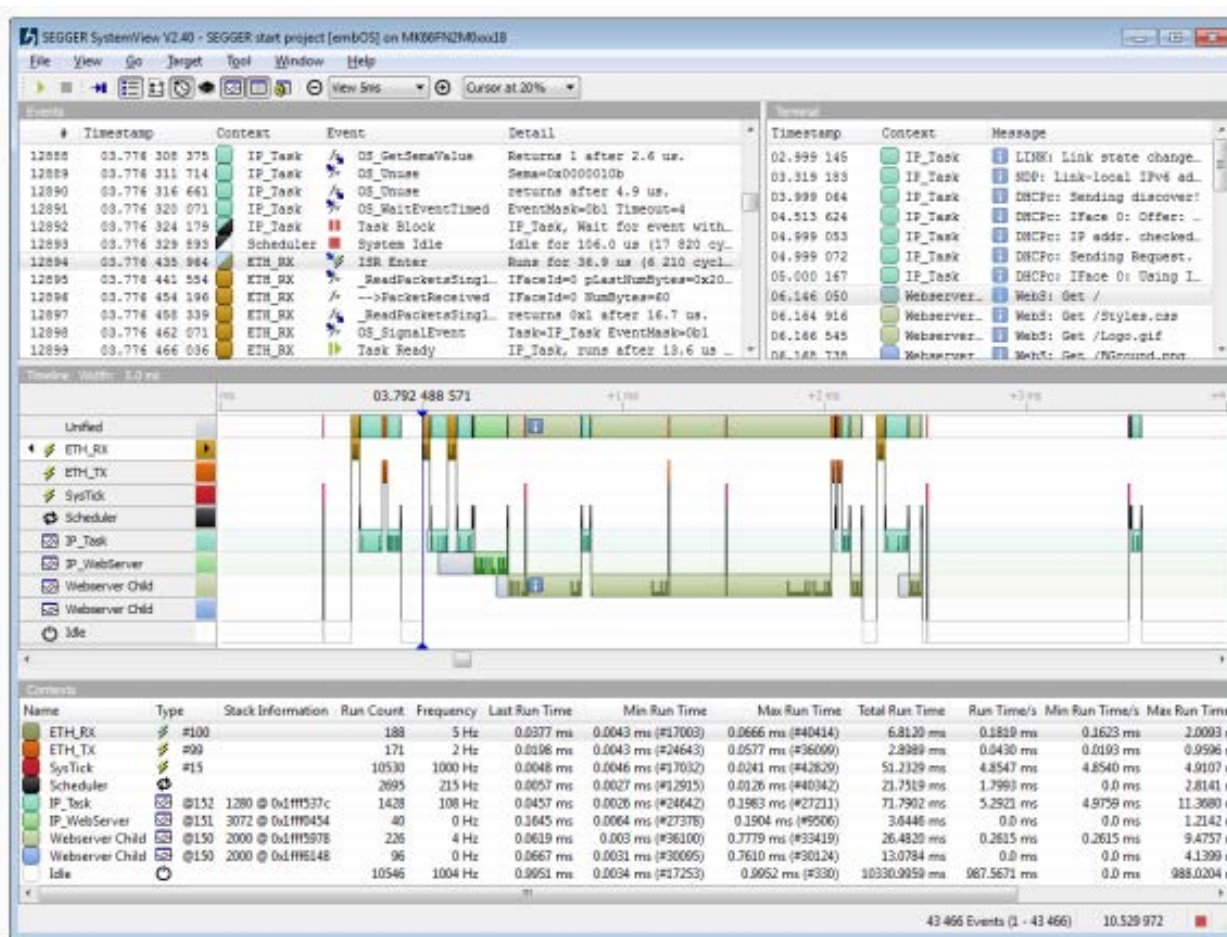
測定された電力サンプルをリアルタイムで記録することにより、DataPlot ウィンドウ内でシームレスに視覚化。記録したデータをランタイム情報と同期することにより、開発者は特定の消費電力急増や意図しない電力使用量の増加を引き起こしたアプリケーションの機能の部分を特定できます。



#	Time	Context	Event	Resource	Detail
60143	4.294 042 893	Sampling Task	• Data Sample	• Current consumption	93.288002 mA
60156	4.295 035 143	Sampling Task	• Data Sample	• USB Voltage	5048.000000 mv
60157	4.295 042 964	Sampling Task	• Data Sample	• Current consumption	85.207001 mA
60170	4.296 035 071	Sampling Task	• Data Sample	• USB Voltage	4914.000000 mv
60171	4.296 042 857	Sampling Task	• Data Sample	• Current consumption	94.222000 mA
60184	4.297 035 179	Sampling Task	• Data Sample	• USB Voltage	4928.000000 mv
60185	4.297 042 964	Sampling Task	• Data Sample	• Current consumption	86.847000 mA
60198	4.298 035 250	Sampling Task	• Data Sample	• USB Voltage	4985.000000 mv
60199	4.298 043 036	Sampling Task	• Data Sample	• Current consumption	86.607002 mA
60212	4.299 035 286	Sampling Task	• Data Sample	• USB Voltage	4986.000000 mv
60213	4.299 043 107	Sampling Task	• Data Sample	• Current consumption	86.665001 mA
60226	4.300 035 107	Sampling Task	• Data Sample	• USB Voltage	5084.000000 mv
60227	4.300 042 893	Sampling Task	• Data Sample	• Current consumption	92.596001 mA
60240	4.301 035 107	Sampling Task	• Data Sample	• USB Voltage	5053.000000 mv
60241	4.301 042 857	Sampling Task	• Data Sample	• Current consumption	86.511002 mA

特定箇所を選択、任意の時点でのタイムスタンプ情報を含んでサンプリングされたデータの値を表示できます。開発者はサンプリングデータを個別に分析できるだけでなく、システム全体のコンテキストに配置し、表示することも可能。

## 組込ソフトウェア開発者向けのリアルタイムシステム分析ツール



## 製品概要

アプリケーションの実行時の状態を表示し、開発者は、デバッガから出力されるアプリケーションの状態をより詳細に確認できます。

SystemViewを使うことにより、アプリケーションが設計通りに動作し、非効率な動きをしていないか、意図しないリソース競合などを起こしていないか、などを確認することができます。実行中のターゲットボードからリアルタイムに記録し、取得したデータは各種分析画面で視覚化されます。

フル機能対応

CPUコア : Cortex-Mx/RX  
RTOS : SEGGER embOS / FreeRTOS

限定機能対応

CPUコア : すべて  
RTOS: すべてのRTOS/non-RTOSアプリケーション

## 組込ソフトウェア開発者向けのリアルタイムシステム分析ツール



J-Linkシリーズと合わせて利用頂くことで  
ソフトウェア開発の様々な課題に対応できます。



Visualize

アプリケーション可視化

組込みアプリケーションの割  
込み、タスク、発生時間など  
を可視化



Timeline

実行時間の可視化

アプリケーションタスク  
の実行時間、CPU負荷を  
確認し設計意図通りの  
動作をしているか確認



Analysis

特定箇所の分析

マーキングしたポイント  
を詳細に分析、実行時間  
実行回数などを確認



Export

ログデータの出力

エラー、メッセージログなど  
を記録。Printf()に似た文字  
列フォーマットをサポート





アプリケーションから取得したデータを元に生成された時間、発生したタスク、割込、発生したイベント、イベントの詳細情報を表示します。

#	Timestamp	Context	Event	Detail
12893	03.792 488 571	ETH_RX	ISR Enter	Runs for 36.964 us
12894	03.792 494 161	ETH_RX	/s _ReadPacketsSingleF	IFaceId=0 pLastNumBytes=0x2002FF80 InInt=1
12895	03.792 506 804	ETH_RX	/s -->EthPacketReceived	IFace=0, 60 bytes @0x00000001, MAC:...161f->1, Data: 88
12896	03.792 510 946	ETH_RX	/s _ReadPacketsSingleF	returns 0x1 after 16.786 us
12897	03.792 514 679	ETH_RX	/s OS_TASKEVENT_Set	pTask=IP_Task Event=0000'0001
12898	03.792 518 643	ETH_RX	Task Ready	IP_Task, runs after 13.643 us
12899	03.792 521 946	ETH_RX	/s OS_TASKEVENT_Set	Returns void after 7.268 us
12900	03.792 525 536	ETH_RX	ISR Exit	Returns to Scheduler
12901	03.792 532 286	IP_Task	Task Run	Runs for 82.036 us
12902	03.792 536 268	IP_Task	/s OS_TASKEVENT_GetTimed	Returns after 163.589 us
12903	03.792 540 893	IP_Task	/s OS_MUTEX_LockBlocked	pMutex=0x1FFF042C
12904	03.792 543 696	IP_Task	/s OS_MUTEX_LockBlocked	Returns 1 after 2.804 us
12905	03.792 570 089	IP_Task	/s OS_EVENT_Set	pEvent=0x1FFF0F30
12906	03.792 573 964	IP_Task	Task Ready	IP_WebServer, runs after 170.464 us
12907	03.792 578 661	IP_Task	/s OS_EVENT_Set	Returns void after 8.571 us
12908	03.792 595 607	IP_Task	/s OS_MUTEX_GetValue	pMutex=0x1FFF042C
12909	03.792 598 232	IP_Task	/s OS_MUTEX_GetValue	Returns 1 after 2.625 us
12910	03.792 601 571	IP_Task	/s OS_MUTEX_Unlock	pMutex=0x1FFF042C
12911	03.792 606 518	IP_Task	/s OS_MUTEX_Unlock	Returns void after 4.946 us
12912	03.792 609 929	IP_Task	/s OS_TASKEVENT_GetTimed	EventMask=0000'0001 Timeout=4 Ticks
12913	03.792 614 321	IP_Task	Task Block	Waiting for Task Event with timeout

Name	Type	Stack Information	Activations	Total Run Time	Time Interrupted	Min Run Time	Max Run Time
ETH_RX	#100		188	0.006 848 994 s	0.000 000 ms	0.004 357 ms	0.066 643 ms
ETH_TX	#99		171	0.002 918 768 s	0.000 000 ms	0.004 393 ms	0.057 750 ms
SysTick	#15		10 530	0.051 237 833 s	0.000 000 ms	0.004 643 ms	0.024 143 ms
Scheduler			2 675	0.022 078 815 s	0.315 893 ms	0.005 179 ms	0.012 607 ms
IP_Task	@152 1280 @ 0x1FFF537C		1 279	0.075 248 131 s	3.413 690 ms	0.012 179 ms	0.424 429 ms
IP_WebServer	@151 3072 @ 0x1FFF0454		39	0.003 670 036 s	0.014 286 ms	0.009 821 ms	0.190 464 ms
Webserver Child	@150 2000 @ 0x1FFF5978		147	0.028 838 440 s	2.274 482 ms	0.009 107 ms	1.176 750 ms
Webserver Child	@150 2000 @ 0x1FFF6148		63	0.013 861 399 s	0.675 161 ms	0.003 964 ms	1.140 107 ms
Idle			1 147	10.376 365 571 s	54.282 976 ms	0.003 429 ms	9.890 607 ms

## イベント

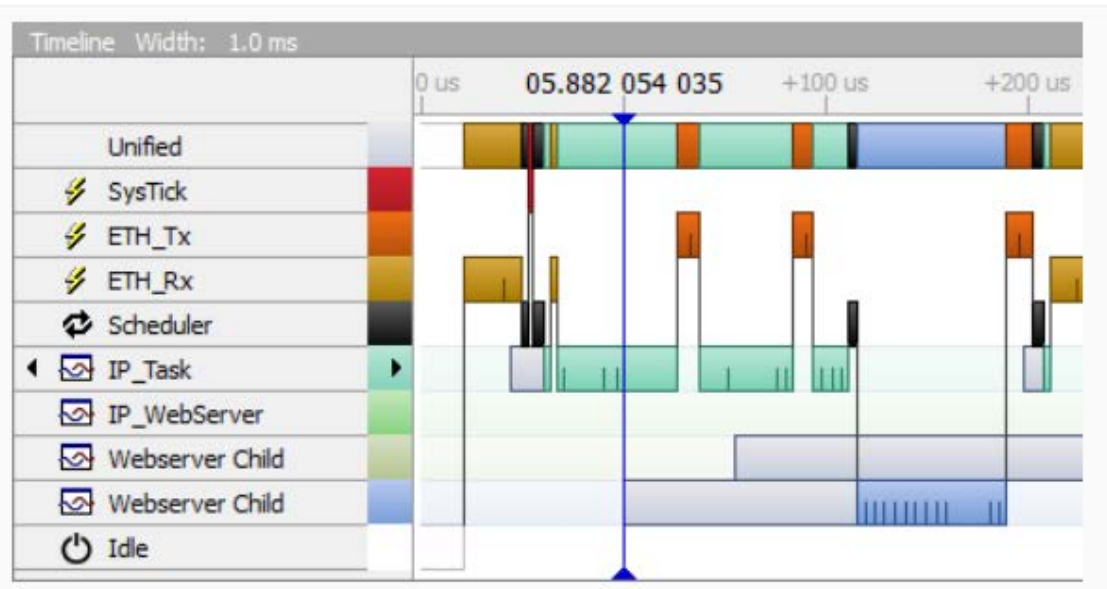
- 割込が発生する頻度
- どの割込でどのようなタスクが動いているか
- タスクや割込が中断されたタイミング
- スケジューラのタスク切り替えトリガー発出時点
- タスクの実行時間
- 割込の実行時間

通常のイベントの長さは 4 ~ 8 バイトで、200 MHz で記録されるまでに約 1 us かかります。毎秒10,000イベントの場合、SystemViewによって追加されるオーバーヘッドはCPU時間の1%未満であり、データ量は帯域幅制限内に簡単に収まります。

イベントのタイムスタンプは、1 CPU サイクルの精度 (200 MHz CPU では 5 ns に相当) にすることができます。



アプリケーションから取得したデータを元に生成された時間、発生したタスク、割込、発生したイベント、イベントの詳細情報を表示します。



## イベントタイムライン

タイムラインには、割込とイベントの動作状況及びスケジューラの動作状況とアイドル時間を表示します。時間軸を拡大縮小し、全体を俯瞰したり、細部の動きを確認できます。これにより、いつ、どのくらいの時間、なぜタスクが実行されたのか、割り込みで何が起こったのかを簡単に分析できます。

以下のようなシステム上の問題点を把握します。

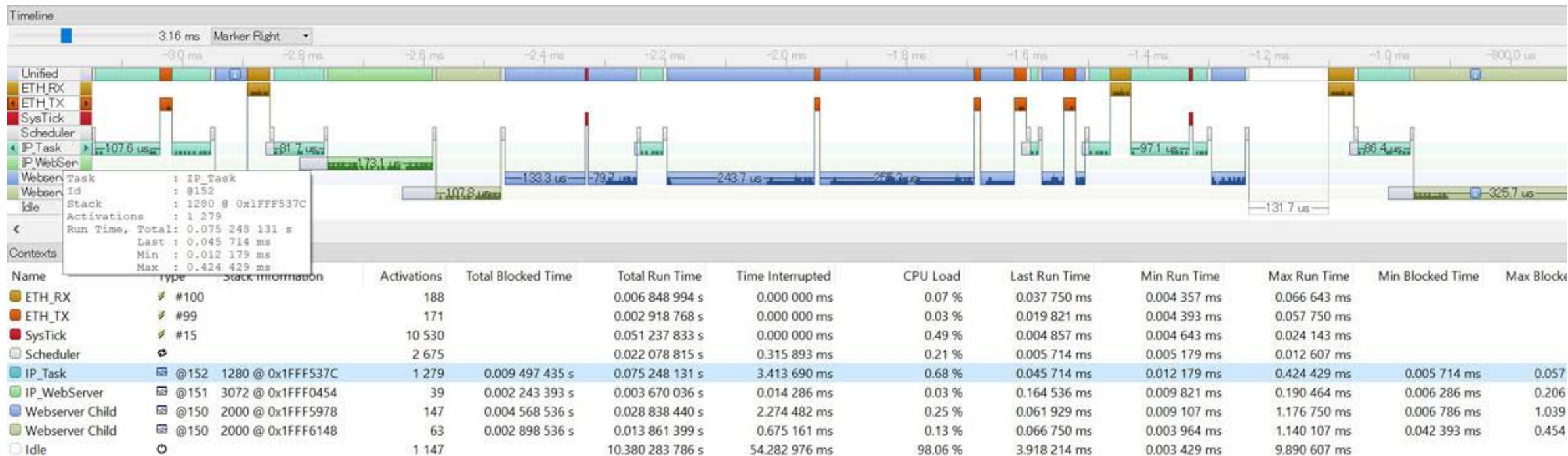
- タスクの優先順位が不正、または優先順位が逆転
- タスク間通信が正しくない
- 非効率的な遅延とタイムアウト
- 不適正／不要な割り込み



アプリケーションタスクの実行時間、CPU負荷を確認し設計意図通りの動作をしているか確認

# コンテキスト

タスクと割込に関する実行状況、頻度、実行時間、CPUの負荷情報が表示されます。  
 この情報を利用する事により、設計者の意図通りにCPUの負荷、アプリケーションが均等に稼働しているかなどを正確に把握する事ができます。



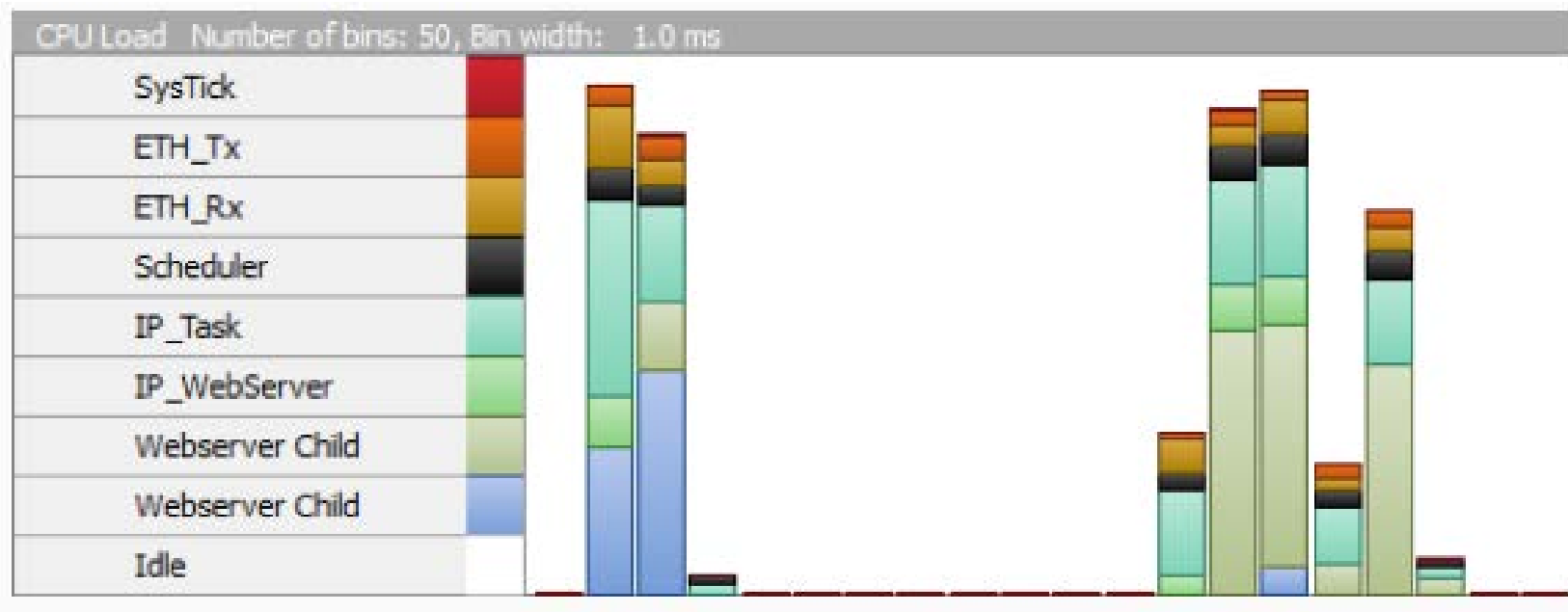
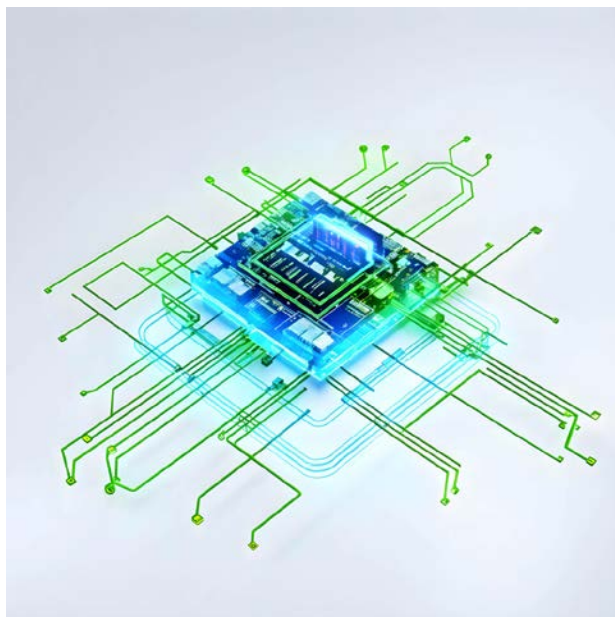




アプリケーションタスクの実行時間、CPU負荷を確認し設計意図通りの動作をしているか確認

# CPUロード

一定時間のコンテキスト別のCPU利用時間が表示されます。これにより、アプリケーションがCPUをどの程度の時間アクティブ利用しているか、アイドル時間をどれだけ作れているかという分析が可能になり、非効率的なアプリケーションの動きをしている箇所（不必要に割込頻度の高い部分）や単純な計算にかかる時間を測定することができます。

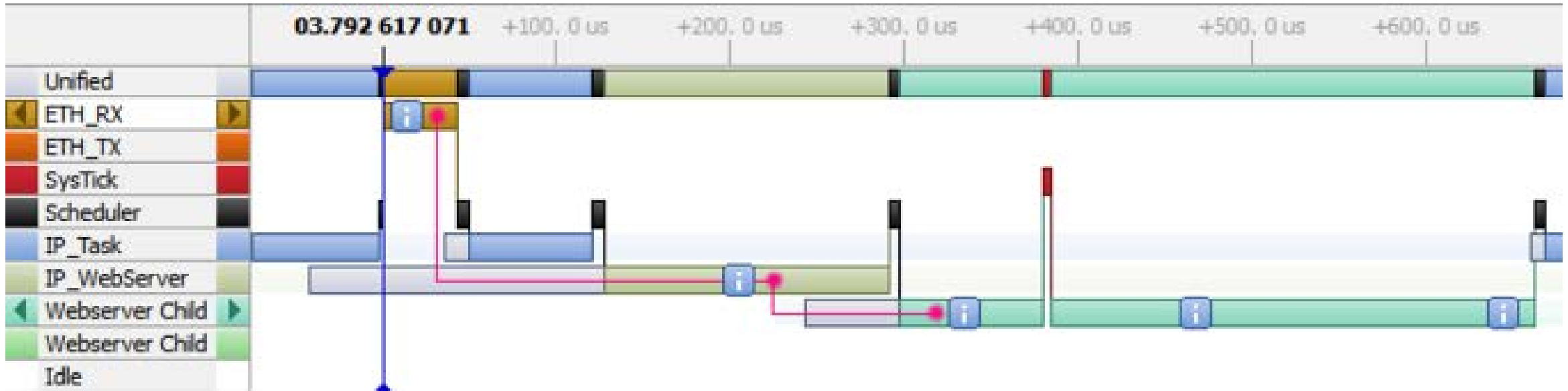




マーキングしたポイントを詳細に分析、実行時間、実行回数などを確認

# マーキング分析

システム内の特定のポイントをマーキングする機能を提供します。ポイント A からポイント B まで、またはポイント A からポイント B からポイント C までの期間を簡単に測定するために、マーカー開始、マーク、およびマーカー停止イベントを生成できます。SystemView アプリケーションは、対応するマーカーを自動的にリンクし、実行時間や測定の実行回数などの詳細情報を追加します。





エラー、メッセージログなどを記録。Printf()に似た文字列フォーマットをサポート

Timestamp	Context	Message
00.646 149 661	IP_Task	LINK: Link state changed: Full duplex, 100MHz
00.966 188 554	IP_Task	NDP: Link-local IPv6 addr.: FE80:0000:0000:0000:0222:C7FF:FEFF:6868 added to IFace: 0
01.646 068 839	IP_Task	DHCPc: Sending discover!
02.160 628 625	IP_Task	DHCPc: IFace 0: Offer: IP: 192.168.11.100, Mask: 255.255.0.0, GW: 192.168.13.1.
02.646 057 732	IP_Task	DHCPc: IP addr. checked, no conflicts
02.646 076 911	IP_Task	DHCPc: Sending Request.
02.647 171 625	IP_Task	DHCPc: IFace 0: Using IP: 192.168.11.100, Mask: 255.255.0.0, GW: 192.168.13.1.
03.793 054 946	Webserver Child	WebS: Get /
03.811 920 982	Webserver Child	WebS: Get /Styles.css
03.813 549 768	Webserver Child	WebS: Get /Logo.gif
03.815 742 839	Webserver Child	WebS: Get /BGround.png
04.740 649 304	Webserver Child	WebS: Get /
05.333 726 911	Webserver Child	WebS: Get /
05.797 747 554	Webserver Child	WebS: Get /
06.046 815 804	Webserver Child	WebS: Get /Styles.css
06.048 682 661	Webserver Child	WebS: Get /Logo.gif
06.050 858 911	Webserver Child	WebS: Get /BGround.png
06.207 269 554	Webserver Child	WebS: Get /
06.223 791 161	Webserver Child	WebS: Get /Styles.css
06.225 423 232	Webserver Child	WebS: Get /Logo.gif
06.227 546 768	Webserver Child	WebS: Get /BGround.png

## ログ出力

SystemView には、メッセージのログ記録も含まれています。簡単な文字列は、ログ、警告、またはエラーメッセージとして記録できます。

ロギング関数を利用することにより、printf() に似た文字列のフォーマットをサポートします。この場合、文字列の書式設定を行う必要性と追加のメモリが必要になります。ターゲットシステムは単にフォーマット文字列とパラメータをイベントに記録するだけで、SystemViewは文字列のフォーマットを処理し、ターミナルウィンドウに出力します。



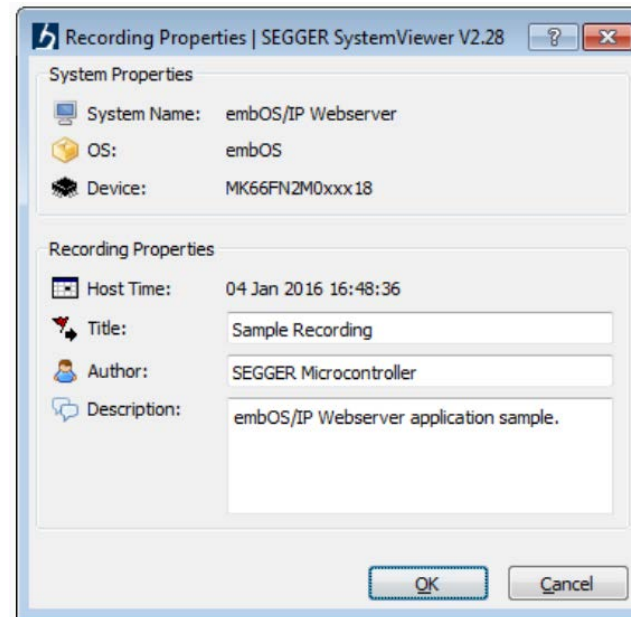


## システム情報表示

Property	Detail
Target System	
Name	embOS/IP Webserver
OS	embOS
Device	MK66FN2M0xxx18
Cycle Frequency	168 000 000 Hz
Cycle Period	6 ns
Uptime	00:00:08.153.000.113
Recording	
Host Time	04 Jan 2016 16:48:36
Duration	00:00:08.122.001
Title	Sample Recording
Author	SEGGER Microcontroller
Description	embOS/IP Webserver application sample.
Tasks	
Switch Count	1 260
Frequency	115 Hz
ISRs	
Load	0.34%
Frequency	1 009 Hz
Total Time	29.9091 ms
Interrupt Count	8 350
Events	
RTT	

# システムインフォメーション

組込システムの状態情報を確認することができます。  
データ取得者設定をし、どの時点でのシステムを記録したデータなのかを後から確認する事なども可能です。



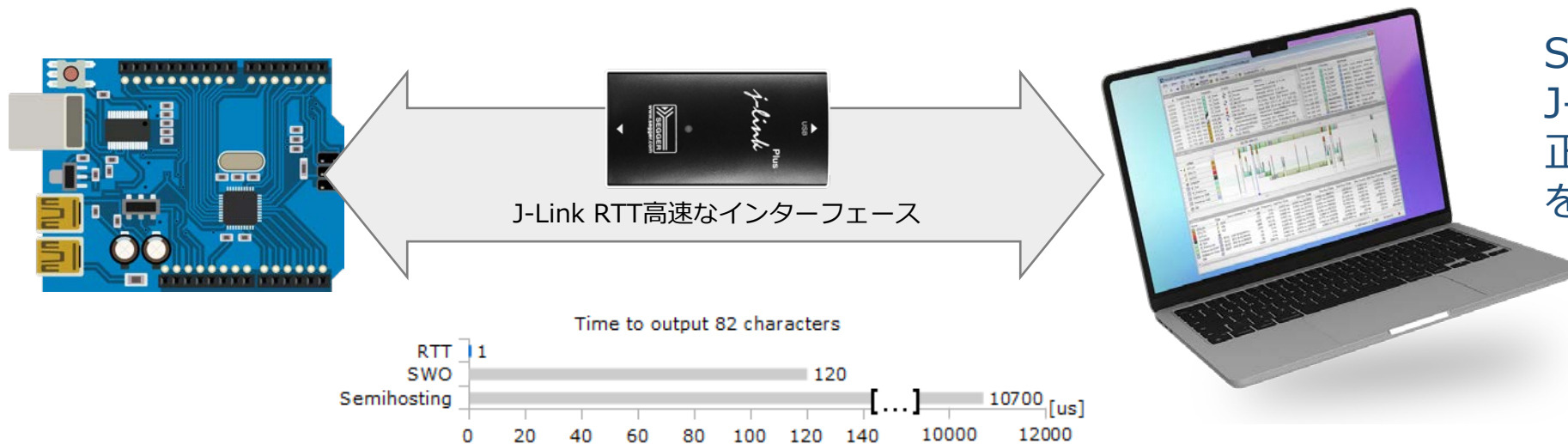


## SystemView テクニカル情報

- J-Link RTTによる高速データ取得
- データ記録方法
- 対応RTOS/環境

お客様アプリケーション動作にほとんど影響を与えないデータ取得・分析

非常に高いパフォーマンスで SWO とセミホスティングの長所を兼ね備えています。



SystemViewは  
J-Link RTTにより  
正確なアプリケーション情報を  
取得できます。

J-Link RTTは、複数のチャンネルを両方向（上限のホストから下限のターゲットまで）でサポートします。これらのチャンネルは、様々な目的に使用でき、ユーザーに可能な限り自由度を提供。

デフォルトの実装では、方向ごとに一つのチャンネルが使用されます。これはプリント可能なターミナルの入出力を目的としています。J-Link RTT ビューアでは、複数の「仮想」端末にこのチャンネルを使用することができ、一つのターゲットバッファで複数のウィンドウ（例：標準出力用、エラー出力用、デバッグ出力用）にプリントできます。（ホストへの）追加チャンネルは、例えばプロファイリング又はイベントトレースデータの送信に使用できます。

対応CPUコア

Cortex-M / Cortex-R / Cortex-A / RISC-V / Renesas RX



## アプリケーションの動的メモリ使用量分析

Heap			
Time	Context	Resource	Detail
1.058 969 300	Audio Enumerator	0x03969720	• Allocate 44 bytes at 0x03969720, remains in use -- 43108510 used, 91109218 free, 32.11% full
1.065 910 600	Audio Enumerator	0x03969754	• Allocate 44 bytes at 0x03969754, remains in use -- 43108562 used, 91109166 free, 32.11% full
1.070 046 100	UI Task	0x03860100	• Allocate 8 bytes at 0x03860100, will be freed by event #3872, lifespan 6.842 ms -- 43108578 used, 91109150 free, 32.11% full
1.070 049 300	UI Task	0x03860110	• Allocate 8 bytes at 0x03860110, will be freed by event #3870, lifespan 6.838 ms -- 43108594 used, 91109134 free, 32.11% full
1.070 050 600	UI Task	0x03969788	• Allocate 32 bytes at 0x03969788, will be freed by event #3770, lifespan 2.800 us -- 43108634 used, 91109094 free, 32.11% full
1.070 051 800	UI Task	0x039697B0	• Allocate 48 bytes at 0x039697B0, will be freed by event #3869, lifespan 6.835 ms -- 43108690 used, 91109038 free, 32.11% full
1.070 053 400	UI Task	0x03969788	• Free 32 bytes at 0x03969788 allocated by event #3768, lifespan 2.800 us -- 43108650 used, 91109078 free, 32.11% full
1.070 053 900	UI Task	0x03969788	• Allocate 8 bytes at 0x03969788, will be freed by event #3871, lifespan 6.833 ms -- 43108666 used, 91109062 free, 32.11% full
1.074 139 700	Audio Enumerator	0x039697E8	• Allocate 44 bytes at 0x039697E8, remains in use -- 43108718 used, 91109010 free, 32.11% full
1.075 112 500	MIDI Enumerator	0x03969798	• Allocate 8 bytes at 0x03969798, will be freed by event #3811, lifespan 11.700 us -- 43108734 used, 91108994 free, 32.11% full
1.075 114 300	MIDI Enumerator	0x0396981C	• Allocate 32 bytes at 0x0396981C, will be freed by event #3810, lifespan 9.300 us -- 43108774 used, 91108954 free, 32.11% full
1.075 123 600	MIDI Enumerator	0x0396981C	• Free 32 bytes at 0x0396981C allocated by event #3809, lifespan 9.300 us -- 43108734 used, 91108994 free, 32.11% full
1.075 124 200	MIDI Enumerator	0x03969798	• Free 8 bytes at 0x03969798 allocated by event #3808, lifespan 11.700 us -- 43108718 used, 91109010 free, 32.11% full
1.075 127 000	MIDI Enumerator	0x0396981C	• Allocate 24 bytes at 0x0396981C, remains in use -- 43108750 used, 91108978 free, 32.11% full
1.075 134 700	MIDI Enumerator	0x0396983C	• Allocate 40 bytes at 0x0396983C, remains in use -- 43108798 used, 91108930 free, 32.11% full
1.075 137 700	MIDI Enumerator	0x03969798	• Allocate 8 bytes at 0x03969798, remains in use -- 43108814 used, 91108914 free, 32.11% full
1.075 139 400	MIDI Enumerator	0x0396986C	• Allocate 24 bytes at 0x0396986C, remains in use -- 43108846 used, 91108882 free, 32.11% full
1.075 141 200	MIDI Enumerator	0x0396988C	• Allocate 24 bytes at 0x0396988C, remains in use -- 43108878 used, 91108850 free, 32.11% full
1.075 142 400	MIDI Enumerator	0x039698AC	• Allocate 12 bytes at 0x039698AC, remains in use -- 43108898 used, 91108830 free, 32.11% full
1.075 143 200	MIDI Enumerator	0x039698C0	• Allocate 8 bytes at 0x039698C0, will be freed by event #3834, lifespan 2.200 us -- 43108914 used, 91108814 free, 32.11% full
1.075 144 700	MIDI Enumerator	0x039698D0	• Allocate 8 bytes at 0x039698D0, will be freed by event #3835, lifespan 1.600 us -- 43108930 used, 91108798 free, 32.11% full
1.075 145 400	MIDI Enumerator	0x039698C0	• Free 8 bytes at 0x039698C0 allocated by event #3832, lifespan 2.200 us -- 43108914 used, 91108814 free, 32.11% full
1.075 146 300	MIDI Enumerator	0x039698D0	• Free 8 bytes at 0x039698D0 allocated by event #3833, lifespan 1.600 us -- 43108898 used, 91108830 free, 32.11% full
1.076 224 100	Audio Enumerator	0x039698C0	• Allocate 44 bytes at 0x039698C0, remains in use -- 43108950 used, 91108778 free, 32.11% full
1.076 886 800	UI Task	0x039697B0	• Free 48 bytes at 0x039697B0 allocated by event #3769, lifespan 6.835 ms -- 43108894 used, 91108834 free, 32.11% full

### ヒープ負荷状況の確認 潜在的なメモリリークを特定

アプリケーションが C/C++ ヒープ、複数のカスタムヒープ、または RTOS のメモリプールを使用している場合、利用状況を監視することができません。SystemView はメモリ状況を分析し、ヒープの負荷を確認、潜在的なメモリリークを特定、ピーク時のメモリ使用量を表示できます。

## ユーザー定義タグ

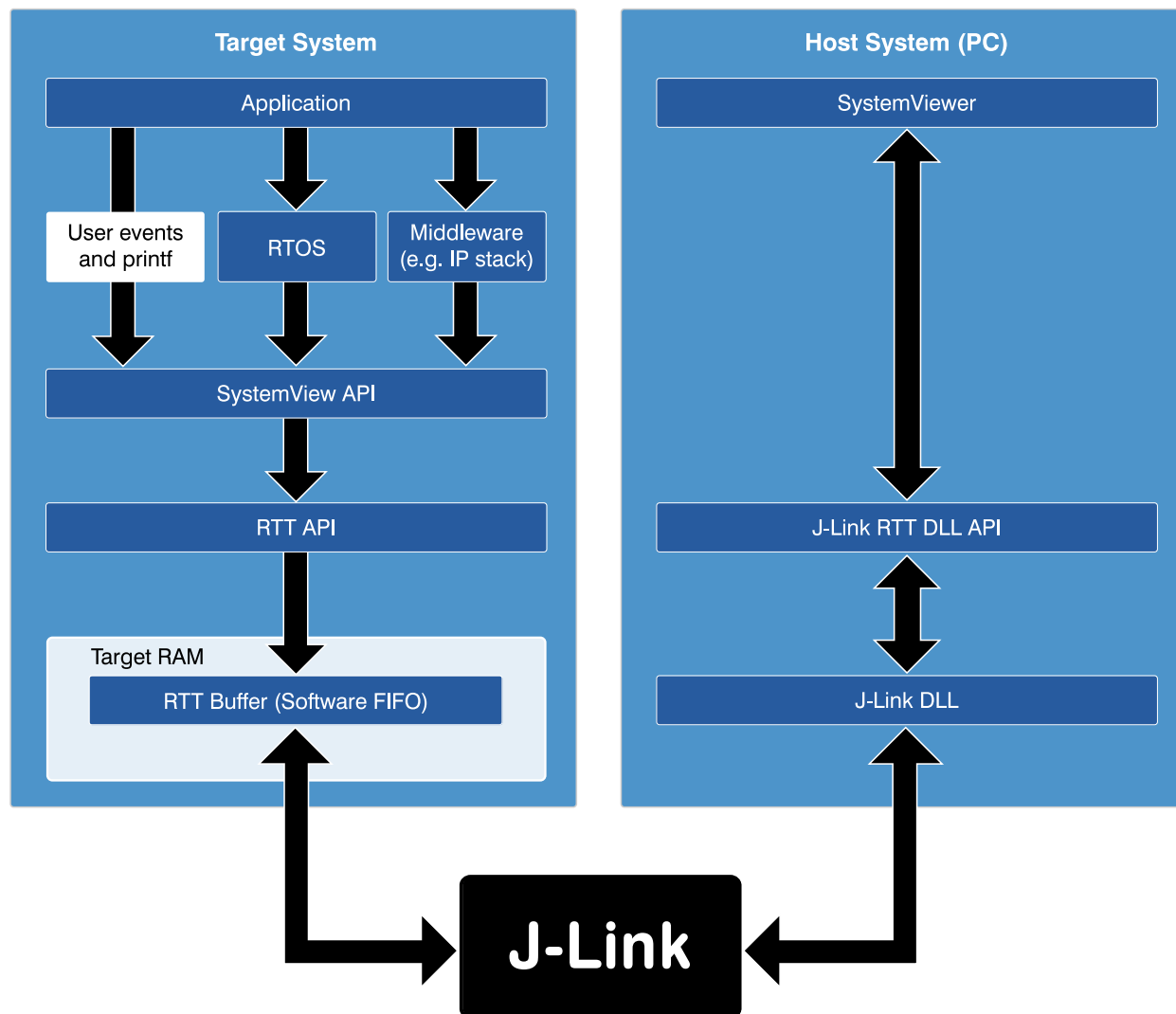
各メモリブロックには利用者任意でタグ付け可能。

例としてヒープ内のメモリブロックが割り当てられるアプリケーションサブシステム（トランスポート層、プロトコル層、データベースなど）を識別します。

## メモリリークの特定

収集した情報を表示し、ヒープの負荷状況を確認、潜在的なメモリリークを特定したり、ピーク時のメモリ使用量を追跡

## 継続的なデータ取得でアプリケーションを可視化



## 簡単に利用頂けます。

SystemViewで連続データ記録を開始するにはJ-Linkとターゲットを接続し、「記録の開始」をクリックするだけ  
 デバイス名選択し、必要に応じてJ-Link接続、ターゲットインターフェイス、および RTT制御ブロック検出を構成  
 設定後、SystemViewはJ-Linkに接続し、RTT経由でデータを継続的に読み取り、データを分析、表示ウィンドウを更新します。  
 記録されたデータは後で分析したり、データファイルとして保存できます。

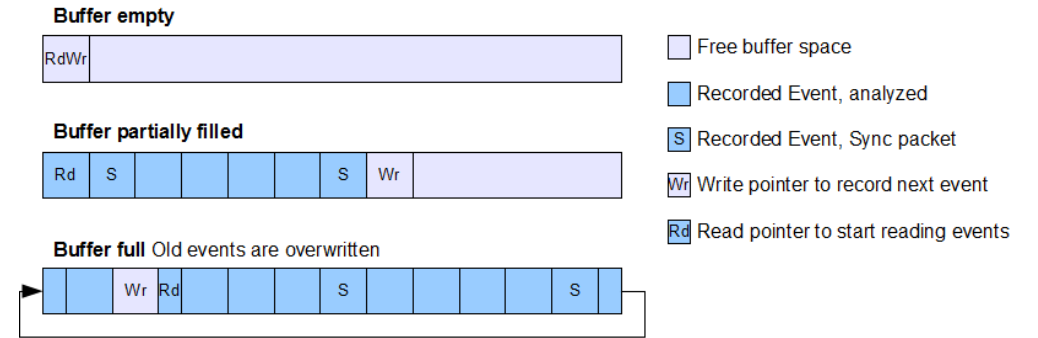
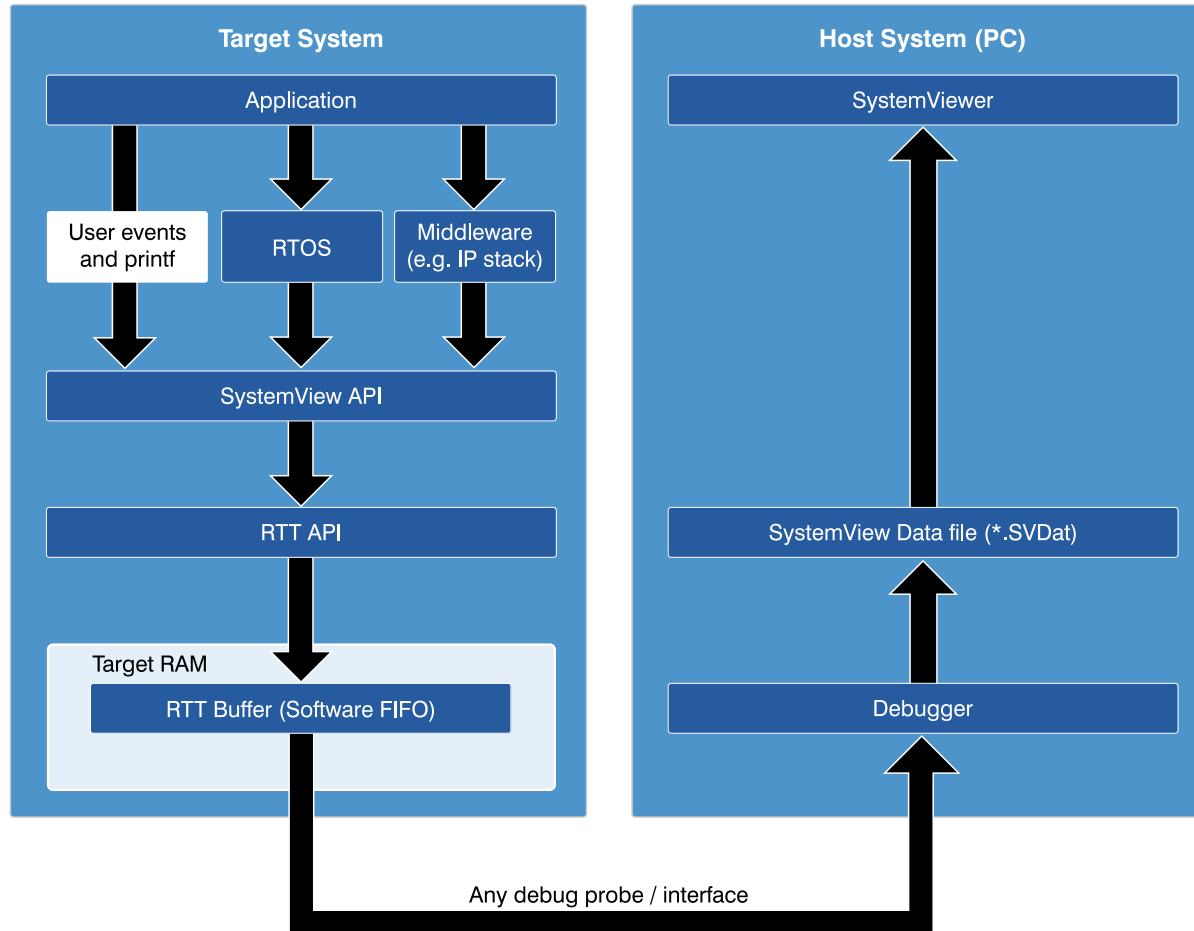


## デバッガと同時利用可能

SystemViewはデバッガと並行して利用可能  
 デバッガの実行中に記録を行うことができます。

詳細については、SystemViewユーザーマニュアルを参照ください。

## 問題解析したいポイントをデータ取得



## シングルショット記録モード

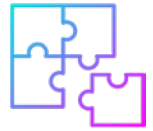
シングルショット記録モードではSystemView バッファがイベントでいっぱいになるまで、定義された開始点からシステムの動作を分析できます。システムの起動を記録し、システムの特定の部分が期待どおりに動作することを確認するために使用できます。

## 事後分析モード

事後分析モードでは、データを読み出すときに最新のシステム アクティビティが表示されます。システムが長時間実行され、突然クラッシュした場合や、システムの長時間テストで、数時間実行した後もシステムの動作をチェックおよび検証する場合に役立ちます。



## SystemViewが利用可能なRTOS・CPU環境



## 対応RTOS

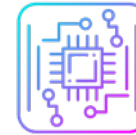
embOS /  FreeRTOS

## ■ 上記以外のRTOS環境

OS上でのタスクやOSの挙動を確認するためには、トレースデータ、プロファイリングデータをSystemViewで取得できる外部インターフェースが必要になります。OSによっては、そのような仕組みを持たないものもありますが、割り込みやユーザーイベントの取得を関数単位で取得することは可能です。

## ■ Non-RTOS・ベアメタル環境

割り込みの挙動を記録し、設計・要求仕様通りに発生することを確認できる他、モジュールの実行時間を測定するために使用できるユーザーイベントを記録できます。



## 対応CPU

Cortex-Mxコア搭載CPU  
Renesas RXコア搭載CPU

連続データ記録を利用する場合は「J-Link RTTインターフェース」を利用します。J-Link RTTについては、J-Linkユーザマニュアルで確認ください。

## ■ Cortex-A / Cortex-Rコア搭載CPU

連続データ記録の利用可否は、ターゲットによって異なります。J-Link RTTは、ターゲットの実行中にターゲットのメモリアクセスを必要とします。Cortex-AおよびCortex-Rでは、これはAHB-APを介して行われます。ターゲットデバイスにAHB-APがある場合、SystemViewは継続的に記録できます。AHB-APが搭載されていない場合は、シングルショットモードと事後モードのみがサポートとなります。

## ■ Classic-ARM (ARM7 / ARM9など) / その他のCPU

連続データ記録は利用できません。シングルショットモードと事後モードのみがサポートとなります。



# SystemViewライセンスモデル

商用開発可能ライセンス

評価フェーズは無償フル機能評価版が利用可能

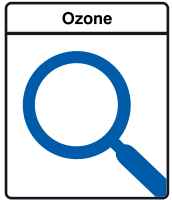
フェーズ	ターゲットボード	利用可能デバッガ	SystemView機能	ライセンス
評価	各種評価ボード	J-Link LITE J-Link on-board J-Link BASE (Compact) J-Link PLUS (Compact) J-Link ULTRA+ J-Link PRO J-Trace PRO	フル機能	無償評価版
商用開発	各種評価ボード 各種開発ボード 自社開発量産ボード	J-Link BASE (Compact) J-Link PLUS (Compact) J-Link ULTRA+ J-Link PRO J-Trace PRO	フル機能	商用開発 ライセンス



商用開発ライセンスは、利用するJ-Linkへライセンス登録されます。

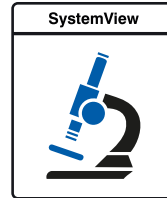
## J-Linkで利用頂けるソフトウェアツールライセンス

## デバッグソフトウェア

**OZONE(オゾン)**

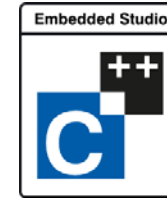
GCC/各種商用コンパイラで生成したELFファイルを利用可能

## システム可視化・記録

**SystemView**

ユーザアプリケーションの可視化記録解析ツール

## 統合開発環境

**SEGGER Embedded Studio**

GCC/LLVM/SEGGER独自コンパイラを搭載オリジナルリンカ・Cライブラリで、コンパクトな出力サイズを実現します。

**J-Link PLUS以上の製品  
利用で、無償商用利用可能**



**フル機能・期間制限なし無償評価版を提供  
SEGGER's Friendly License**

ユーザ様が評価利用とされている期間においては、無制限に無償利用可能  
正式開発採用される場合は、ライセンスをご購入ください。

<https://www.segger.com/purchase/licensing/license-sfl/>

**商用開発利用には、有償ライセンスが必要になります。**

Commercial-use License (CUL)

<https://www.segger.com/purchase/licensing/license-cul/>

教育・学術・趣味でのご利用はライセンス費用は不要



# 提供会社

EmbiTeK | SEGGER



# SEGGER Microcontroller GmbH



組み込みシステムで30年以上の経験を持ち、最先端のRTOSおよびソフトウェアライブラリを開発  
ハードウェアツール(開発 / 生産用)とソフトウェアツールをカバーします。

**CEO : Ivo Geilenbruegge**

**設立 : 1992年**

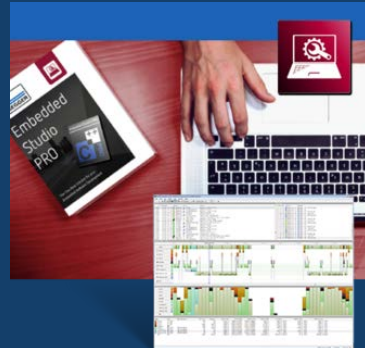
**本社 : モーンハイム・アム・ライン (ドイツ)**

**拠点 : 米国 / 中国**

**30カ国以上に販売代理店を通して展開**



RTOS/ミドルウェア



IDE

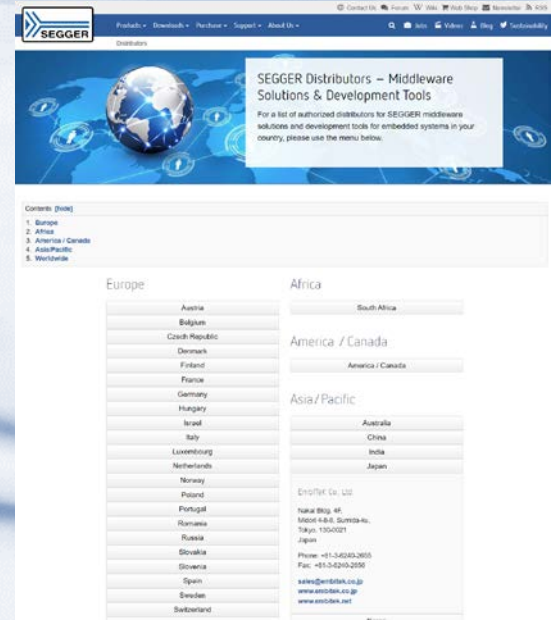


デバッグツール



書き込みツール

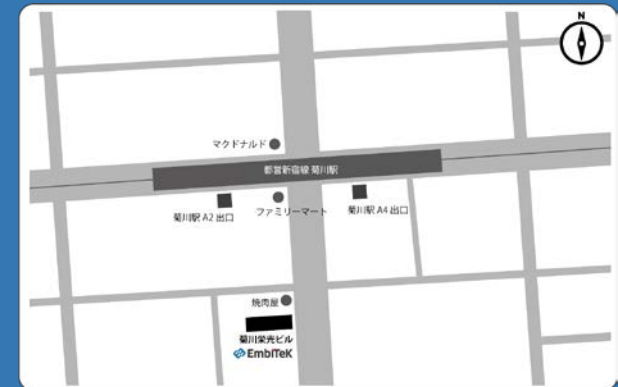




お客様の要件に合わせて、様々なシナリオで適合できる最適なソフトウェア開発環境ソフトウェアコンポーネントを提供します。

代表取締役：サントシュ パウル  
 設立：2007年  
 本社：東京都墨田区菊川2-3-6 菊川栄光ビル 601

日本国内唯一のSEGGER社製品販売オフィシャルパートナー  
 テクニカルサポート／ポーティング受託開発サービスを提供



都営新宿線「菊川駅」徒歩3分

# Arm Cortex/RXソフトウェア開発から量産をサポート

製品開発フローの課題に合わせて対応



デバッガ  
開発ツール

RTOS



MPU 対応

機能安全認証  
IEC61508 SIL3  
IEC62304 class C



SSL	暗号ライブラリ	セキュリティ認証	GUI
Modbus	SSH	ブートローダ	圧縮・解凍
IoT Toolkit HTTP client JSON Parser	MQTT Dropbox Client	USB Host HID MTP MassStorage CDC Printer FTDI LAN MIDI Audio HUB CCID CP21xx UART Video	
TCP/IP IPv4 / IPv6 DHCP server DHCP client ACP ARP AutoIP DNS client mDNS server LLMNR DNS-SD Loopback ICMP NetBIOS NS CoAP RAW sockets FTP server FTP client SMTP client SNMP Agent SNTTP client NTP client PTP OC client TCP UDP Web Socket client Web server UPnP Web Socket server PPP/PPPoE Wifi support		USB Device HID MSD (virtualMSD) MTP CDC-ACM CDC-NCM CDC-ECM RNDIS IP-over-USB Printer MIDI Audio Video Bulk DFU	
ファイルシステム NAND SPI/QSPI フラッシュ NOR SD SDHC SDXC MMC eMMC CF USB メモリ			

Arm Cortex / RX CPU

量産書込







製品については、お気軽に以下窓口へお問い合わせください。

TEL : 03-6240-2655  
FAX : 03-6240-2656  
e-mail : sales@embitek.co.jp  
website : <https://www.embitek.co.jp>



**EmbiTeK Online Shop**

<https://www.embitek.shop/>



<http://www.youtube.com/@embitek>