embOS-Ultra

8/16/32bit マイコン対応リアルタイムOS「エンボス」

消費電力削減

2022/12/20

EmbiTeK

embOS-Ultraを利用することでJ-Link消費電力削減

製品特長

J-Linkは、ファームウェアにembOS-Ultraを採用することで
スタンバイモードとエコモードを新たに追加しました。

J-Linkは、非アクティブ時間を検出すると自動的にスタンバイモードに移行し、内部クロック周波数が低下、LEDが暗くなります。
周期ティックを使用しないembOS-Ultraを利用することで、CPUの負荷を軽減し、省エネ効果を最大限に発揮します。

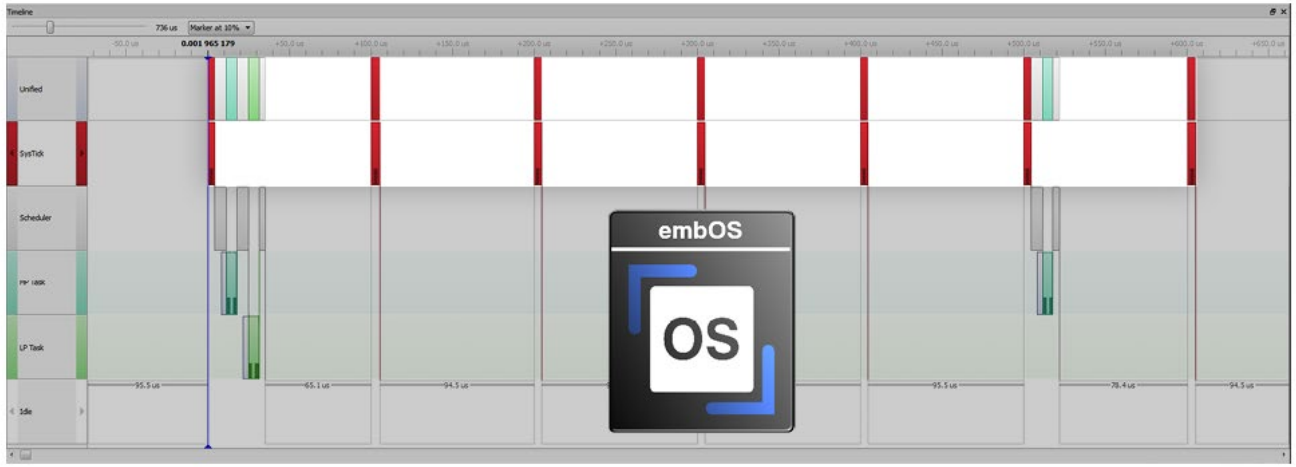
エコモードは、J-Linkコンフィギュレータで設定し、LEDの輝度、クロック周波数の低減により、パフォーマンス上、影響を受けない範囲で省エネを実現しています。

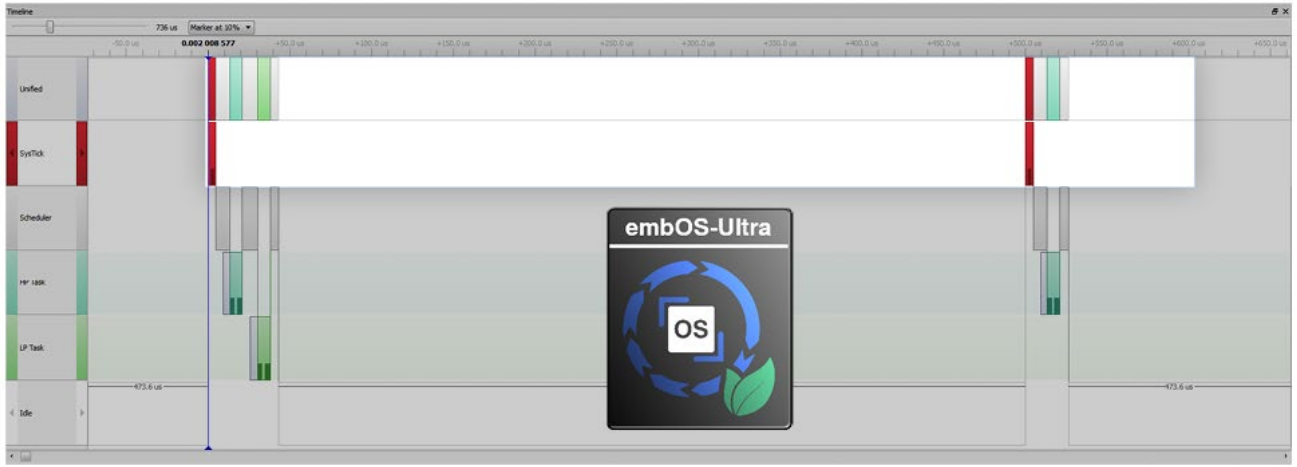
J-Linkは、元々少ない電力で動作していますが、スタンバイモードとエコモードの実装により、無駄な電力を削減し、環境負荷を下げるよう務めています。

Cycle-resolution Timing テクノロジー

製品特長

他のRTOSでは実現できない精度と時間分解能を提供します。





タイムアウト、遅延、定期タイマーなど、すべての時間ベースのイベントのスケジューリングを、マイクロ秒または **CPU** サイクルで指定

通常の **1** ミリ秒単位のシステム ティックを、必要なときに正確に割り込みを生成する ハードウェア タイマーに置き換えることができます。この手法により、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、省電力なシステムを構築することができます。

embOSアプリケーションと100%互換性

製品特長

従来のembOSで開発されたアプリケーションは、そのまま利用可能

既存の **API** と **RTOS** の動作が維持されるため、アプリケーションの変更は必要ありません。**embOS-Ultra** は、従来の **embOS API** 呼び出しが使用されるミリ秒単位のタイミングを提供し、新しい **API** 呼び出しが使用されるマイクロ秒またはサイクルの解像度を提供します。

従来の **embOS API** は、拡張された高精度 **embOS-Ultra API** と同じアプリケーション内で混在させることができます。どちらかを選択する必要はありません。

embOS概要

多くの市場実績
汎用性、信頼性の高いRTOS
メモリ保護機能対応 – embOS MPU
機能安全対応 – embOS-Safe

小さなフットプリントで、汎用性の高いRTOSソリューション

SEGGER J-Linkで利用されているソフトウェア

累計100万台以上の販売実績を持つJ-Linkシリーズ

embOSは、J-Link・J-Trace,Flasherシリーズの基幹ソフトウェアとして、25年以上の実績を持つRTOSです。

embOSユーザメリット

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

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

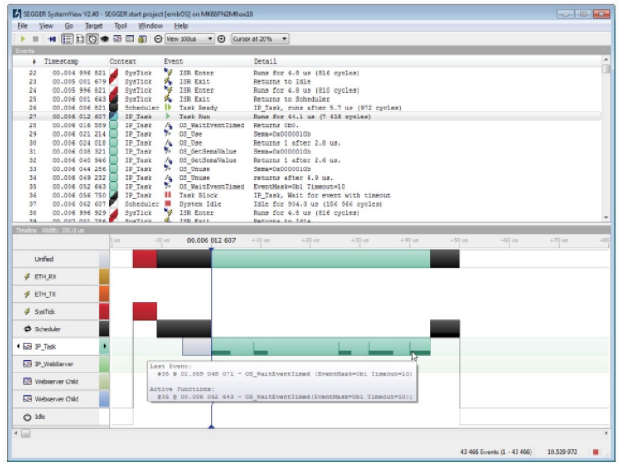
・RTOSプロファイル機能を標準実装
・SystemViewツールでRTOS機能のリアルタイム診断可能

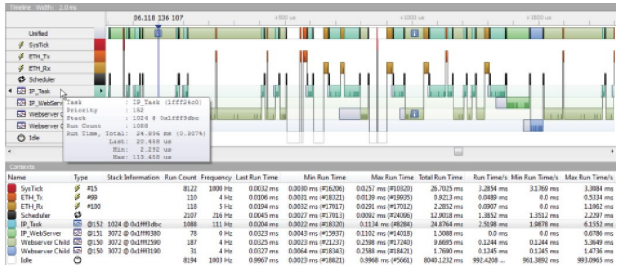
習得しやすいサンプルソースコード

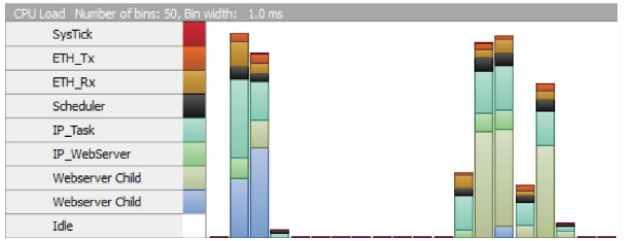
豊富な標準サンプルソースコードで利用方法の習得がしやすくなっています。

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







SystemView

embOSの実績

25年以上に渡り、欧州・米国を中心に利用実績を積んだ組込用RTOS

欧州・米国を中心に1,000ライセンス以上の販売実績

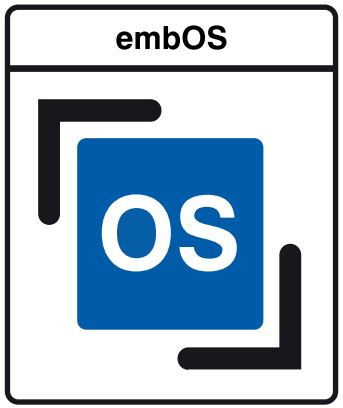
10億台以上の量産品出荷（SEGGER社調べ）

「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。

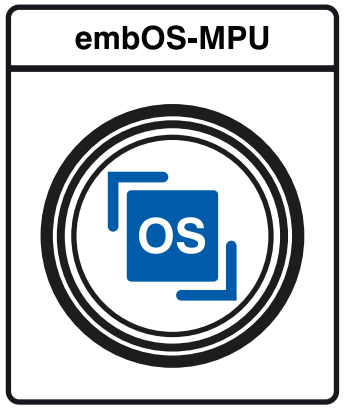
embOSライセンスバリエーション

ロイヤリティフリー・開発製品無制限のユーザライセンスで提供可能

ソースコード提供ライセンス



embOS基本パッケージ

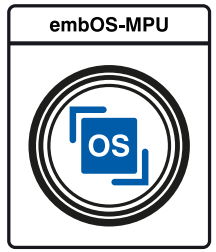


メモリ保護機能

機能安全認証

オブジェクト提供ライセンス

オブジェクト版

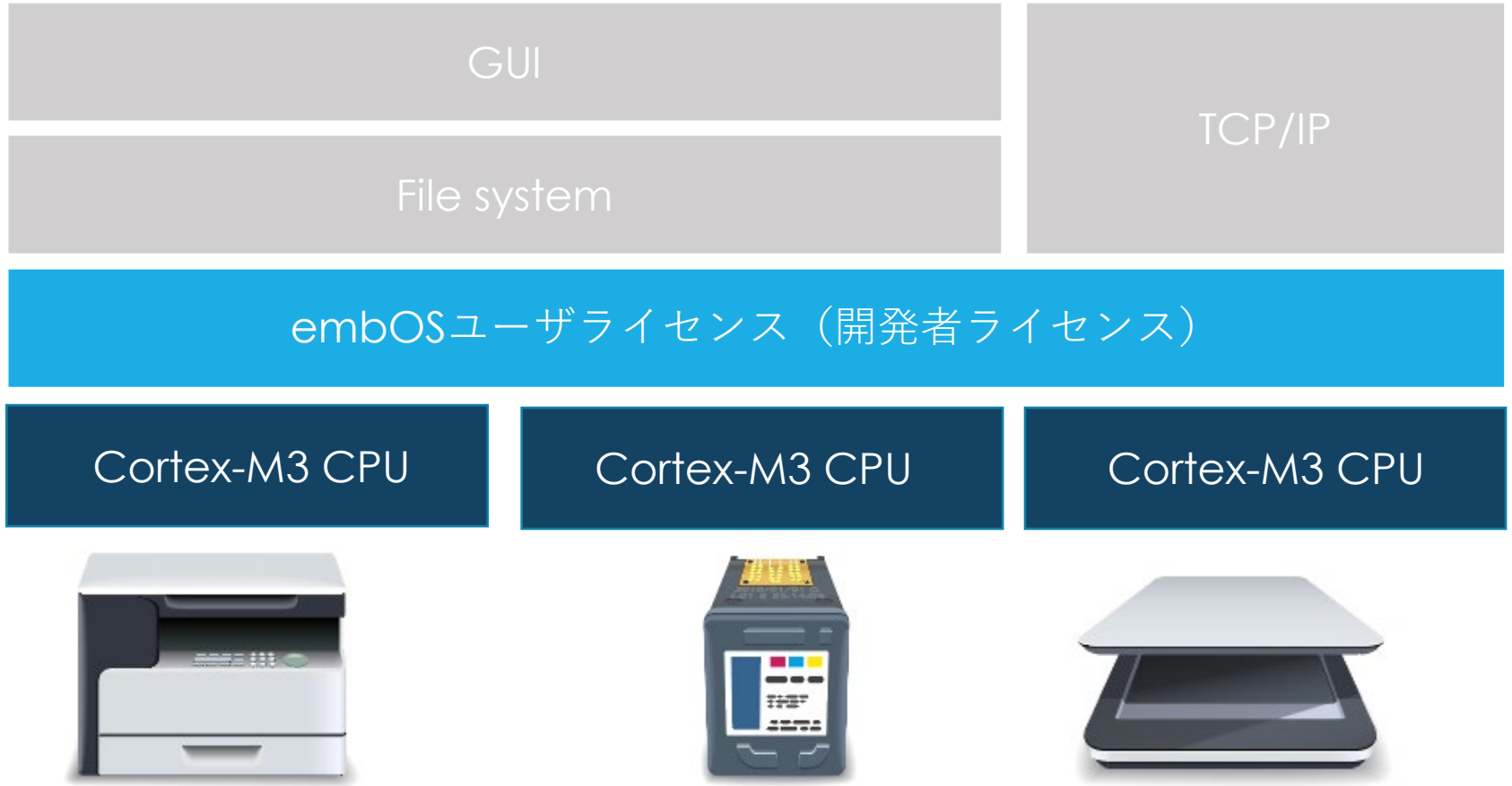


オブジェクト版

長期間の実績と信頼性が実現するオブジェクト版ローコストライセンス

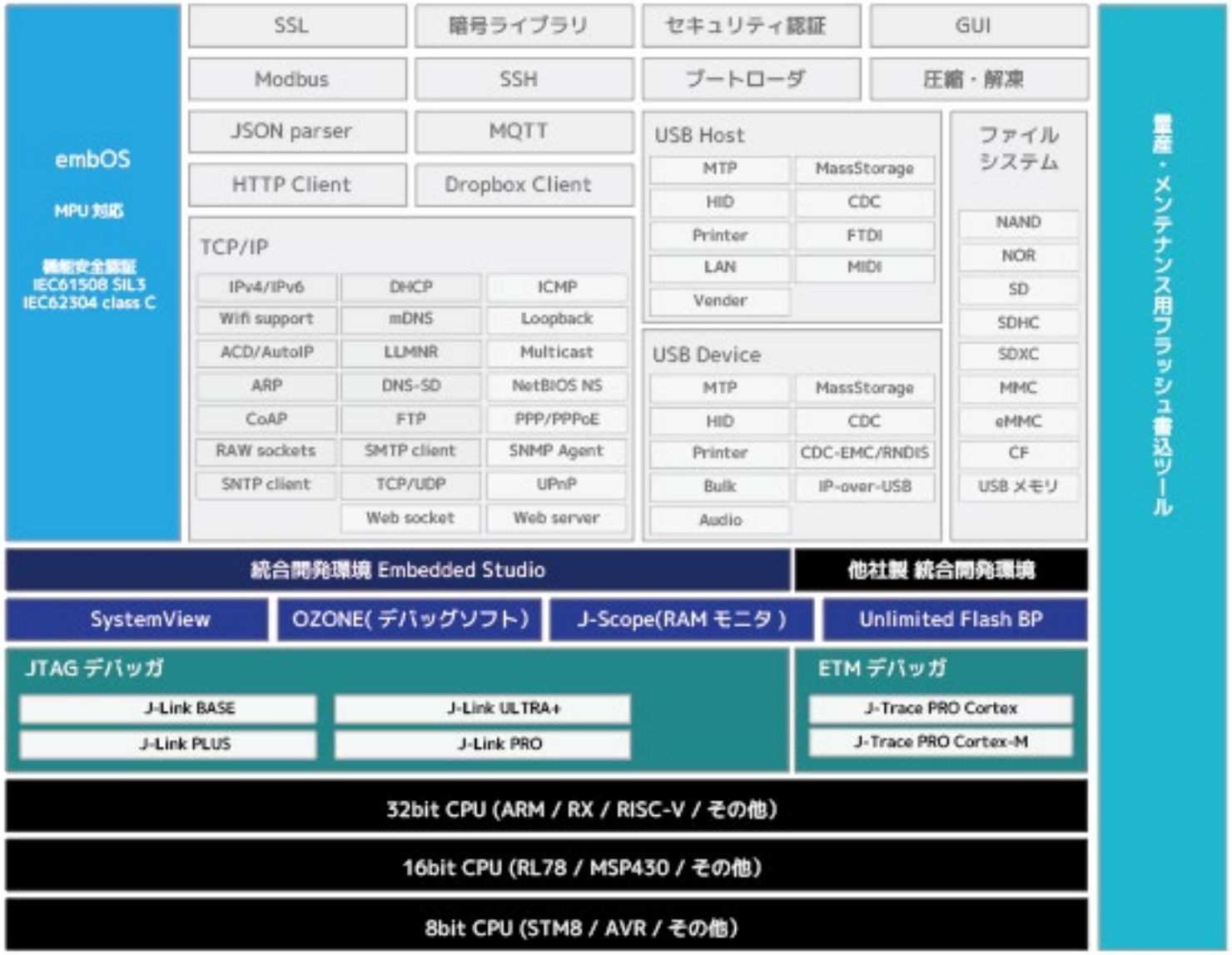
embOSのユーザメリット

ロイヤリティフリー・開発製品無制限のユーザライセンスで提供可能



開発プロジェクト無制限
様々な開発で利用可能
マイコン変更も対応
（Cortex-M3ライセンス）

embOSは数多くのミドルウェアをサポート

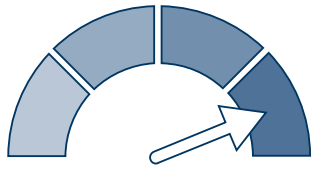


数多くのソフトウェアモジュールを提供

御社のソフトウェア開発のベースとして、
利用可能です。

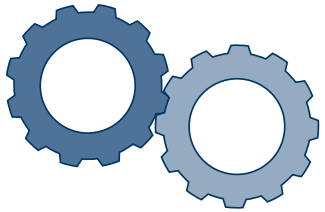
embOSの利点

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



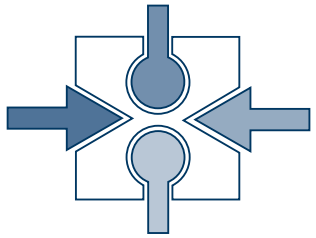
効率性

低いROM/RAM使用量にもかかわらず、
高いリアルタイム性能を実現します。



信頼性

数十億の量産デバイスに搭載出荷実績を持ち、
機能安全規格に準拠しています。



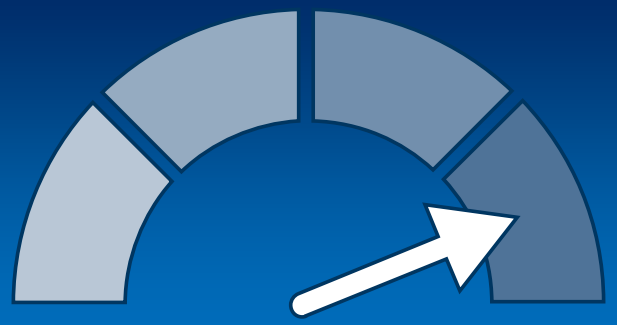
汎用性

幅広いコア・コンパイラ対応で、
マイコンやツールに依存しない資産を構築

使いやすさ

分かりやすいAPI、開発を支援するツールなど。
開発ユーザ視点に立った使いやすさを実現。

embOSの効率性



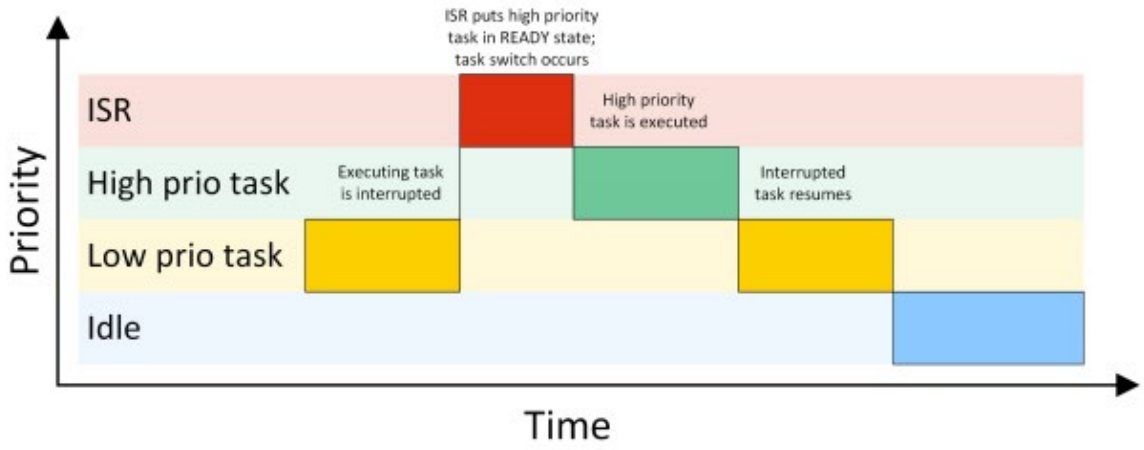
マルチタスク
省ROM/RAMリソース

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。

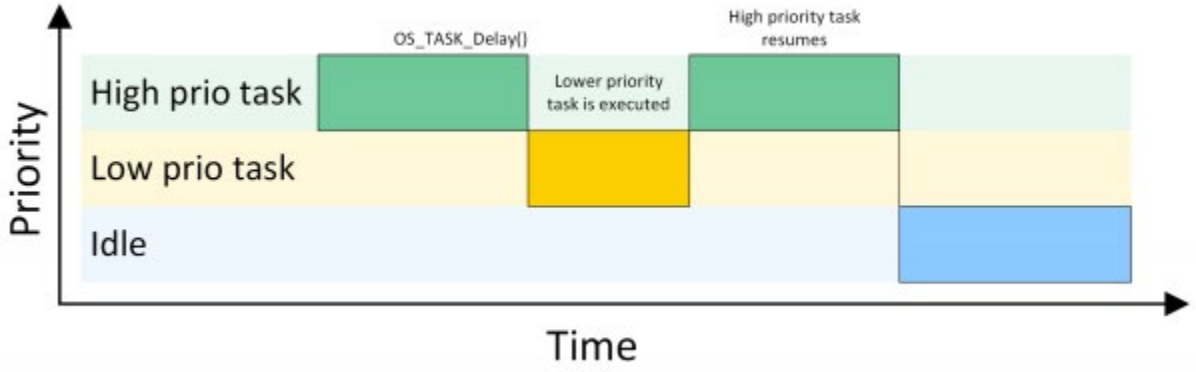
embOSの効率性：マルチタスクシステム

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

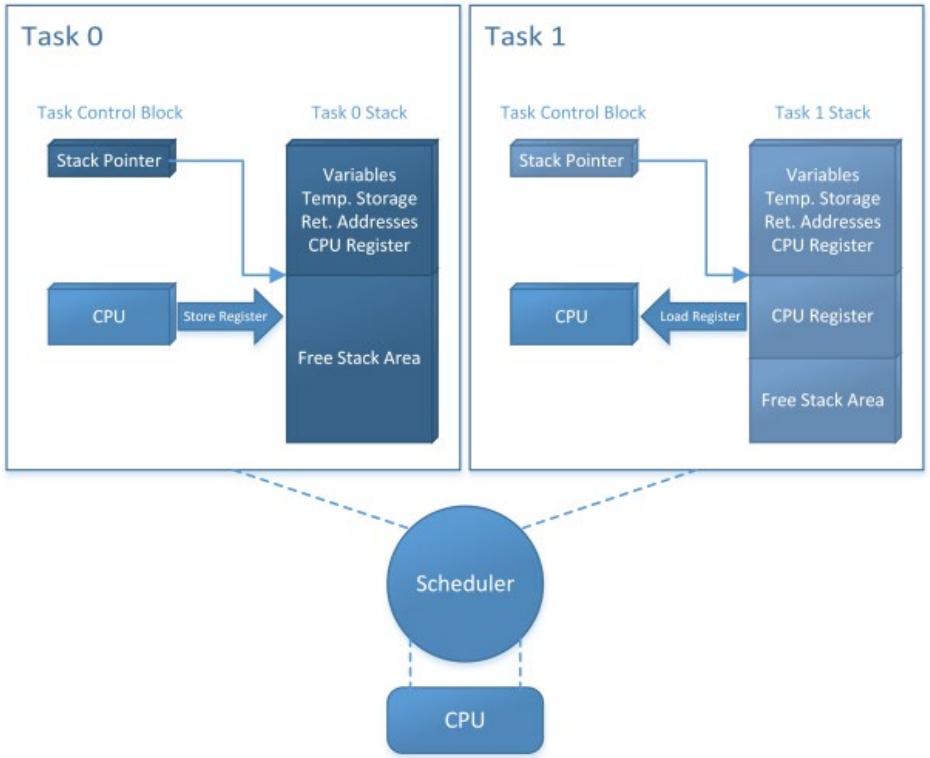
■ プリエンプティブマルチタスク



■ 非プリエンプティブマルチタスク

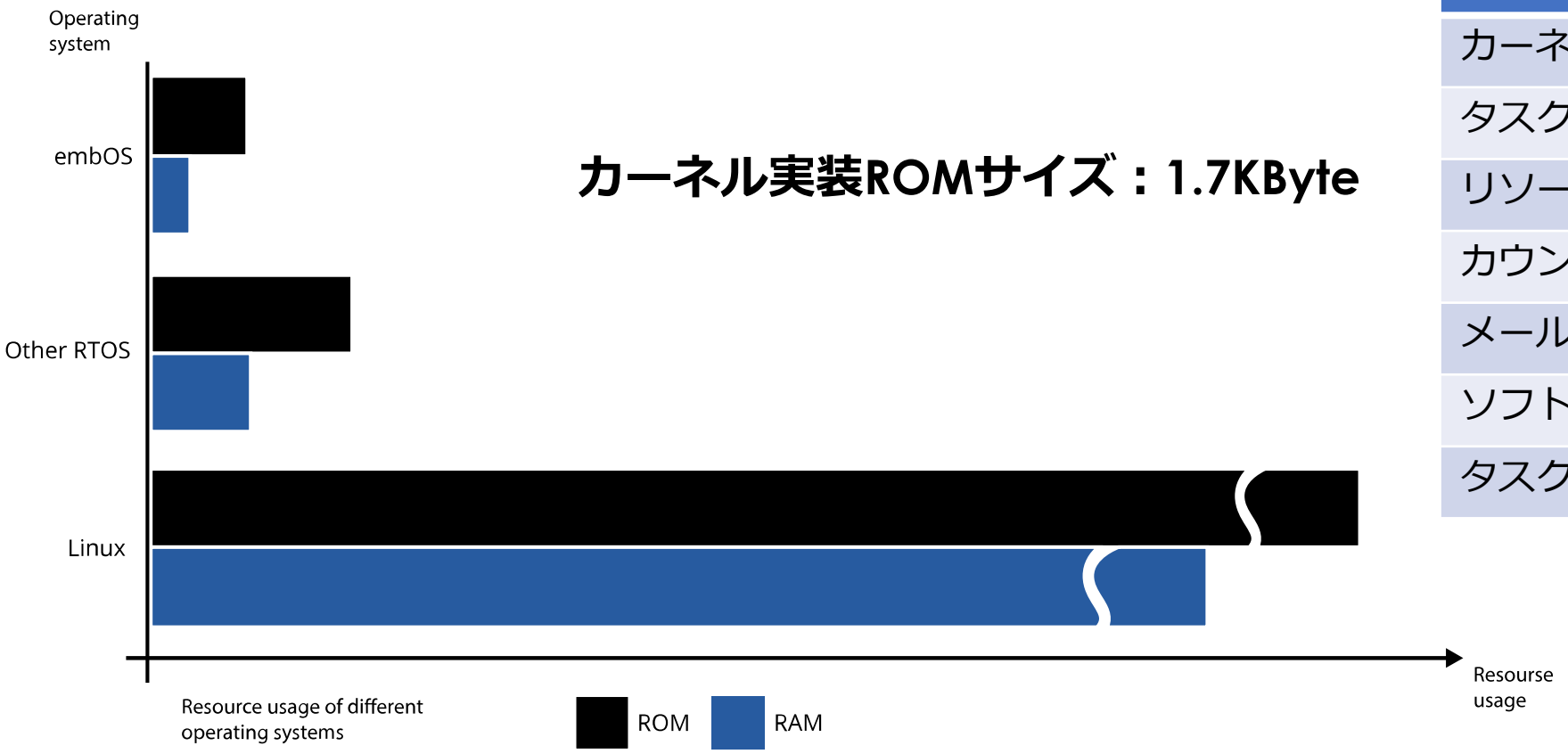


ラウンドロビン方式のスケジューリングや
プライオリティでコントロールするアルゴリズムと
小型のRTOSながら、高性能なRTOS機構を持ちます。



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です。



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)

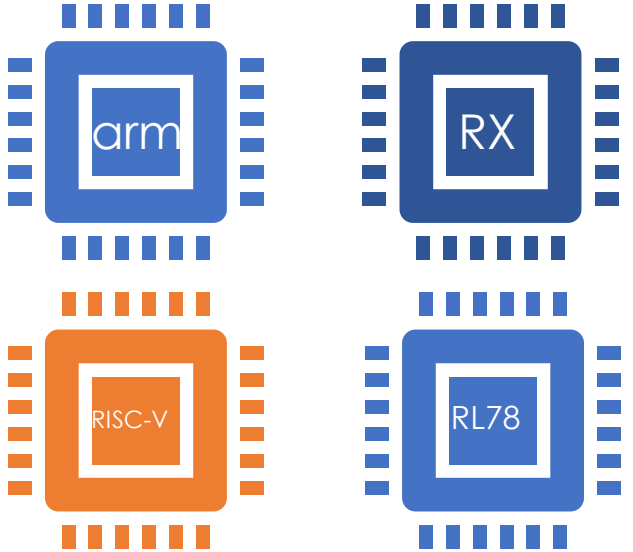
embOSの汎用性

マルチタスク
省ROM/RAMリソース

ユーザ様の環境に合わせて利用できるRTOS。

embOSの汎用性

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
8/16/32bitマイコンを一つのRTOSでカバーしています。



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

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

対応マイコンリスト

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
IDT	Cortex-M	ZAMC4100
Infineon	Cortex-M	TLE9877, XMC4300, XMC4500, XMC4700, XMC4800
	C16x	C16x
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,
Nordic	Cortex-M	nRF51xxx, nRF52xxx
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-M	STM32F0xx, STM32F1xx, STM32F2xx, STM32F3xx, STM32F4xx, STM32F7xx, STM32H7xx, STM32L0xx, STM32G0xx, STM32L1xx, STM32L4xx, STM32W10x, STM32WB
	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
Xilinx	Cortex-A	XC7Z007S, XC7Z010, XC7Z020
	Cortex-R	XCZU3EG

embOSの汎用性：マルチコア・マルチOS

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。

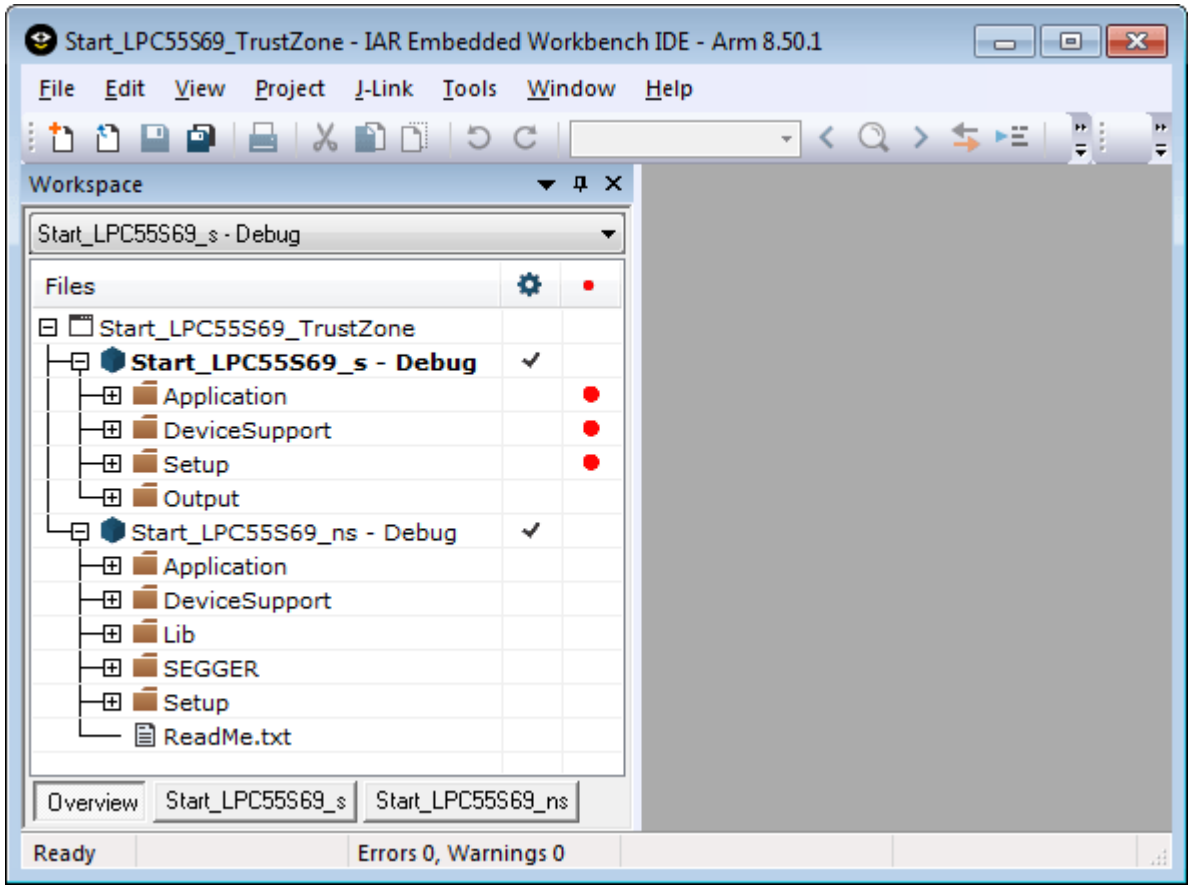
これにより、共有メモリ・周辺機器などへのアクセスを実現します。

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。

Cortex-Mセキュア機能「Arm v8-M TrustZone」対応

製品特長

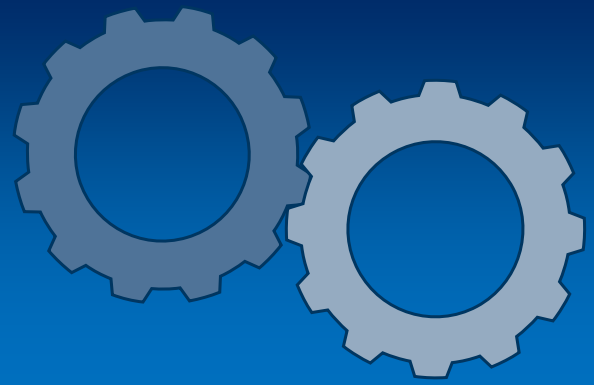
セキュアエリアと非セキュアエリアを分離サポート対応



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。

embOSの信頼性



機能安全認証にも対応可能な高品質ソースコード

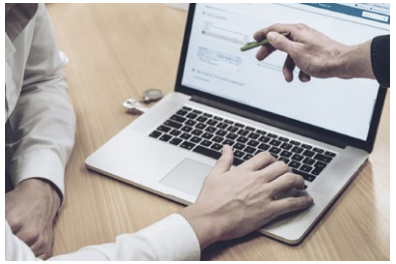
25年以上の利用されているRTOSソリューション

embOSの信頼性

embOSは、様々な領域の製品で安心して利用することができます。

機能安全認証対応（embOS-safe)
IEC 61508 SIL 3 / IEC 62304 Class C認証取得済みRTOS

数十億台の量産製品出荷実績
リリース以来、25年にわたりお客様製品に搭載され量産出荷されています。



MISRA-C2012準拠のオリジナルコード
オープンソースコードは一切利用せず、SEGGER社のembOS開発チームにより開発されたオリジナルRTOS。

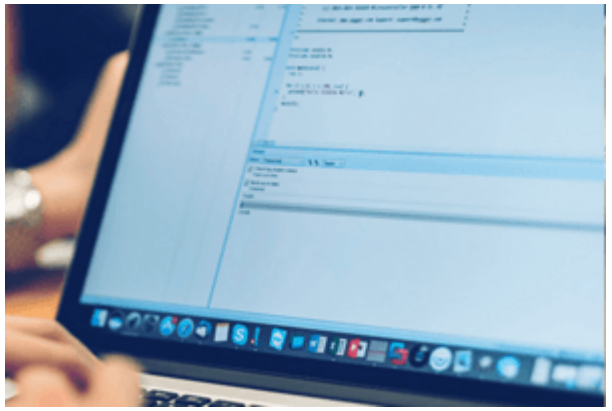
embOSの使いやすさ

充実した開発支援ツール
サポート体制

開発しやすいRTOSソリューションで、開発時間を短縮。
様々な開発支援ツールを利用できます。

使いやすいRTOSソリューション

embOSは、様々な領域の製品で安心して利用することができます。



洗練されたソースコード
見やすく分かりやすい
ソースコード提供

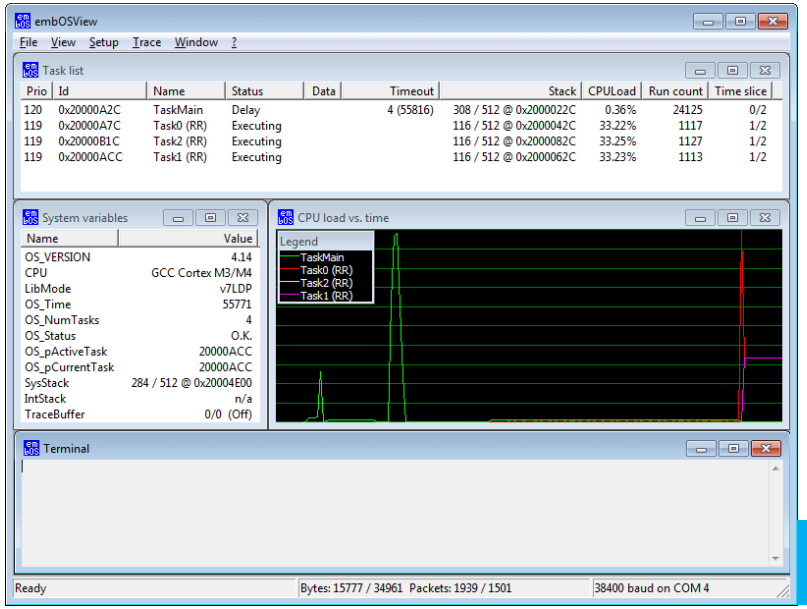
分かりやすいAPI
扱いやすいAPIを提供
初めてでも使いやすく

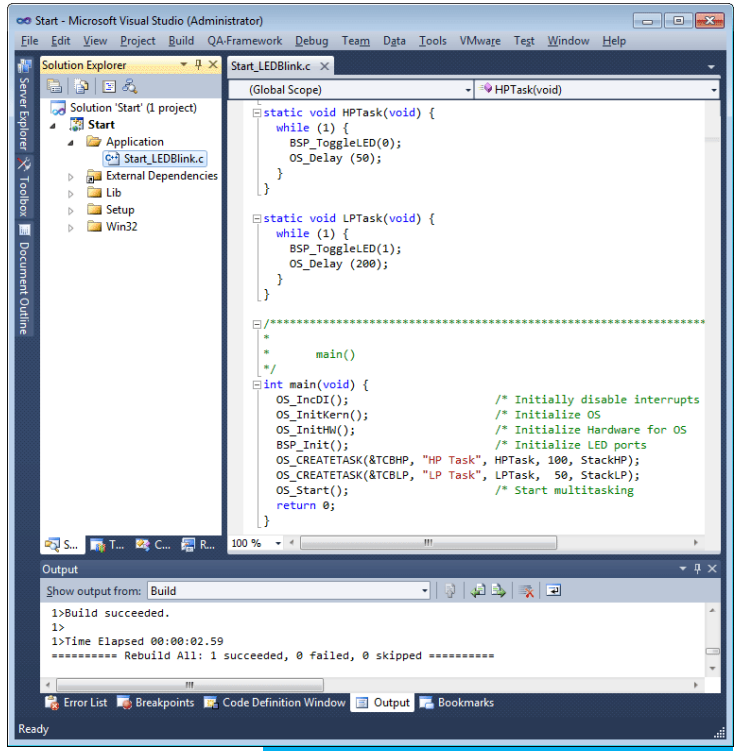
BSPサポート
500以上の評価ボードへ
BSPを提供しています。

日本語サポート
当社より日本語サポートを
提供
英語でSEGGER社開発者から直接
サポートを受けることも可能

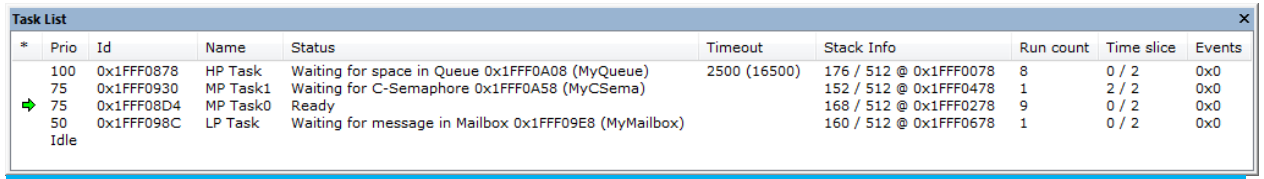
embOS開発支援ツール

ソフトウェア開発を支援する様々な無償提供ツール群





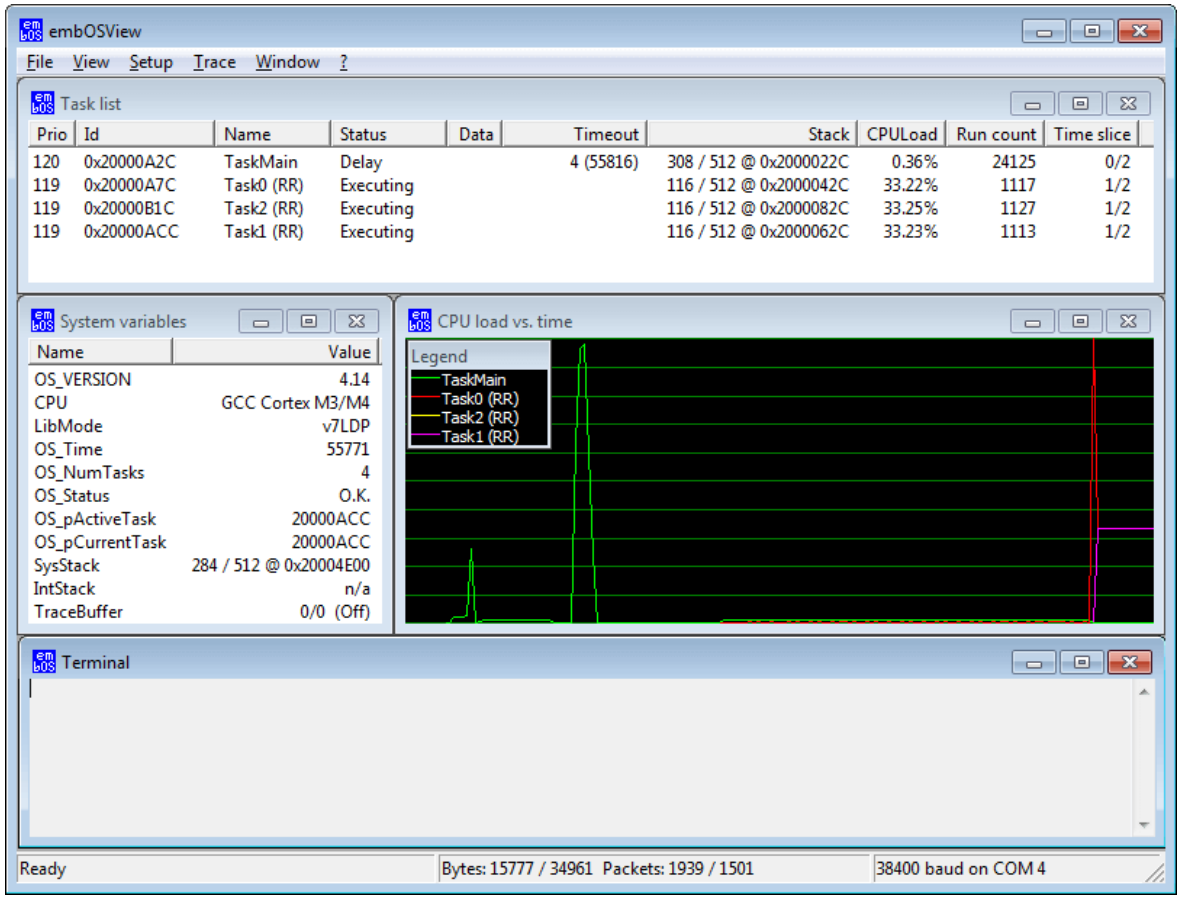
embOS
シミュレーション



SEGGER Embedded Studio Plug-in / EW Plug-in

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



コンパイラやIDEに依存しないため、
お客様の環境で利用することができます。

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 View

embOS Viewでは、ターゲットからリアルタイムに情報取得

Name	Description
Prio	現在のプライオリティタスク
Id	タスクID, タスクコントロールブロックのアドレス
Name	作成名
Status	現在のタスク状況 (ready, executing, delay, reason for suspension)
Data	ステータス状況
Timeout	次の立ち上がり時間
Stack	利用スタックサイズ/最大スタックサイズ/スタックポインタ
CPULoad	タスクによるCPU利用率
Run Count	リセットからの回数
Time slice	ラウンドロビン情報
OS_Version	embOSバージョン

CPU	ターゲットCPU,コンパイラ
LibMode	ターゲットアプリケーションに利用されるライブラリモード
OS_Time	タイマーティックからの現在時間
OS_NumTasks	定義されたタスクの数
OS_Status	現在のOS状況
OS_pActiveTask	実行されるべきアクティブタスク
OS_pCurrentTask	現在の実行タスク
SysStack	利用サイズ、最大サイズ、システムスタックポインタ
IntStack	利用サイズ、最大サイズ、割込スタックポインタ
TraceBuffer	トレースバッファの状況（サイズ・システムカウンタ）

embOS Plug-in for IDE

各種IDE上でembOSの状況を表示

Task List

*	Prio	Id	Name	Status	Timeout	Stack Info	Run count	Time slice	Events
	100	0x1FFF0878	HP Task	Waiting for space in Queue 0x1FFF0A08 (MyQueue)	2500 (16500)	176 / 512 @ 0x1FFF0078	8	0 / 2	0x0
	75	0x1FFF0930	MP Task1	Waiting for C-Semaphore 0x1FFF0A58 (MyCSema)		152 / 512 @ 0x1FFF0478	1	2 / 2	0x0
➪	75	0x1FFF08D4	MP Task0	Ready		168 / 512 @ 0x1FFF0278	9	0 / 2	0x0
	50	0x1FFF098C	LP Task	Waiting for message in Mailbox 0x1FFF09E8 (MyMailbox)		160 / 512 @ 0x1FFF0678	1	0 / 2	0x0
	Idle								

タスクリストで、各タスクレベルの状態を表示します。

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	

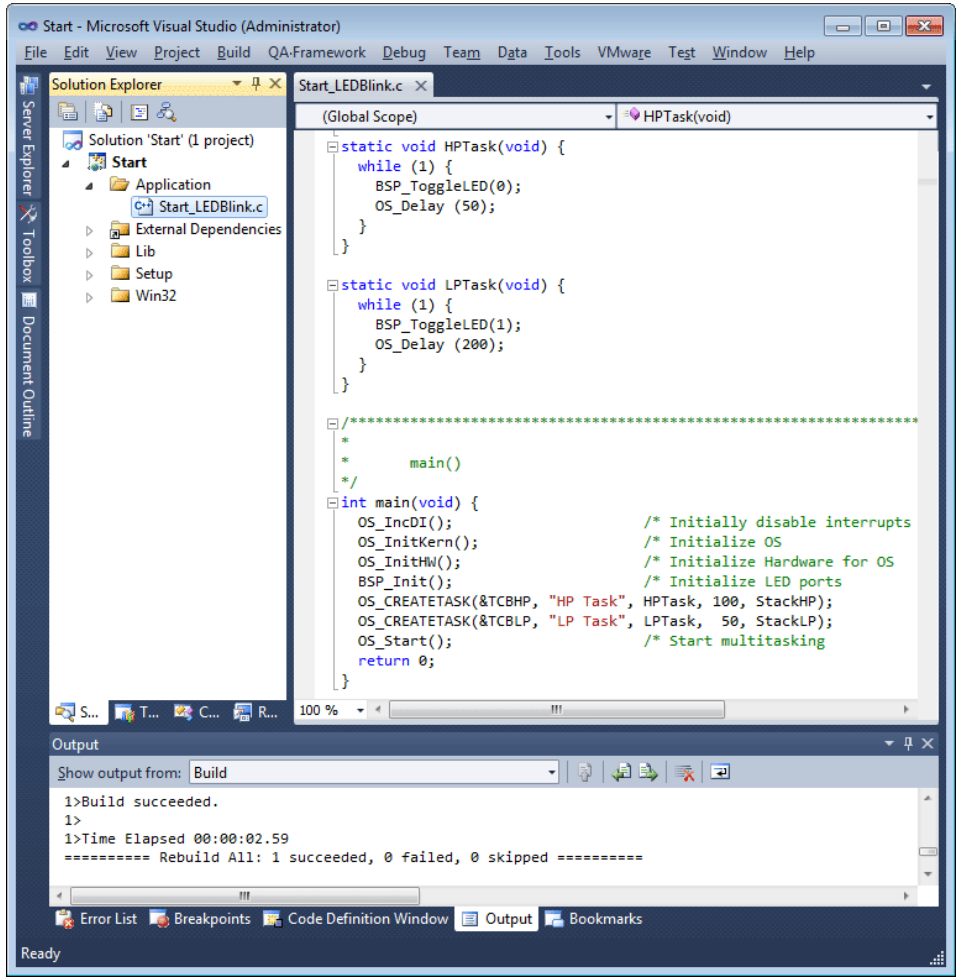
Id(Semaphores)	Name	Count	Waiting Tasks
0x200013B8	Semaphore 0	0	0x20000A20 (Background Task 1)

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

SEGGER EmbeddedStudioでは
標準実装されていますので、
簡単に利用可能です。

embOS シミュレーション

VisualStudio等でソフトウェアシミュレーション



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出来ます。

embOS が標準で持つプロファインリングAPI

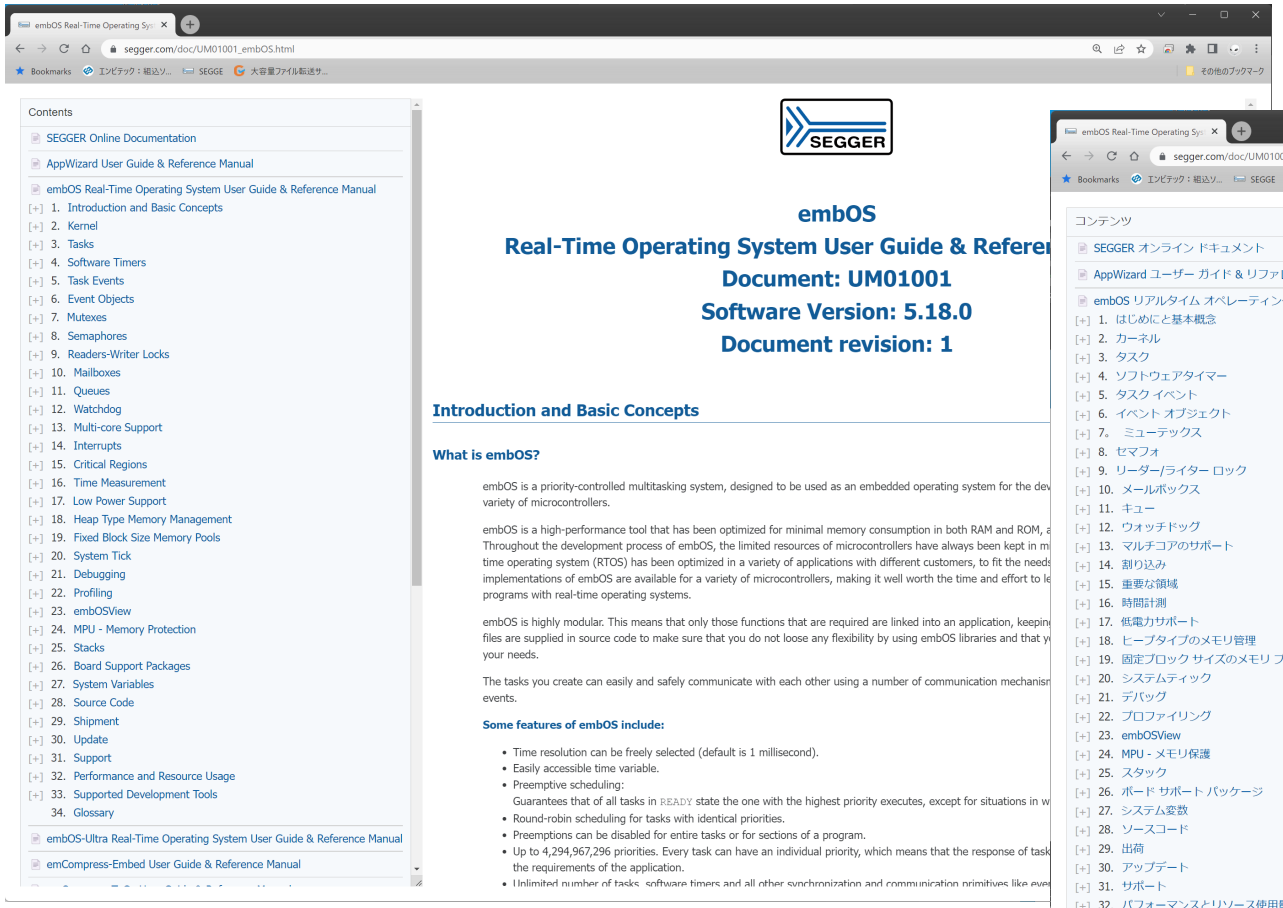
embOSでは、OSプロファイル・トレース機能を標準搭載

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。

HTMLベースのマニュアル

最新のユーザマニュアルをHTML形式で公開

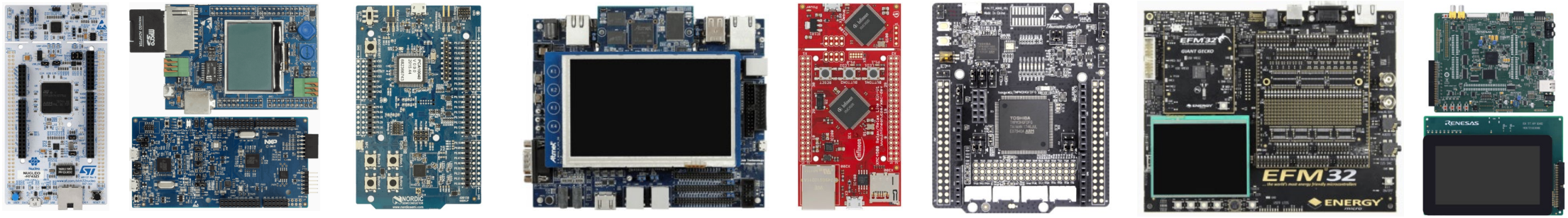


ブラウザの翻訳機能で日本語表示可能

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

評価ボード・BSP

多くのCPU・評価ボード用のBSPを用意オブジェクト評価版を提供



評価ボード一覧：
https://www.embitek.co.jp/download/ps/EVAL_SW.pdf

無償評価版：すべてのAPIを評価利用頂くことが可能です。
https://www.segger.com/downloads/embos/

ポーティング・受託開発

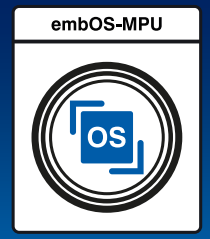
エンビテックでは、お客様導入を受託対応でサポート

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。



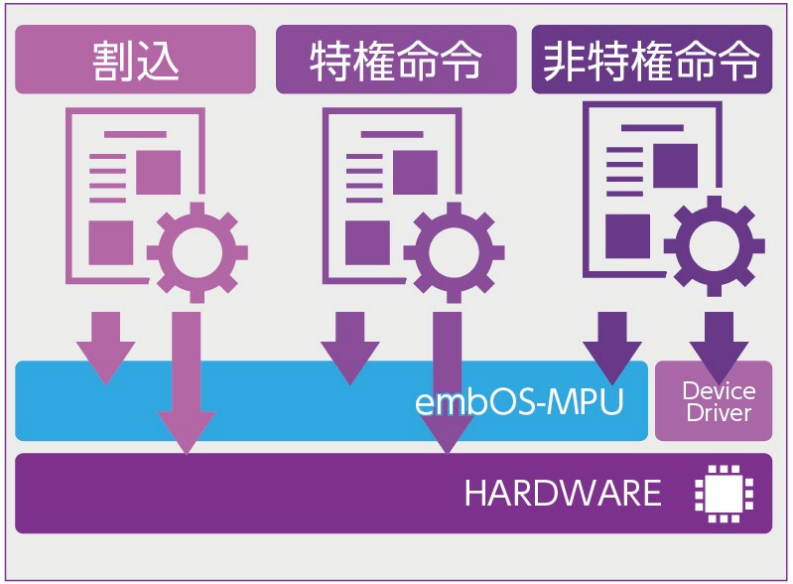
embOS-MPU

メモリ保護機能付きRTOS

embOS MPU

Cortex-Aなどのハイエンドマイコンで実装されているMPU機能に対応

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。

embOS MPUサンプルコード

embOS MPUサンプルコードで利用方法を確認可能

embOS-MPU will terminate the LPTask as soon as the task tries to write to memory outside its task stack. Additionally the error callback function is called. The error callback function prints a message with the task id and the error cause. The HPTask and the OS are not affected and still run.

```
1  /*********************************************************************
2  *
3  *       _ErrorCallback()
4  *
5  *  Function description
6  *    User callback function which is called when an unprivileged task
7  *    does something disallowed, e.g. tries to write to memory which
8  *    does not belong to the task
9  */
10 static void _ErrorCallback(OS_TASK* pTask, OS_MPU_ERRORCODE ErrorCode) {
11   static const char* _sErrTxt[] = {  "OS_MPU_ERROR_INVALID_REGION", "OS_MPU_ERROR_IN
12                                      "OS_MPU_ERROR_INVALID_API", "OS_MPU_ERROR_HARDF
13                                      "OS_MPU_ERROR_MEMFAULT", "OS_MPU_ERROR_BUSFAULT
14                                      "OS_MPU_ERROR_USAGEFAULT", "OS_MPU_ERROR_SVC"};
15   printf("Task with ID 0x%x has been stopped due to error %s\n", (OS_U32)pTask, _sEr
16 }
```

https://www.segger.com/products/rtos/embos/editions/embos-mpu/embos-mpu-sample-application/

embOS-Safe

IEC 61508 SIL 3 / IEC 62304 Class C認証取得済みRTOS

機能安全認証対応RTOS

embOS-safe

RTOSは、安全性を重視したアプリケーションで最も重要なコンポーネントです。

embOS-safeを利用する事により、安全性が重要なコードを分離し、他のタスクと干渉することなく動作出来るようにする事が可能です。これにより、アプリケーションの機能安全対応をより簡素に行う事が出来ます。

embOS-safeは、産業・医療・自動車・家電などの安全分野向けに設計されています。

産業機器
IEC61508 SIL3

医療機器
IEC62304 ClassC

車載
ISO26262(お問合せ)

家電製品
IEC61508 SIL3

embOS-safeは、お客様製品の機能安全認証取得をサポート

機能安全認証対応（embOS-safe)
IEC 61508 SIL 3 / IEC 62304 Class C認証取得済みRTOS

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。

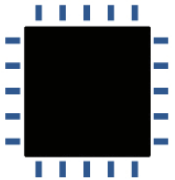
ライセンス

量産ロイヤリティ不要・開発ライセンス

embOSの提供ライセンス体系

使いやすいライセンス

ユーザ様のニーズに合わせて、選べるライセンス

	プロダクト	プロダクトファミリ	ユーザ	CPU
				
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

すべてソースコード提供・量産ロイヤリティ不要

無償６ヶ月保守含む／以降は任意で保守継続。保守がなくてもソフトウェアの開発利用／量産は可能

大規模開発に最適なプロダクト・プロダクトファミリ

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 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 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。

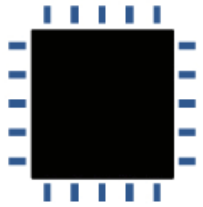
例）「J-Link シリーズ」で契約し、「J-Link BASE」「J-Link PLUS」「J-Link PRO」を開発する。
※適用範囲について、適宜ご相談ください。

開発プラットフォームとして最適なユーザ・CPU

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	1 CPU アーキテクチャ	1種類

「ユーザライセンス」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	1 CPU 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。

お問合せ窓口

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。

株式会社エンビテック

TEL: 03-6240-2655
FAX　: 03-6240-2656
E-mail：sales@embitek.co.jp
https://www.embitek.co.jp
https://www.embitek.co.jp/catalog.pdf