SEGGER
www.segger.com

embOS

RTOS Embedded Software

EmbiTeK

embOSとは？

1992年リリース以降1,000ライセンス以上の販売実績、10億台以上の量産品出荷（SEGGER調べ）

「embOS」は省電力を求められるバッテリ駆動を前提としたシングルチップ製品からハイエンドシステムまで、
あらゆる用途に利用できハードウェアの性能を最大限に活用できます。グローバルの市場において産業機器・IoT機器・ネットワーク機器
コンシューマ・自動車・医療機器・航空宇宙電子機器など様々な分野で利用されています。

コンパクト実装
ROM:1.7KByte
RAM: 1KByte以下

幅広いCPU対応
Arm Cortex / RISC-V / RX
RH850 / RL78 / SH2A / AVR
PICxx / STM8 / MSP430他

ANSI-Cソースコード
MISRA-C2012コーディングルール
開発環境・コンパイラ依存なし

量産ロイヤリティなし
開発ライセンス
量産に対しての継続コストなし

洗練された
ソースコード

信頼性の高いソースコード

embOS はこれまで30年以上、市場実証され継続的に改善、最適化されている
非常に信頼性の高いコードです。特に信頼性を求められる市場でも高い評価を受けています。

整理されたファイル構造とAPI構造

・コンフィギュレーション・各種設定・運用を分かりやすいAPIで実装
・ユーザアプリケーションの最適化を柔軟に設定可能
・コンフィギュレータは不要で、アプリケーション側の編集・最適化を自由にハンドリング
・アプリケーションからタスクコンテキストの拡大可能

導入容易性
開発しやすさ

習得しやすいサンプルソースコード

豊富な標準サンプルソースコードで利用方法の習得がしやすくなっています。

開発ツールとの連携でデバッグ・アプリケーションの可視化

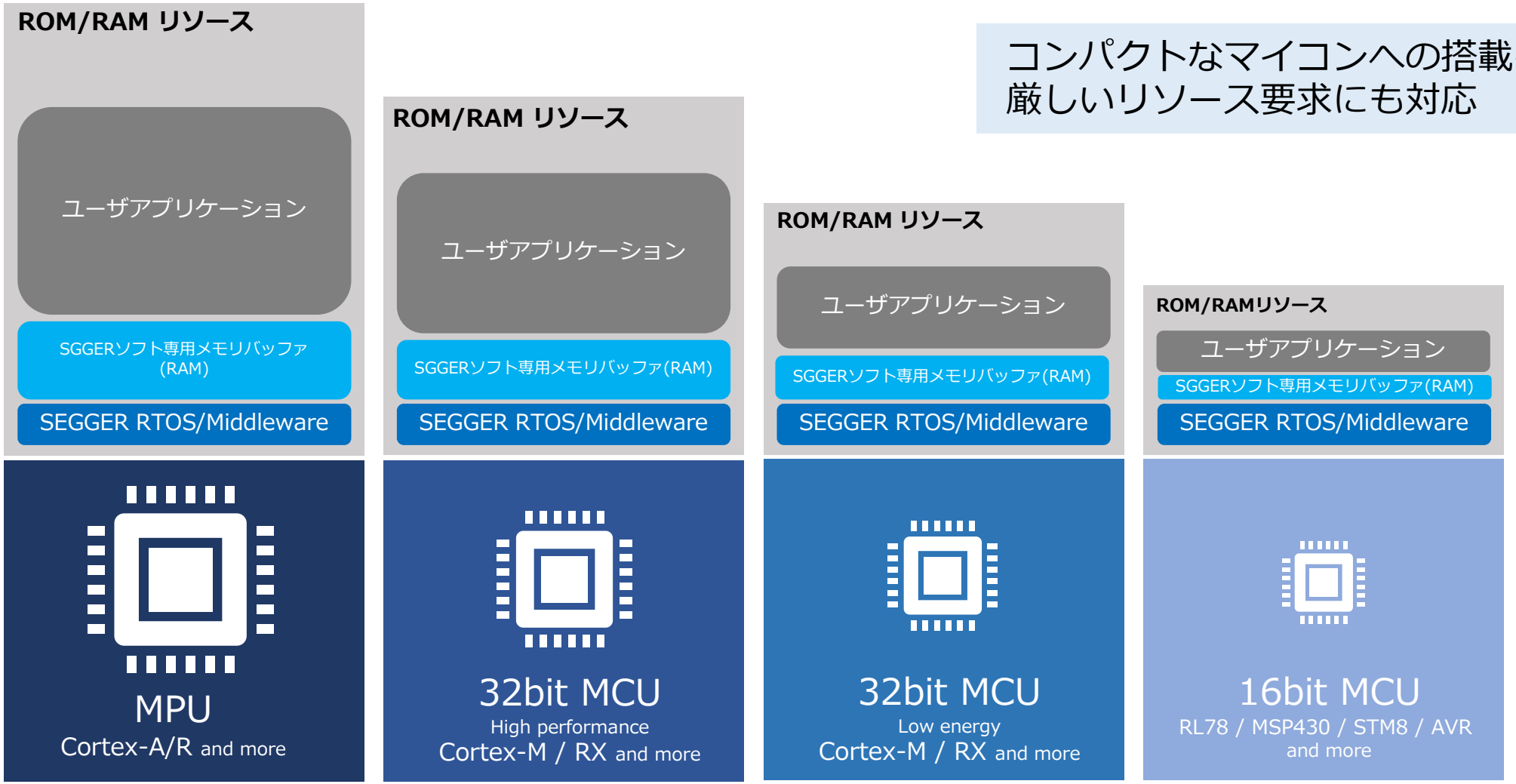
・RTOSプロファイル機能を標準実装
・SystemViewツールでRTOS機能のリアルタイム診断可能

ソフトウェア
継続性を強化

突然のマイコン変更や製品横展開を容易に実現

・ポーティング・マイコン対応が対応済みコアであれば、ユーザ様で簡単に実施可能
・コア制限のユーザライセンスで、追加コストなしにマイコン変更も容易に可能

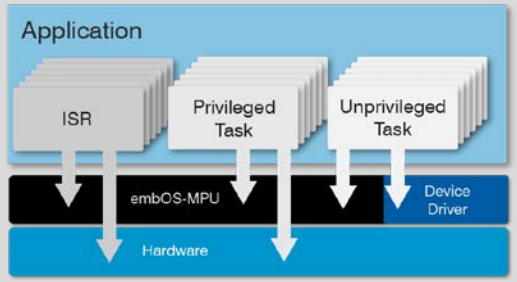
リソースの限られたマイコンでもハードウェアの性能を活かし、機能制限なく利用できるよう設計



コンパクトなマイコンへの搭載を前提に開発
厳しいリソース要求にも対応

SEGGERソフト専用メモリバッファ:
SEGGERソフトウェアが処理速度を向上させるために使用する一時的なメモリプール。ユーザアプリケーション、ハードウェアリソースに合わせて、領域サイズを変更可能です。

開発ニーズに合わせて選択できるライセンスモデル

標準ライセンス	メモリ保護	高精度・超省電力	機能安全認証
			
従来の「embOS」パッケージと変更はありません	高精度「embOS-Ultra」MPU対応パッケージも提供可能		
ソースコードライセンス オブジェクトコードライセンス	ソースコードライセンス オブジェクトコードライセンス	ソースコードライセンス オブジェクトコードライセンス	ソースコードライセンス
一般的なコア、コンパイラ、開発ツールで利用でき幅広いアプリケーションに対応のグローバルスタンダードRTOS	メモリ保護により、組み込みデバイスのセキュリティが強化。安定性と安全性を向上。	サイクル分解能タイミングを使用して、市場の他の RTOS より優れた精度と時間分解能を提供	TÜV SÜD Germany認証 IEC 61508 SIL 3 IEC 62304 Class C ISO 26262 ASIL D
30年以上の市場実績 ほとんどのマイコンで利用可能		 J-Link / Flasher シリーズ搭載RTOS	 コンパイラとCPUコアの組み合わせで厳密に認証取得

embOSの効率性

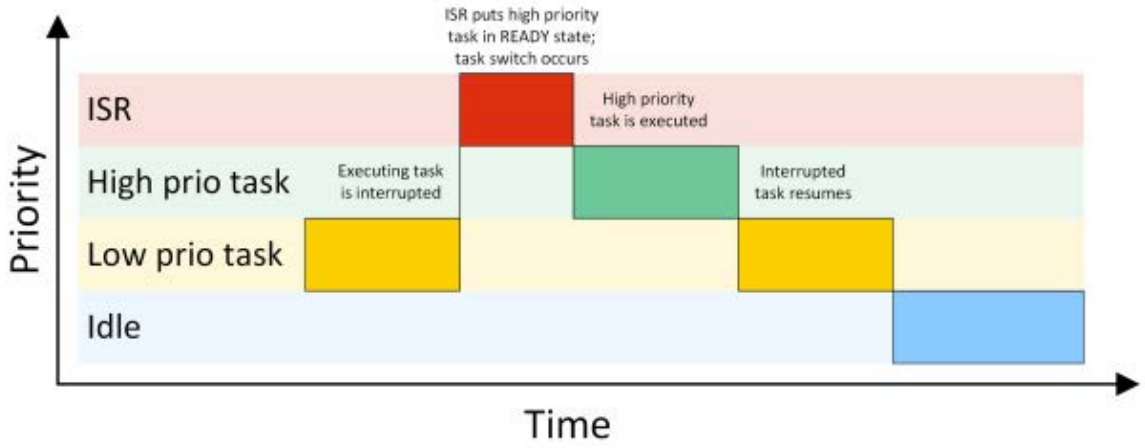
- マルチタスク
- 省ROM/RAMリソース

RTOSのオーバヘッドは最小限に
パフォーマンスの高いアプリケーション開発に最適です。

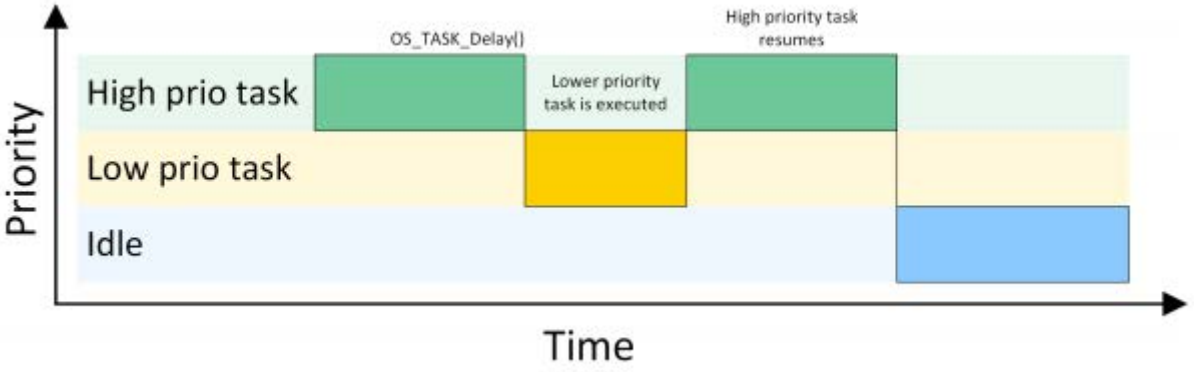
組込アプリケーションの基盤OSとして、設計されたリアルタイムOS

ユーザアプリケーションに最適なマルチタスクシステムを構築可能

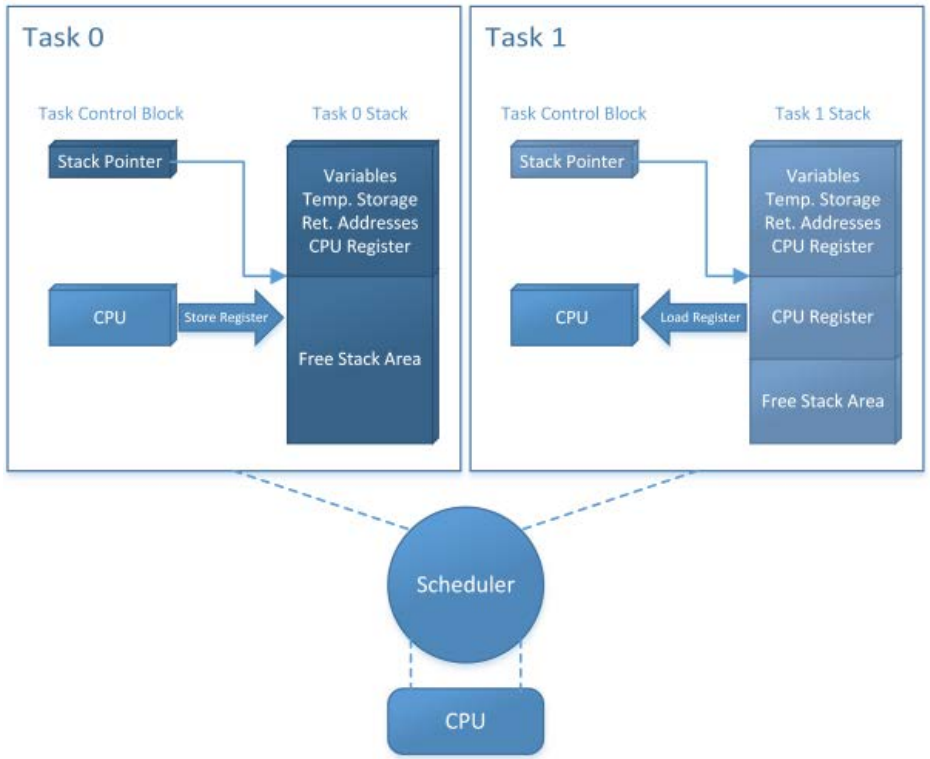
■ プリエンプティブマルチタスク



■ 非プリエンプティブマルチタスク

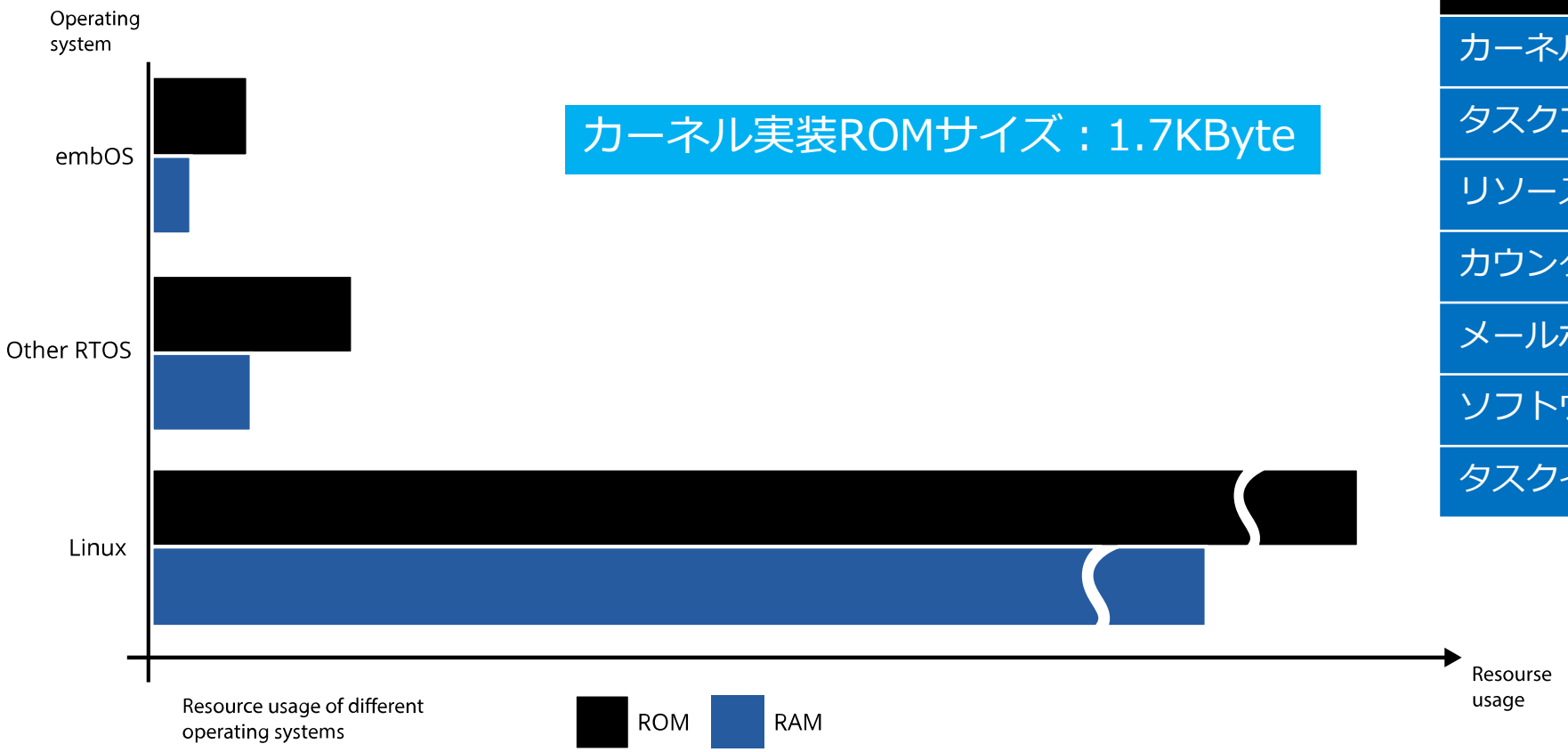


ラウンドロビン方式のスケジューリングや
プライオリティでコントロールするアルゴリズムと
小型のRTOSながら、高性能なRTOS機構を持ちます。



embOSの効率性：ROM/RAMサイズ

極めて小さなリソースで、効率的なアプリケーション実装が可能



カーネル実装ROMサイズ：1.7KByte

embOSリソース	RAM
カーネル	71 B
タスクコントロールブロック	36 B
リソースセマフォ	16 B
カウンタセマフォ	8 B
メールボックス	24 B
ソフトウェアタイマ	20 B
タスクイベント	0 B

(Byte)

コンパクトでありながら、性能・機能も優位性の高いRTOS

	embOS	µITRON
基本機能	タスク、ソフトウエアタイマ、リソースセマファ、カウントセマフォ、メールボックス、キュー、タスクイベント、イベントオブジェクト、可変長メモリプール（Heap）、固定長メモリプール	タスク、周期ハンドラ、アラームハンドラ、セマフォ、イベントフラグ、メールボックス、メッセージバッファ、ランデブ用ポート、可変長メモリプール、固定長メモリプール
スタック	システムスタック、タスクスタック、割り込みスタック	システムスタック、タスクスタック、**周期ハンドラ用スタック**、割り込みスタック
タスクスケジューラ	プリエンプティブ、Round-robin（同じ優先度のタスクの場合）	プリエンプティブ、Round-robin（同じ優先度のタスクの場合）
OSロック	OSロック機能	OSロック機能
タスク優先度の最大値	4,294,967,296	△
省エネ対応	**ticklessサポート、IOデバイスPower管理API**	△
WatchDog管理	**WatchDog管理API**	**△（アプリで実装）**

	embOS	µITRON
アプリケーションでタスクコンテキストの延長	**可能**	**×**
OSプロフィール・トレース	**標準サポート**	△
MPU対応	**MPU対応（embOS-MPU）**	**△（MMUドライバで実装）**
マルチコア	**マルチコアサポート（OS間同期・通信用API）**	**△（アプリで実装）**
OSデバッグツール	**自社ツール（embOSView、SystemView）**	**CS+用対応デバッグライブラリなし**
コンパイラ・デバッグツールサポート	一般C/C++コンパイラ、ELFデバッガ	一般C/C++コンパイラ、ELFデバッガ
Configuratorサポート	**無し**	△
ミドルウエアサポート	embOS対応はもちろん、OSなし又は他社RTOS環境でも使用可能な各種類のミドルウエアファミリー	自社RTOS専用のミドルウエアファミリー
シミュレーターサポート	有り	△

embOSの汎用性

- 汎用性の高いRTOS
- BSPサンプル提供のあるデバイス
- マルチコア・マルチOS対応

市場のほぼすべてのデバイス、開発環境で利用可能

利用したいデバイスで、使いたい開発環境で利用可能なRTOS

ハイエンドからローエンドマイコンまで、幅広く利用可能

MISRA-C2012に準拠したソースコード

80以上のコアとコンパイラの組合せサポート

500を超える評価ボードBSPを用意

8/16/32/64bitマイコンを一つのRTOSでカバーしています。

代表的なコア

ARM(Cortex-A/R/M, ARM7/9/11)、RISC-V、RX、RH850、RL78、FR16/30、STM8、MSP430、78K0、H8/H8S/H8SX、M16C/R8C、M32C、SH2/SH2A

主なIDE/コンパイラ

SEGGER Embedded Studio、IAR EWxxx、ARM MDK、ARM DS-5、GHS、ST Atollic TrueSTUDIO、GNU

BSPサンプル提供のあるデバイス

BSPサンプルのあるデバイスでは、すぐに利用可能。コア対応が取れていれば簡単にポーティング実装

デバイスメーカ	コア	マイコン・シリーズ
Ambiq Micro	Cortex-M	Apollo512-KBR, AMAPH1KK-KBR
Analog Devices	ARM7/9/11	ADuC7026
Cypress	Cortex-M	CY8C4245, CY8C5868, CY9BF506, CY9BF506, CY9BF618, S6E2
	F16	Fujitsu F2MC-F16LX 90xxx
	FR	MB913xx, MB914xx
GigaDevice	Cortex-M	GD32F150, GDF190, GD32F303, GD32F307, GD32F450
	RISC-V	GD32VF103
Holtek	Cortex-M	HT32F1253, HT32F52253
IDT	Cortex-M	ZAMC4100
Infineon	Cortex-M	TLE9877, XMC4300, XMC4500, XMC4700, XMC4800
	C16x	C16x
Maxim	Cortex-M	MAX32570, MAX32631
Microchip	Cortex-A	ATSAMA5xxx
	Cortex-M	SAM3x, SAM4x, SAMG5x, SAMDJ20xx, SAMD21xx,SAME70xx, SAML10xx, SAML11xx, SAMR21xx, SAMV71xx
	ARM7/9/11	AT91M5xxx, AT91R40xxx, AT91RM9xxx, AT91SAM7xxxx, AT91SAM9xxxx,
	AVR32	AVR32UC3xx, AVR32AP7000
	AVR	ATmega64, ATmega1xx, ATmega2xx, ATXmega1xx,
MindMotion	Cortex-M	MM32F103
Nordic	Cortex-M	nRF51xxx, nRF52xxx, nRF53xx
Nuvoton	Cortex-M	NUC029LAN, NUC442JI8
NXP	Cortex-A	iMX6Q, iMX6UL, i.MX6U5
	Cortex-M	i.MXRT595S, i.MXRT685S, i.MXRT1176, i.MXRT10xx, K21xxx, K22xxx, K24xxx, K26xxx, K40xxx, K60xxx, K64xxx, K65xxx, K66xxx, K70xxx, KE02xxx, KL25xxx, KV31xxx, LPC11xx, LPC12xx, LPC15xx, LPC17xx, LPC18xx, LPC43xx, LPC54xxx, LPC55xxx, S32Kxxx,
	ARM7/9/11	iMX25, LPC21xx, LPC23xx, LPC24xx, LPC31xx, LPC32xx

デバイスメーカ	コア	マイコン・シリーズ
Renesas	Cortex-A	RZ/A1, RZ/G1E
	Cortex-M	RE01
	RX	RX100, RX200, RX600, RX62x, RX63x, RX64x, RX65x, RX66x, RX71x, RX72x
	RH850	RH850¥F1x
	RL78	RL78(Simulator), RL78G13, RL78G14, RL78G1C, RL78L12, RL78L13
	SH2A/SH2	SH2A72xx, SH2A76xx, SH72xx
	V850	V850EIA1, V850EMS1, V850E2MN4, V850ESJG2, V850SA1, V850SB1
	78K	K078F0xx, K0R78F1xx
SiFive	RISC-V	E31, FE310-G002
Silicon Labs	Cortex-M	EFM32G, EFM32GG, EFM32HG, EFM32PG, EZR32LG
	8051	C8051F930
ST	Cortex-A	STM32MP157(A7)
	Cortex-M	STM32F0xx, STM32F1xx, STM32F2xx, STM32F3xx, STM32F4xx, STM32F7xx, STM32H7xx, STM32L0xx, STM32G0xx, STM32L1xx, STM32L4xx, STM32W10x, STM32WB, STM32MP157(M4)
	STM8	STM8
TI	Cortex-A	AM33xx, AM35xx
	Cortex-R	TMS570
	Cortex-M	CC13xx, LM3S9xx, LM3S8xx, LM3S19xx, LM3S69xx, LM3S89xx, MSP432, TM4C12xx, TMS47xx
	MSP430	MSP430F1xx, MSP430F5xxx, MSP430FGxxx
Toshiba	Cortex-M	TMPM330, TMPM369, TMPM3HQ, TMPM4G9, TMPM4GR, TMPM4NR MPM4KN
Xilinx	Arm Aarch64	XCZU3EG
	Cortex-A	XC7Z007S, XC7Z010, XC7Z020
	Cortex-R	XCZU3EG

マルチコア・マルチOSで利用可能なOS間同期・通信用APIをサポート

Routine	Description	main	Priv Task	Unpriv Task	ISR	SW Timer
OS_SPINLOCK_Create()	Creates a hardware-specific spinlock.	•	•			
OS_SPINLOCK_Lock()	Acquires a hardware-specific spinlock. Busy waiting until the spinlock becomes available. This function is unavailable for some architectures.	•	•			
OS_SPINLOCK_Unlock()	Releases a hardware-specific spinlock.	•	•			
OS_SPINLOCK_SW_Create()	Creates a software-implementation spin-lock.	•	•			
OS_SPINLOCK_SW_Lock()	Acquires a software-implementation spin-lock.	•	•			
OS_SPINLOCK_SW_Unlock()	Releases a software-implementation spin-lock.	•	•			

コアの同期、データ交換を制御するために利用可能なSpinlock APIが含まれます。

これにより、共有メモリ・周辺機器などへのアクセスを実現します。

汎用的なロックメカニズムにより、プロセスの同期を行う事ができます。

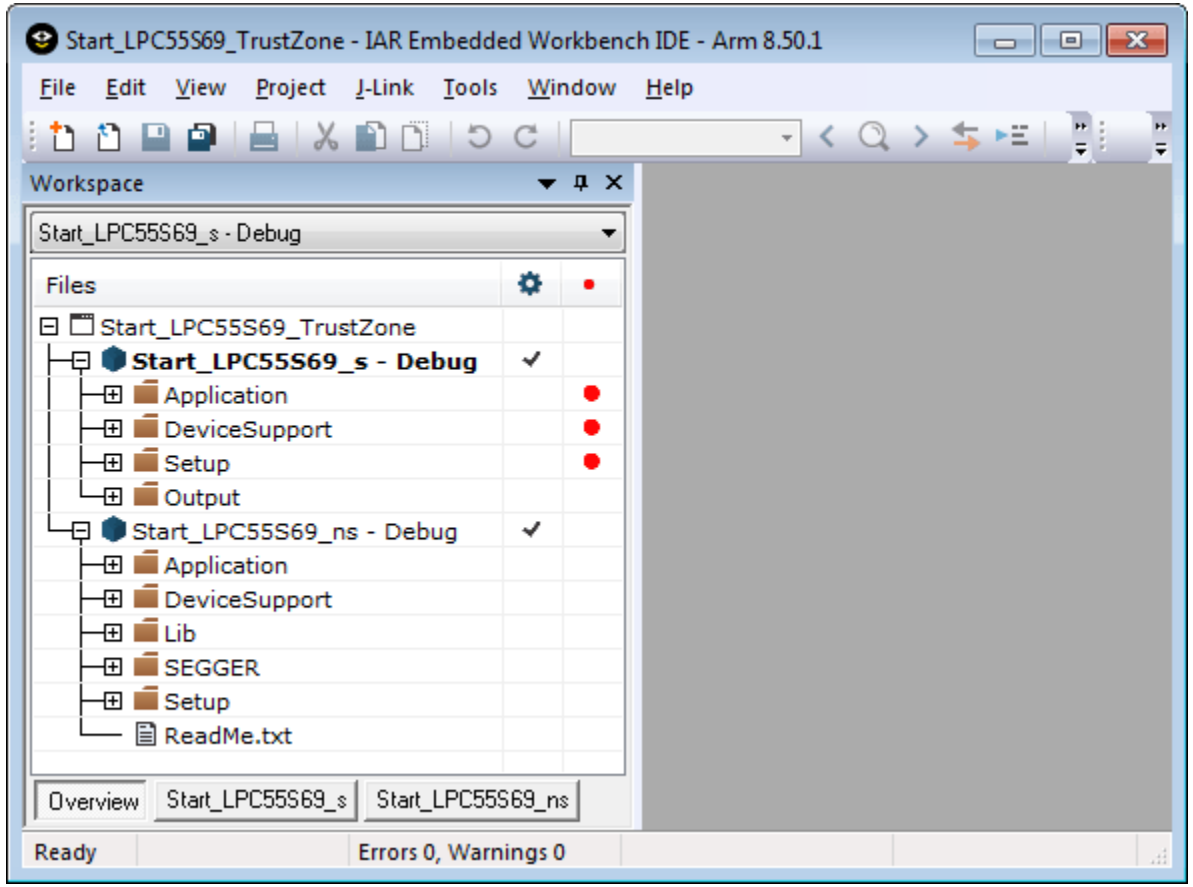
embOSすべてのパッケージ共通

embOSの特長機能

- Arm v8-M TrustZone対応
- 正確な実行時間測定
- ゼロ割込レイテンシー
- Tickless サポート

コンパクトながら高性能な機能を実装

セキュアエリアと非セキュアエリアを分離サポート対応



Cortex-Mマイコンで搭載されている「Arm v8-M TrustZone」を利用する事により、セキュアなアプリケーション設計をサポートします。セキュアブート、ファームウェアアップデート、キーなどをアプリケーションの他の部分から分離することで、外部の攻撃から守る安全なアプリケーションを開発することができます。

非セキュアエリアから実行するembOSアプリケーションでタスクはセキュア状態から関数を呼び出すことができます。

Cortex-M23/33などArm v8-Mアーキテクチャ搭載のマイコンデバイスで利用可能

タイミングを計算できるサイクルの正確な実行時測定をサポート

embOSは、タイマーサイクル解像度で実行時間を取得するAPI関数を提供します。
実際の解像度は、ハードウェアタイマーの周波数によって異なります。

例）ハードウェアタイマーが400 MHzで実行されている場合、
1タイマーサイクルは1秒/400.000.000サイクル= 2,5ナノ秒/サイクルに相当します。

OS_TIME_GetCycles（）は、タイマーサイクルで現在のシステム時刻を
64ビット値として返します。このAPIにより、オーバーフローすることなく
非常に長い期間を測定することもできます。
400 MHzのタイマー周波数を使用すると、最大1462年の期間を測定できます。

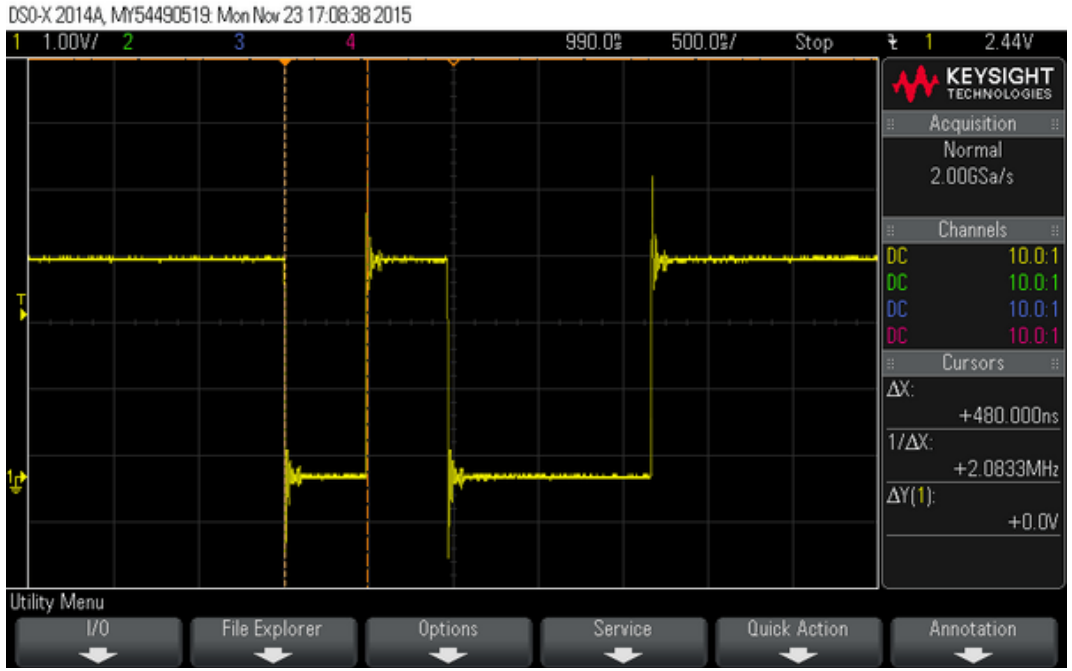
マイクロ秒の正確なタイミングのためのAPI関数も提供

```c
void MeasureTime(void) {
  OS_U64 tStart
  OS_U64 tEnd;

  tStart = OS_TIME_Getus64();
  DoSomeThing();
  tEnd = OS_TIME_Getus64();
  printf("Time to run the function: %u micro seconds", tEnd - tStart);
}
```

タイマーサイクルをナノ秒またはマイクロ秒に変換するAPI関数を提供

```c
void Convert(void) {
  OS_U64 cycles;
  OS_U64 nano_sec;
  OS_U64 micro_sec;

  cycles    = OS_TIME_GetCycles();
  nano_sec  = OS_TIME_ConvertCycles2ns(cycles);
  micro_sec = OS_TIME_ConvertCycles2us(cycles);
}
```

embOSはゼロレイテンシ割込をサポートしています。



優先度の高い、緊急性の高い割込が発生した際にRTOSのオーバヘッドを受けることなく、可能な限り早くコードを実行する事が、組込機器には求められます。

embOSではゼロレイテンシ割込はRTOSによる遅延が発生することなく、RTOSなしと同じくらい高速に処理する事のできる仕組みを持っています。

ゼロレイテンシ割込を利用したサンプルによる、パフォーマンス測定結果を以下URLにて公開しています。
https://www.segger.com/products/rtos/embos/technology/performance/

Tickless サポートにより省電力性を向上

embOSでは、各システムティック毎のタイマー割込を使う代わりにハードウェアタイマーを利用して対応します。

embOSのTicklessサポートは主に3つのAPIを利用して実現します。

OS_TICKLESS_GetNumIdleTicks（）
OS_TICKLESS_Start（）
OS_TICKLESS_AdjustTime（）

embOS Tickless サポートのサンプルは、以下URLにてソース公開しています。
https://www.segger.com/products/rtos/embos/technology/tickless-support/

embOSすべてのパッケージ共通

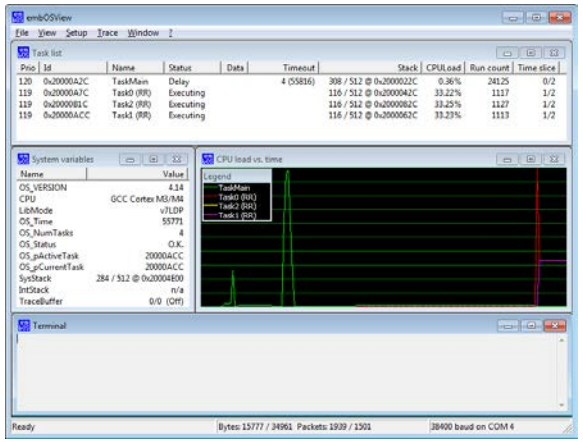
embOS開発・導入支援

- 開発支援ツール
- SystemView（商用開発有償）でアプリケーションを可視化・記録
- プロファイリングAPI
- ウェブ公開の製品マニュアル
- 評価ボード無償評価版
- プロフェッショナルサポート・エンビテックポーティング・実装支援

使いやすいAPIでアプリケーション開発、支援ツールも充実

ソフトウェア開発を支援する様々な無償提供ツール群

embOS View

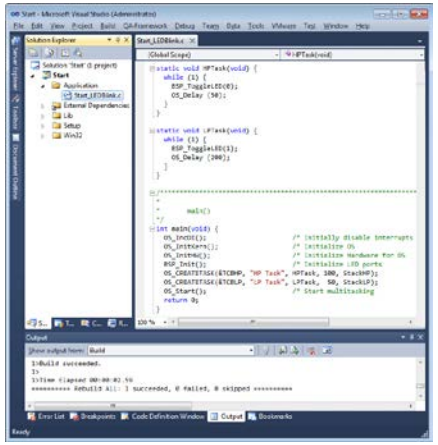


ターゲットで動作しているアプリケーション状況を可視化

ターゲットの接続
embOS Viewはターゲットとシリアル接続(UART)
ARM、RXの場合はEthernetや他の通信チャネルも対応可能

提供バージョン
すべてのembOSライセンス
無償評価版にも提供

embOS シミュレーション

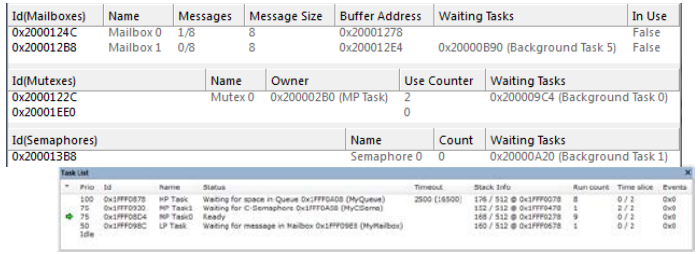


embOSシミュレーションはハードウェアを利用する事なくソフトウェアで動作をシミュレーションすることができます。すべてのembOS APIをシミュレーション可能

MinGW用／Microsoft VisualStudio用が用意されています。

embOSシミュレーションを使ってハードウェアシミュレーションをする事も可能です。LEDやディスプレイなどコントロールすることもできます。

RTOS認識プラグイン



Id(Mailboxes)	Name	Messages	Message Size	Buffer Address	Waiting Tasks	In Use
0x2000124C	Mailbox 0	1/8	8	0x20001278		False
0x200012B8	Mailbox 1	0/8	8	0x200012E4	0x20000B90 (Background Task 5)	False

Id(Mutexes)	Name	Owner	Use Counter	Waiting Tasks
0x2000122C	Mutex 0	0x200002B0 (MP Task)	2	0x200009C4 (Background Task 0)
0x20001EE0			0	

Id(Semaphores)	Name	Count	Waiting Tasks
0x20001388	Semaphore 0	0	0x20000A20 (Background Task 1)

SEGGER Embedded Studio
IAR Ewxxx対応

各種IDE上でembOSの状況を表示

タスクリスト：
各タスクレベルの状態を表示します。

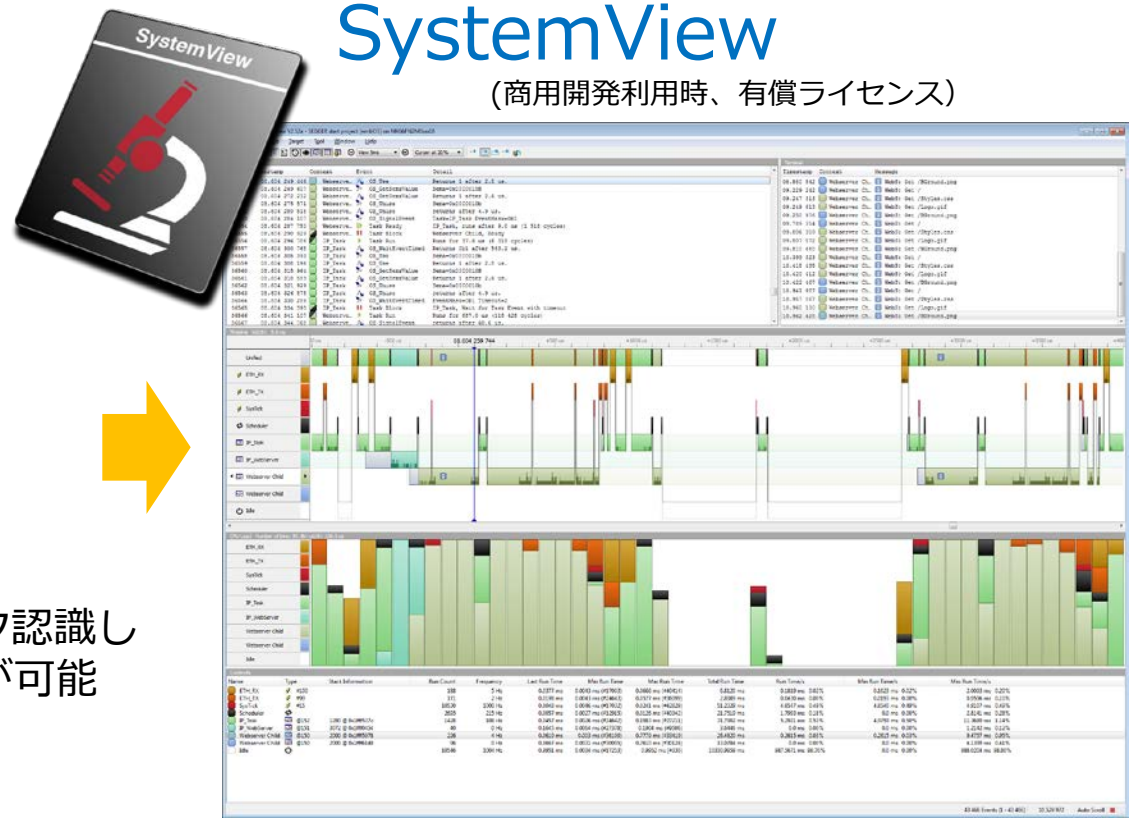
embOS上のオブジェクト、メールボックスやセマフォなどの状態を表示

アプリケーション全体が設計通りに機能しているか確認

最適化により、製品のパフォーマンスを改善

SystemView

(商用開発利用時、有償ライセンス)



- 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
- 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
- CPUの占有率や利用状況

embOSのOSタスク認識し
タイムライン表示が可能

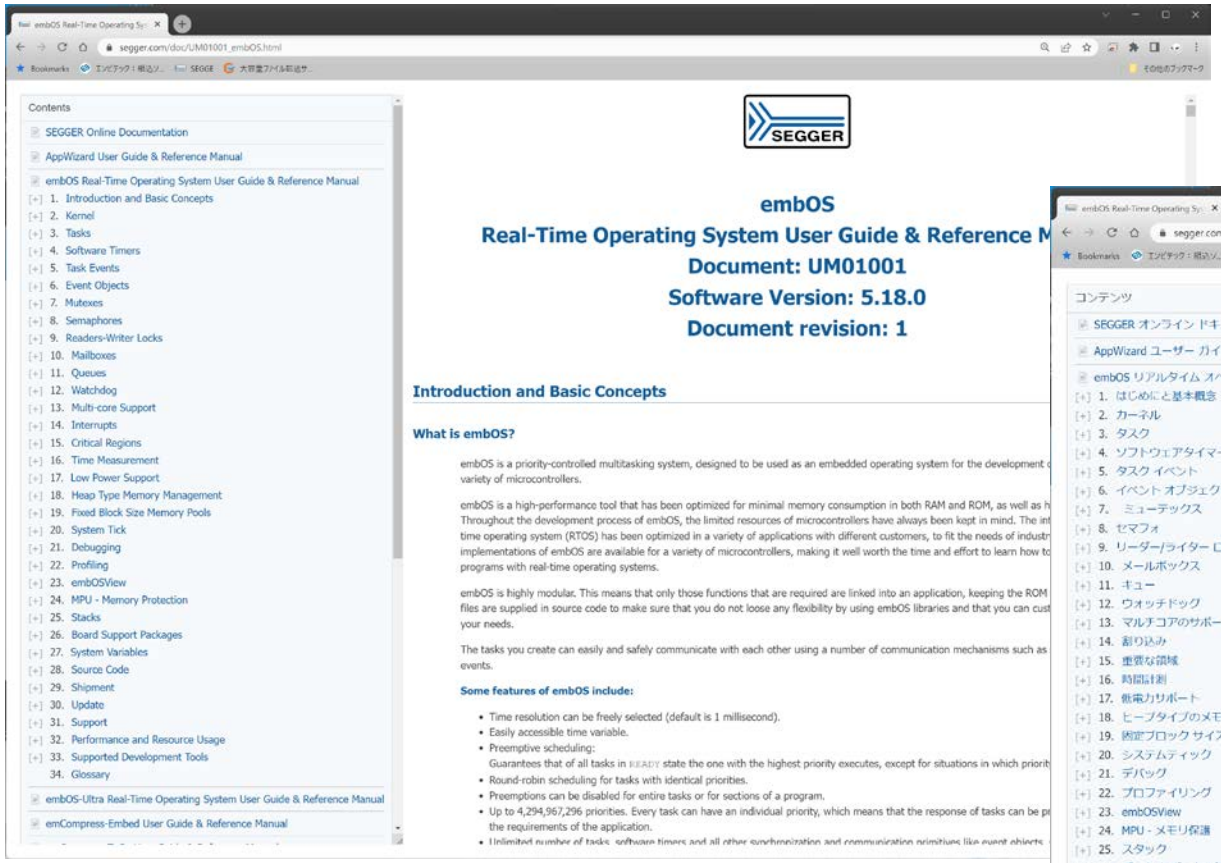
Cortex-M4/ M7 / M33搭載 200MHz以上のデバイスで毎秒10,000イベントの場合、SystemViewによって追加される
オーバーヘッドはCPU時間の1%未満であり、データ量は帯域幅制限内に簡単に収まります。

embOSでは、OSプロファイル・トレース機能を標準搭載

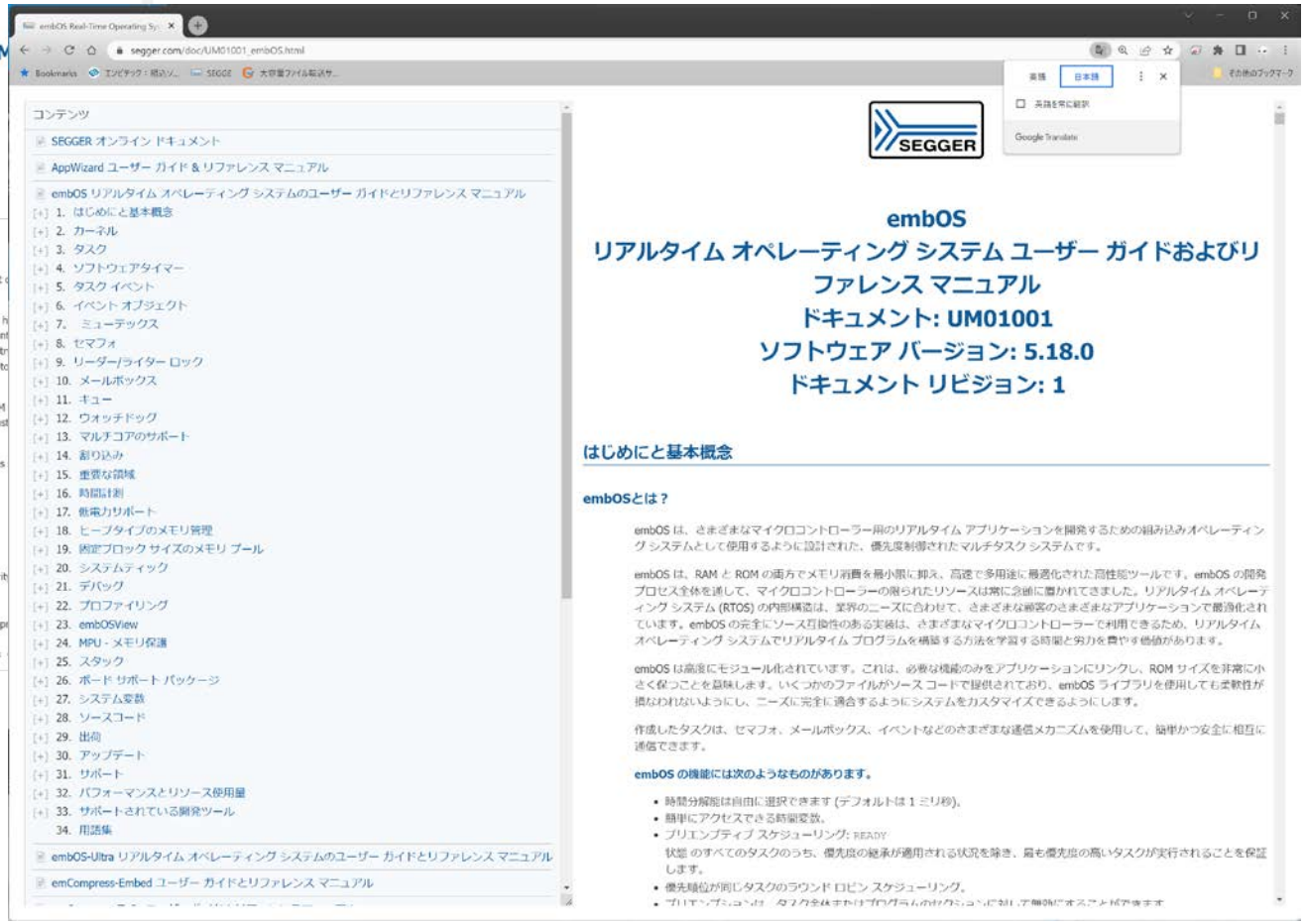
Routine	Description	main	Priv Task	Unpriv Task	ISR	SW Timer
OS_STAT_AddLoadMeasurement()	Initializes the periodic CPU load measurement.	•	•			
OS_STAT_AddLoadMeasurementEx()	Initializes the periodic CPU load measurement.	•	•			
OS_STAT_Disable()	Disables the kernel profiling.	•	•		•	•
OS_STAT_Enable()	Enables the kernel profiling (for an indefinite time).	•	•		•	•
OS_STAT_GetExecTime()	Returns the total task execution time.	•	•		•	•
OS_STAT_GetLoad()	Calculates the current task's CPU load in permille.	•	•		•	•
OS_STAT_GetLoadMeasurement()	Retrieves the result of the CPU load measurement.	•	•		•	•
OS_STAT_GetNumActivations()	Return the number of task activations.	•	•	•	•	•
OS_STAT_GetNumPreemptions()	Return the number of task preemptions.	•	•	•	•	•
OS_STAT_Sample()	Starts the kernel profiling and calculates the absolute task run time for all tasks since the last call to OS_STAT_Sample().	•	•		•	•

外部ツールに依存することなく、
TICKタイムだけでなく、μ秒単位の時間計測
APIなどを利用し、表示することも可能です。

製品マニュアルをライセンス導入前から閲覧可能



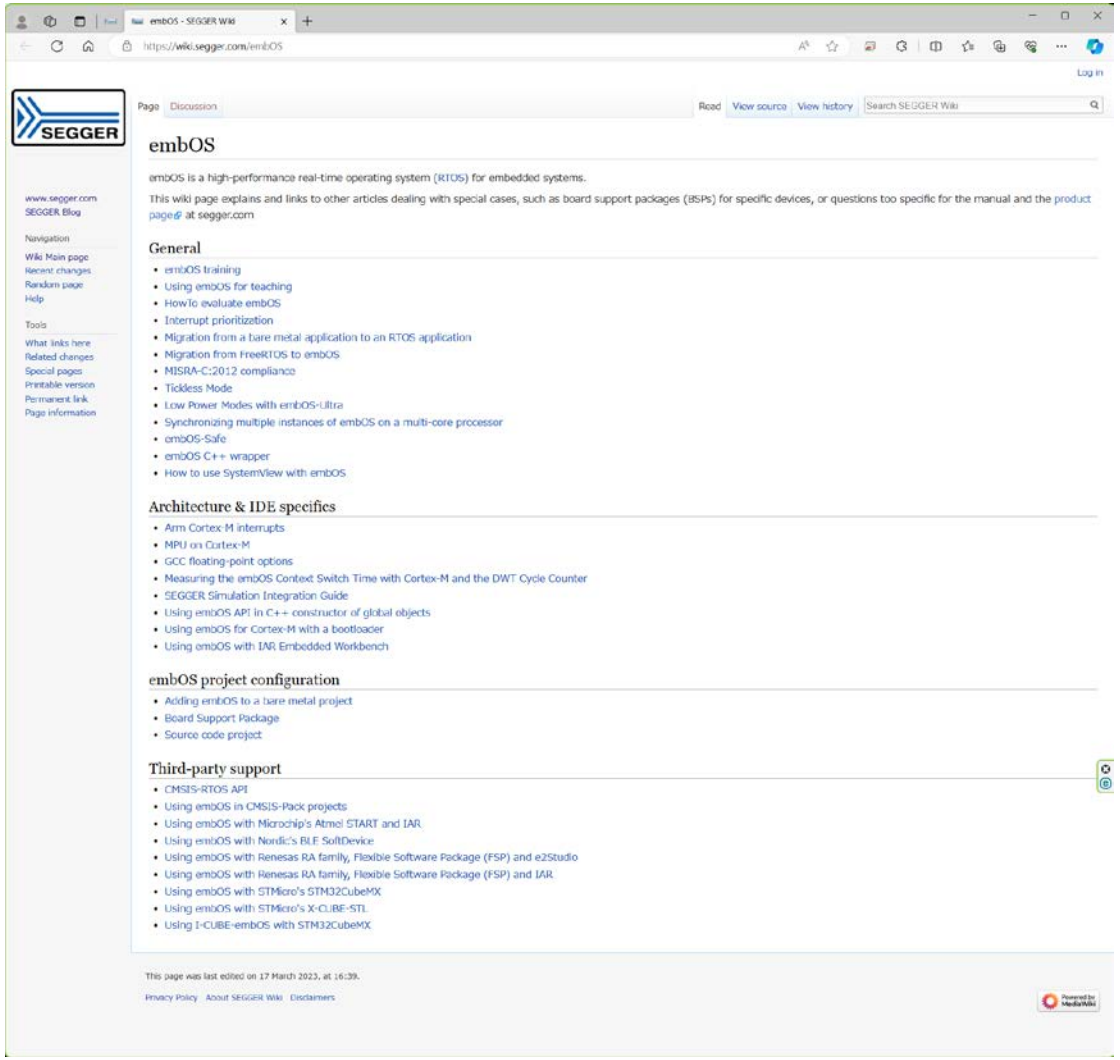
https://www.segger.com/doc/UM01001_embOS.html



ブラウザの翻訳機能で日本語表示可能

Google Chrome推奨

SEGGER Wikiで技術情報、Tipsを確認



https://wiki.segger.com/embOS

掲載内容

embOS の評価方法
割り込みの優先順位
ベアメタルから RTOS アプリケーションへマイグレーション
FreeRTOS からembOS へマイグレーション
MISRA-C:2012 準拠について
Tickless モード
embOS-Ultra による低電力モード
マルチコアCPUでのembOS複数インスタンスの同期
embOS-safe
embOS C++ wrapper
SystemView利用方法

Arm Cortex-M 割込み
Cortex-M MPU
GCC 浮動小数点オプション
Cortex-MとDWTサイクルカウンタを使用したembOSコンテキストスイッチ時間の測定
SEGGER シミュレーション
グローバルオブジェクトの C++ コンストラクタでの embOS API使用
ブートローダでembOS (Cortex-M)を使用する
IAR Embedded WorkbenchでembOS利用

など

ブラウザの翻訳機能で日本語表示可能 Google Chrome推奨

SEGGERのプロフェッショナルサポートとエンビテックの付加対応、お客様導入を受託対応でサポート

エンビテック受託開発サポート

＋

エンビテック日本語サポート
日本語問い合わせ窓口

＋

SEGGERライセンス保守
開発者サポート

＋

高性能・高品質なソフトウェア
embOS

お客様ハードウェアへのポーティング対応

embOSは比較的容易にお客様ハードウェアへの実装が可能ですが、
必要に応じて当社でお客様ハードウェアへの実装受託対応を行う事が可能です。

RTOSアプリケーションポーティング対応

既存のFreeRTOSやiTRONソフトウェアをembOSにポーティング対応します。
既存RTOSからのAPI調整などを行い、ソフトウェア資産を有効活用する提案も可能です。

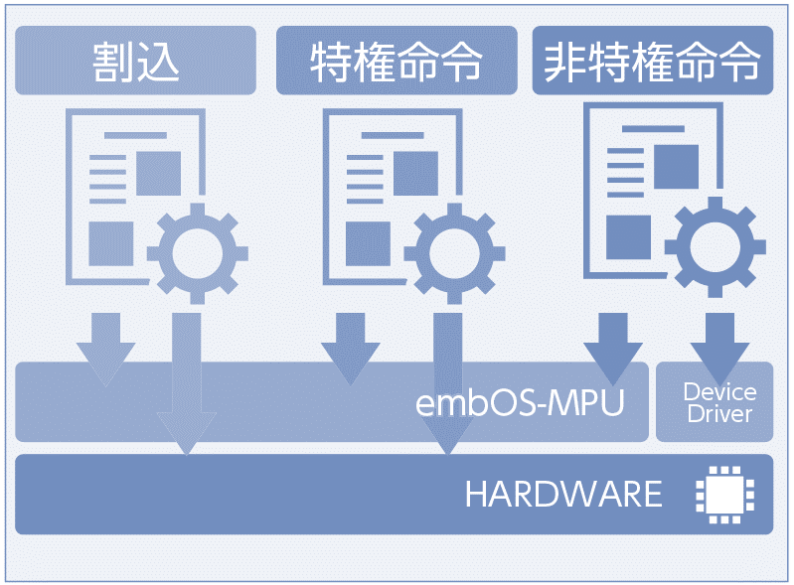
embOSパッケージ

- embOS-MPU
- embOS-Ultra
- embOS-Safe

お客様アプリケーション、仕様要件によってパッケージを選択可能

MPU機能を利用してセキュリティ強化、アプリケーションの安定性を向上

メモリ保護機能を持つことによりOSと特権タスクはメモリ保護され、非特権タスクの悪影響から隔離されます。embOSと互換性のあるAPIを実装ため、embOSで開発されたアプリケーションも最小の工数でembOS-MPUに適合させることができます。



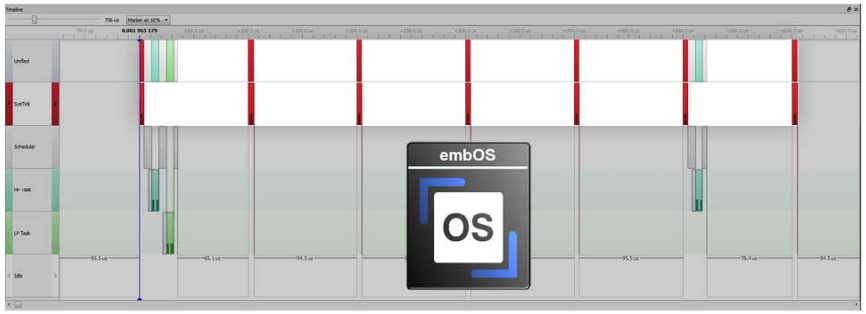
「特権命令」は特権状態で動作します。MPU設定の初期化タスクやデバイスドライバを含みます。特権命令を使用するタスクは、完全な信頼性を確認する必要があります。

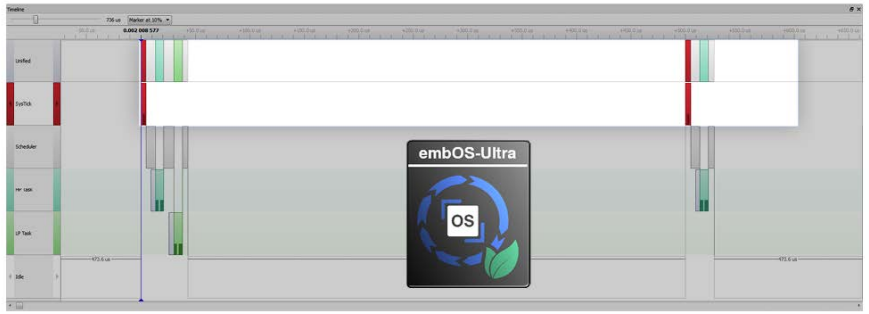
「非特権命令」アプリケーションは特権のない状態で実行されるため、不具合が発生した場合においてもメモリ保護機能により、基本システムに影響を与えることはありません。

　embOS-MPUは、ハードウェアのメモリ保護ユニット(MPU)と、embOS-MPUで実装されたソフトウェア機能により、1つのタスクがシステム全体に影響を与えないようにします。これにより、あるタスクでバグが発生した場合でも、他のタスクやオペレーティングシステムが実行を継続することができます。
　embOS-MPUでは、特権タスクはメモリにフルアクセスできます。非特権タスクは、それぞれの個別のメモリ領域に対し、特定のアクセス権限を持ちます。また周辺機器にアクセスするため、追加のメモリロケーションとOS制御構造、デバイスドライバ、特定のembOS APIなどを非特権タスクから呼び出すように設定も可能です。

高性能の分解能のタイミングを使用しパフォーマンスを向上させ、電力消費を削減する RTOS

サイクル分解能タイミングを使用して、市場の他の RTOS よりも優れた精度と時間分解能を提供します。すべての時間ベースのイベントのスケジュールをマイクロ秒または CPU サイクルで指定できます。「サイクル解決タイミングテクノロジー」は、正確なタイミングを実現するターゲット固有の技術を一貫した API 呼び出しに置き換えます。embOS-Ultra に切り替えるとすぐにパフォーマンスが向上し、電力が節約されます。





embOS-Ultraは一般的な 1 ミリ秒のシステム ティックを必要なときだけ正確に割り込みを生成するシングルショットハードウェア タイマーに置き換えます。この技術を使用すると従来のシステムティック割り込みが排除され、CPU アクティビティが減少し電力が節約されます。

embOS-Ultraを使用すると、タスクの遅延やソフトウェアタイマーなどの操作をシステムに依存しないマイクロ秒などの高解像度単位で指定できるようになりました。これによりRTOS の潜在的なアプリケーションの範囲が大幅に拡大します。

サポートしているCPUコア：Cortex-A/R/Mコア・RISC-V

その他のCPUコアへの対応はご相談ください。

機能安全認証RTOS

各製品分野に対応した機能安全認証を取得しています。

産業機器
IEC61508 SIL3

医療機器
IEC62304 ClassC

車載
ISO26262 ASIL D

家電製品
IEC61508 SIL3

通常embOSと共通化

embOSとAPIは共通化されていますので、
embOS用に開発されたプロジェクトをそのままリユースができます。

embOS機能安全マニュアル・認証キット

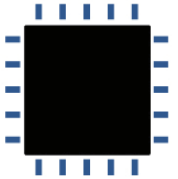
embOS機能安全マニュアルを含むドキュメントがパッケージされています。

embOSライセンスモデル

ニーズに合わせて選択可能なライセンスモデル

SEGGER社ソフトウェアライセンスモデル

永久ライセンス・量産ロイヤリティなしで継続的な費用は必須ではありません。

	シングルプロダクト	プロダクトファミリ （個別提案）	シングルデベロッパ （ユーザ）	CPU （個別提案）
				
開発可能製品数	1製品	1製品ファミリ	無制限	無制限
利用可能開発者数	無制限	無制限	1名	無制限
CPU	1CPU型番	1CPU型番	1CPUアーキテクチャ	1CPUアーキテクチャ
コンパイラ	1種類	1種類	1種類	1種類

多数の開発者で1つの製品を開発する。
プロジェクト単位で予算計上

複数の開発プロジェクトで共通利用
開発プラットフォーム化に最適

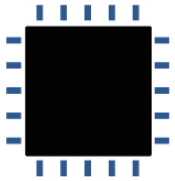
開発プロジェクトは無制限／開発者人数に応じたライセンス

シングルデベロッパ（ユーザ）	開発可能製品数	利用可能開発者数	CPU	コンパイラ
	無制限	1名	1CPUアーキテクチャ	1種類

「シングルデベロッパライセンス」は開発プロジェクトに制限されず、無制限に製品開発が可能です。開発者様が複数の開発プロジェクトを担当するなど、多品種開発に最適なライセンスです。

CPUアーキテクチャが同じCPUであれば、製品毎のCPU変更（デバイスメーカ変更）も対応可能です。

CPU	開発可能製品数	利用可能開発者数	CPU	コンパイラ
	無制限	無制限	1CPUアーキテクチャ	1種類



「CPUライセンス」は同一CPUアーキテクチャのCPUで複数の開発プロジェクト、開発者の人数に係わらず利用可能です。本ライセンスにより、SEGGER社製RTOS/ミドルウェアを含むソースコードを企業内で、共有ができます。御社内のソフトウェアプラットフォーム化に最適なライセンスです。
本ライセンスは、すべてお客様のご要望に従い都度提案となりますので、必ずしもCPUの制限事項が１CPUアーキテクチャになるわけではなく、ご要望に応じたライセンス提案をさせていただいています。

大規模な開発プロジェクトに最適なシングルプロダクト／プロダクトファミリライセンス

開発者の人数は無制限（外部協力会社含む）で特定の製品開発に利用可能なライセンス

シングルプロダクト	開発可能製品数	利用可能開発者数	CPU	コンパイラ
	1製品型番	無制限	1デバイス型番	1種類

複数の開発者で１つの製品（製品型番）開発が可能です。開発者様が多い大規模開発や品種展開を想定しない製品開発に最適。
製品メーカ様へのライセンスで、該当製品開発に係わる開発者は本ライセンスで利用可能です。受託開発で利用検討の場合は、ライセンス契約者として、受託元様での契約をお願いいたします。
例）「J-Link BASE」で契約し、「J-Link BASE」を開発する。

プロダクトファミリ	開発可能製品数	利用可能開発者数	CPU	コンパイラ
	1製品ファミリ	無制限	1デバイス型番	1種類

「プロダクトライセンス」の適用範囲を広げて、１製品シリーズの開発が可能です。開発者様が多い大規模開発で、派生製品開発を行う場合に最適となります。プロダクトファミリの定義は、お客様の要望に応じて、都度SEGGER社と協議の上、ライセンス費用提示となります。

例）「J-Linkシリーズ」で契約し、「J-Link BASE」「J-Link PLUS」「J-Link PRO」を開発する。
※適用範囲について、適宜ご相談ください。

提供会社

EmbiTeK | SEGGER

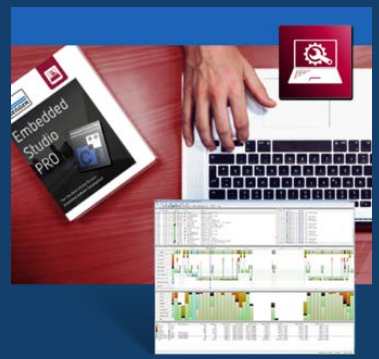
SEGGER Microcontroller GmbH

組込みシステムで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

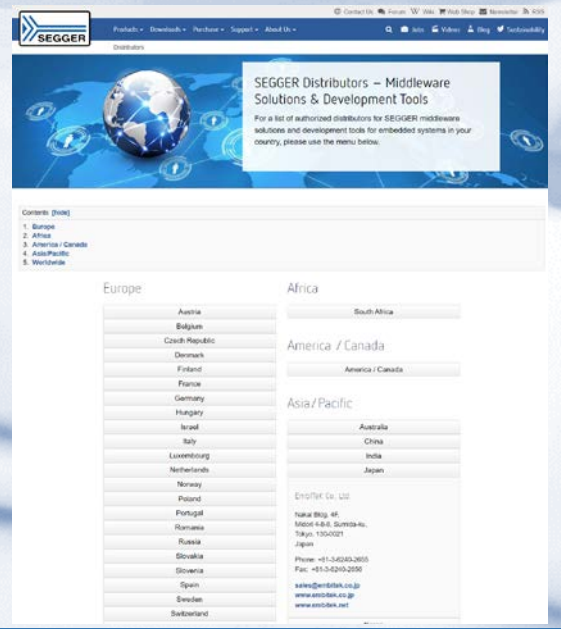

デバッグツール


書込みツール

EmbiTeK

EmbiTeK

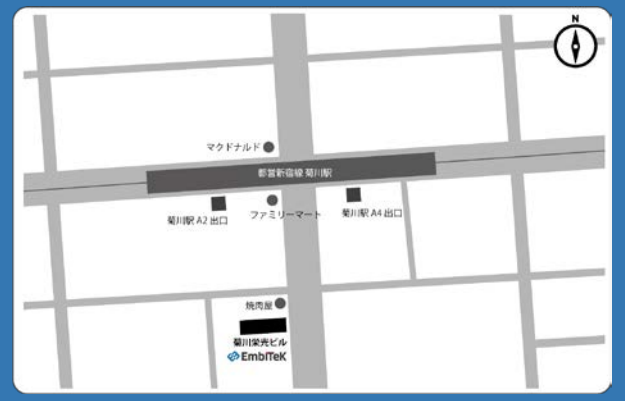
株式会社エンビテック



お客様の要件に合わせ、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 パワル
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

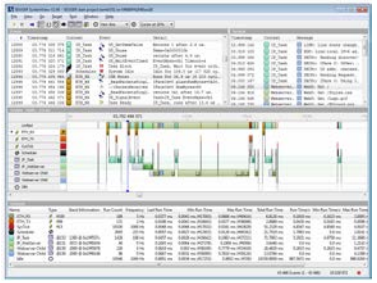
Arm Cortex/RXソフトウェア開発から量産をサポート

製品開発フローの課題に合わせて対応

SEGGER
www.segger.com

EmbiTeK

デバッガ
開発ツール



RTOS

embOS

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

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	SNTP client	NTP client
PTP OC client	TCP	UDP
Web Socket client	Web server	UPnP
Web Socket server	PPP/PPPoE	Wifi support

USB Host

HID	MTP
MassStorage	CDC
Printer	FTDI
LAN	MIDI
Audio	HUB
CCID	CP21xx UART
Video	

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

EmbiTeK

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/

YouTube
http://www.youtube.com/@embitek