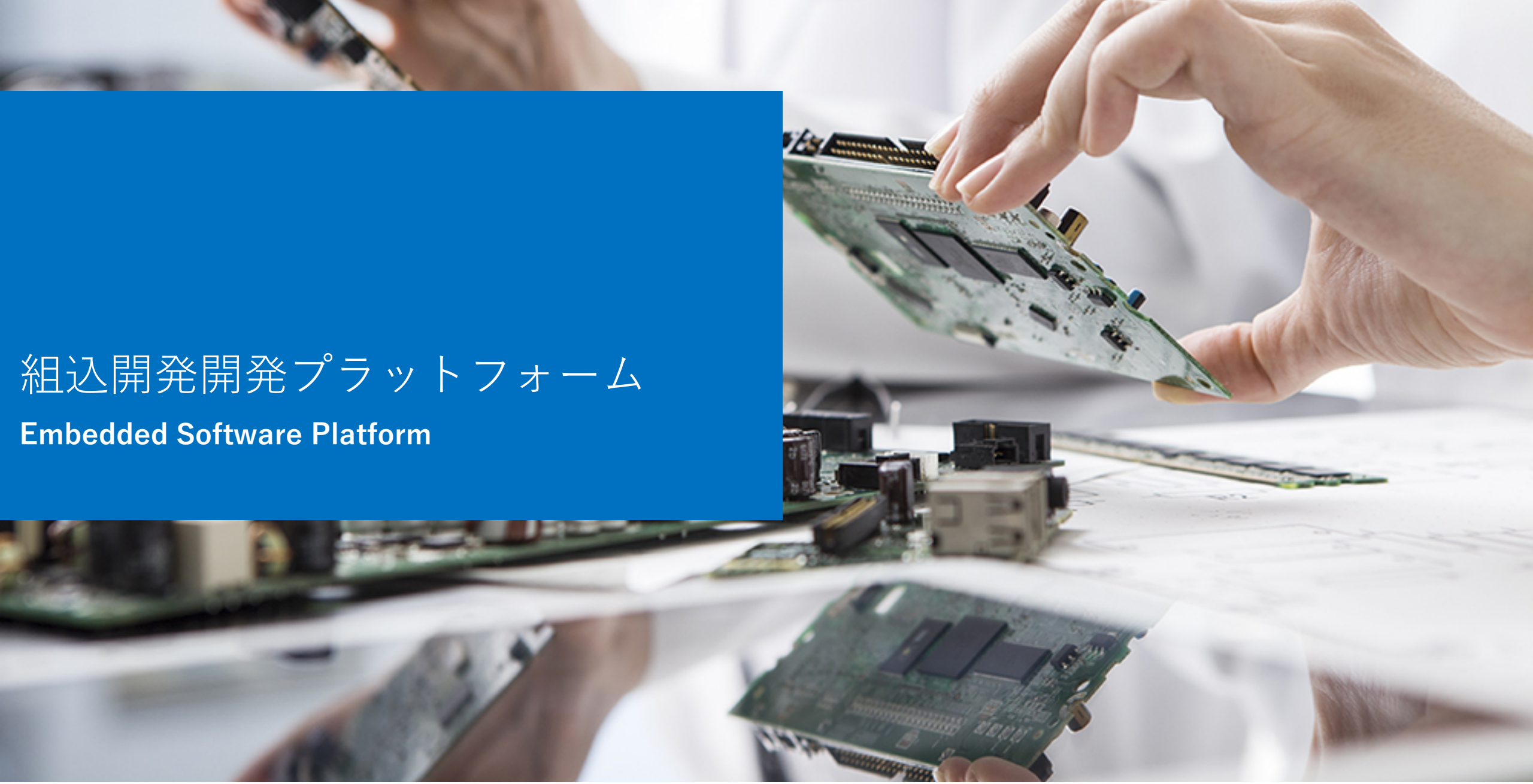


組込開発開発プラットフォーム

Embedded Software Platform

2021/10/20

Copyright © 2021 EmbITeK Co., Ltd. All Rights Reserved.

EmbiTeK

新製品開発にあたってのソフトウェア開発方針

新製品開発にあたって、お客様仕様を満たす開発方針、アプローチは様々かと存じます。

これまでの資産をベースに、新機能を追加する。

インターネット通信が機能追加されるので、OSを導入する必要性がある？

RTOSとLinux、Androidなど、製品に適切なOSは何がよい？

RTOSを使う場合、FreeRTOSなど無償製品を使う？

必要なミドルウェア・ソフトウェアモジュールはどういう指標で選定すべき？

開発プロジェクトにおけるお客様のご要望

新規開発プロジェクト

既存製品の延長上の製品・新コンセプトの製品開発

Add-on

これまでのソフトウェア資産をベースに機能追加

【事例】
これまでベアメタル（non-RTOS）でのアプリケーション開発に通信プロトコル実装が必要

iTRONアプリケーションで新機能を実装

ソフトウェア部品の要求

Restructure

ベースとなるソフトウェアを新規構築

【事例】
これまでの製品群とは全く異なるコンセプトの製品開発でソフトウェアを全面見直し

次世代製品開発にあたり、長期的な運用を視野にCPUから見直し

ソフトウェアプラットフォームの要求

Add-on

これまでのソフトウェア資産を
ベースに機能追加

ソフトウェア部品の製品群

iTRON
FreeRTOS
など
既存システム

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
JSON parser	MQTT		
HTTP Client	Dropbox Client		

TCP/IP

IPv4/IPv6	DHCP	ICMP
Wifi support	mDNS	Loopback
ACD/AutoIP	LLMNR	Multicast
ARP	DNS-SD	NetBIOS NS
CoAP	FTP sever	FTP client
RAW sockets	SMTP client	SNMP Agent
SNTP client	TCP/UDP	UPnP
PTP OC client	Web socket	Web server
PPP/PPPoE		

USB Host

MTP	MSD
HID	CDC
Printer	FTDI UART
LAN	MIDI
Vendor	HUB
Audio	CCID

USB Device

MTP	MSD (virtualMSD)
HID	CDC-ACM
Printer	CDC-ECM
Bulk	RNDIS
Audio	IP-over-USB
Video	DFU

ファイルシステム

- NAND
- NOR
- SD
- SDHC
- SDXC
- MMC
- eMMC
- CF
- USB メモリ

既存のシステムに必要な機能をアドオン

【お客様要求】
ソフトウェアの基本構造は変えず、できるだけ少ない工数で新機能実装

■メリット
新製品の追加要求に対応
ソフトウェア工数は最小

必要なコストも最低限

ソフトウェア部品の製品群

non-RTOS
既存システム

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
JSON parser	MQTT		
HTTP Client	Dropbox Client		

TCP/IP

IPv4/IPv6	DHCP	ICMP
Wifi support	mDNS	Loopback
ACD/AutoIP	LLMNR	Multicast
ARP	DNS-SD	NetBIOS NS
CoAP	FTP sever	FTP client
RAW sockets	SMTP client	SNMP Agent
SNTP client	TCP/UDP	UPnP
PTP OC client	Web socket	Web server
PPP/PPPoE		

USB Host

MTP	MSD
HID	CDC
Printer	FTDI UART
LAN	MIDI
Vendor	HUB
Audio	CCID

USB Device

MTP	MSD (virtualMSD)
HID	CDC-ACM
Printer	CDC-ECM
Bulk	RNDIS
Audio	IP-over-USB
Video	DFU

ファイルシステム

- NAND
- NOR
- SD
- SDHC
- SDXC
- MMC
- eMMC
- CF
- USB メモリ

ベアメタルシステム（non-RTOS環境）に必要な機能をアドオン

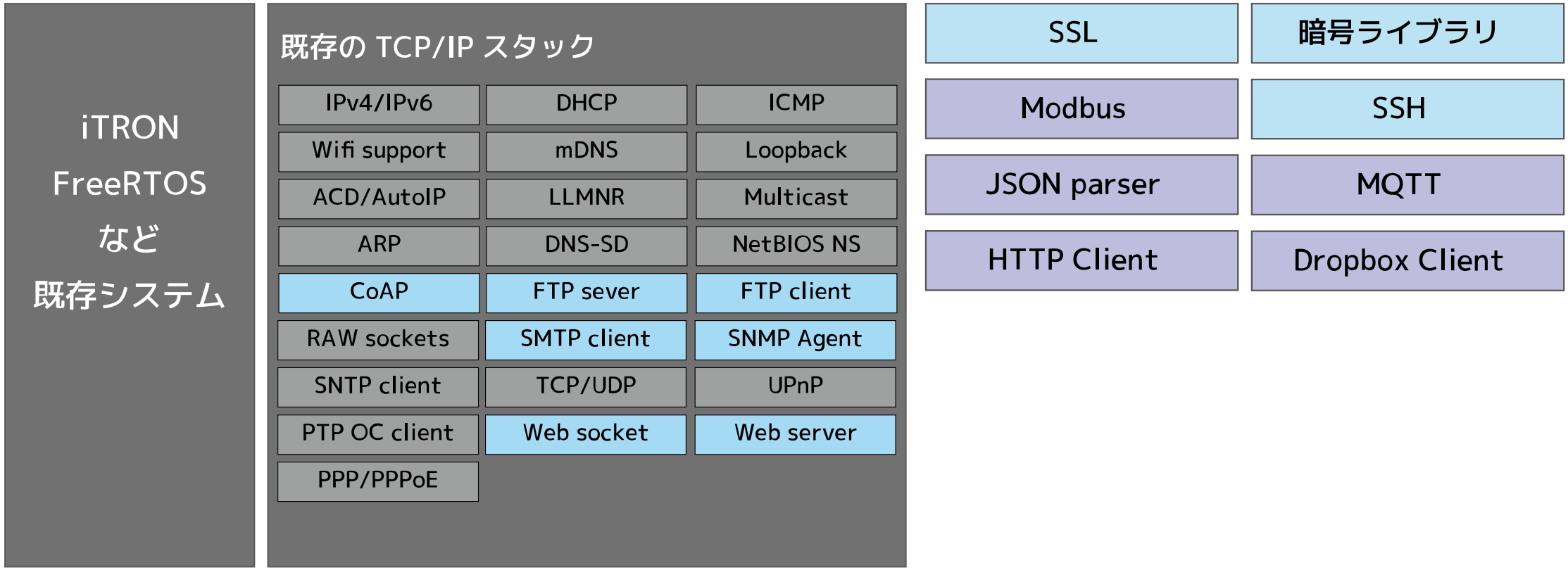
【お客様要求】
これまでRTOS搭載していない製品へ機能要件だけ追加したい。

※
Non-RTOSの場合
一部マルチタスク機能が必須となるソフトウェアは動作しません。

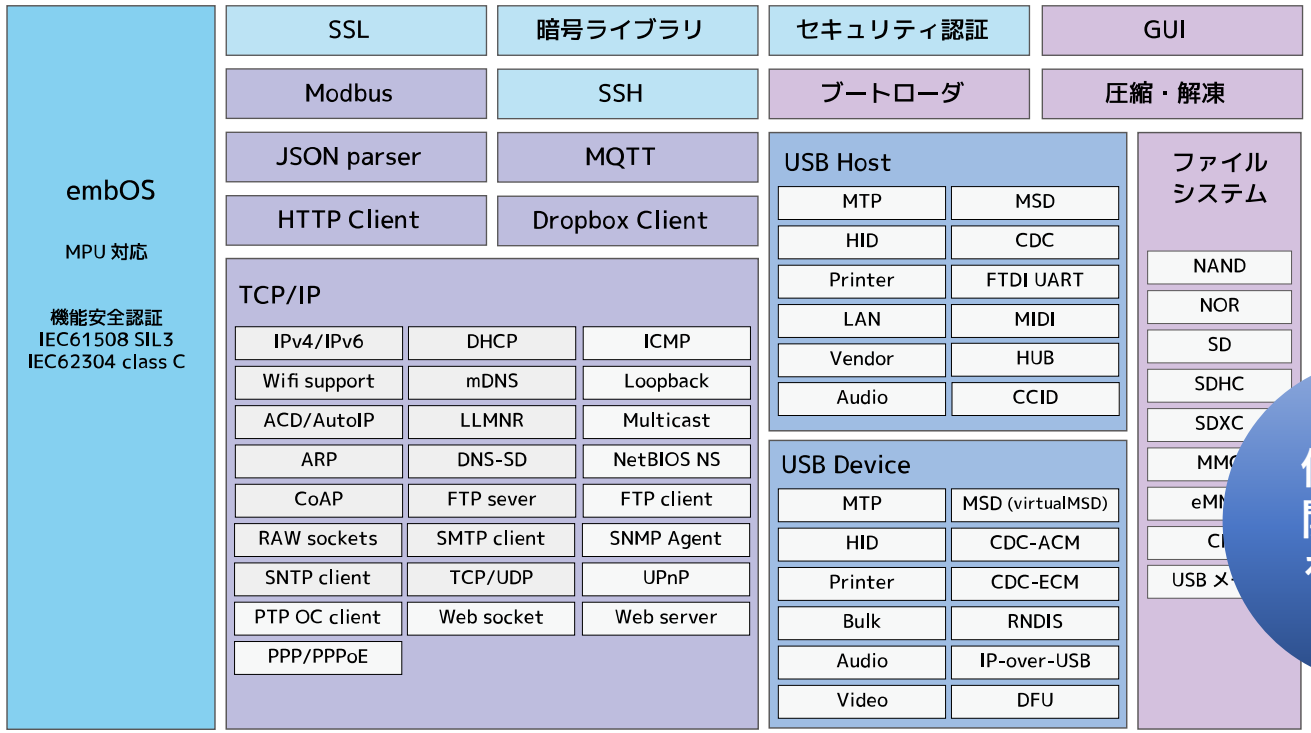
ソフトウェア部品の製品群

既存システム（TCP/IPネットワーク通信実装済み）に必要な機能をアドオン

【お客様要求】
既にTCP/IPネットワーク通信機能まで実装できているシステムに、暗号化やIoTクラウド連携のためのJSON parserやHTTPクライアントを追加したい。



コンパイラやCPUの依存性を極力排除



8/16/32bit
CPUコア

など

使いたい
開発環境
を選べる

SEGGER Embedded Studio

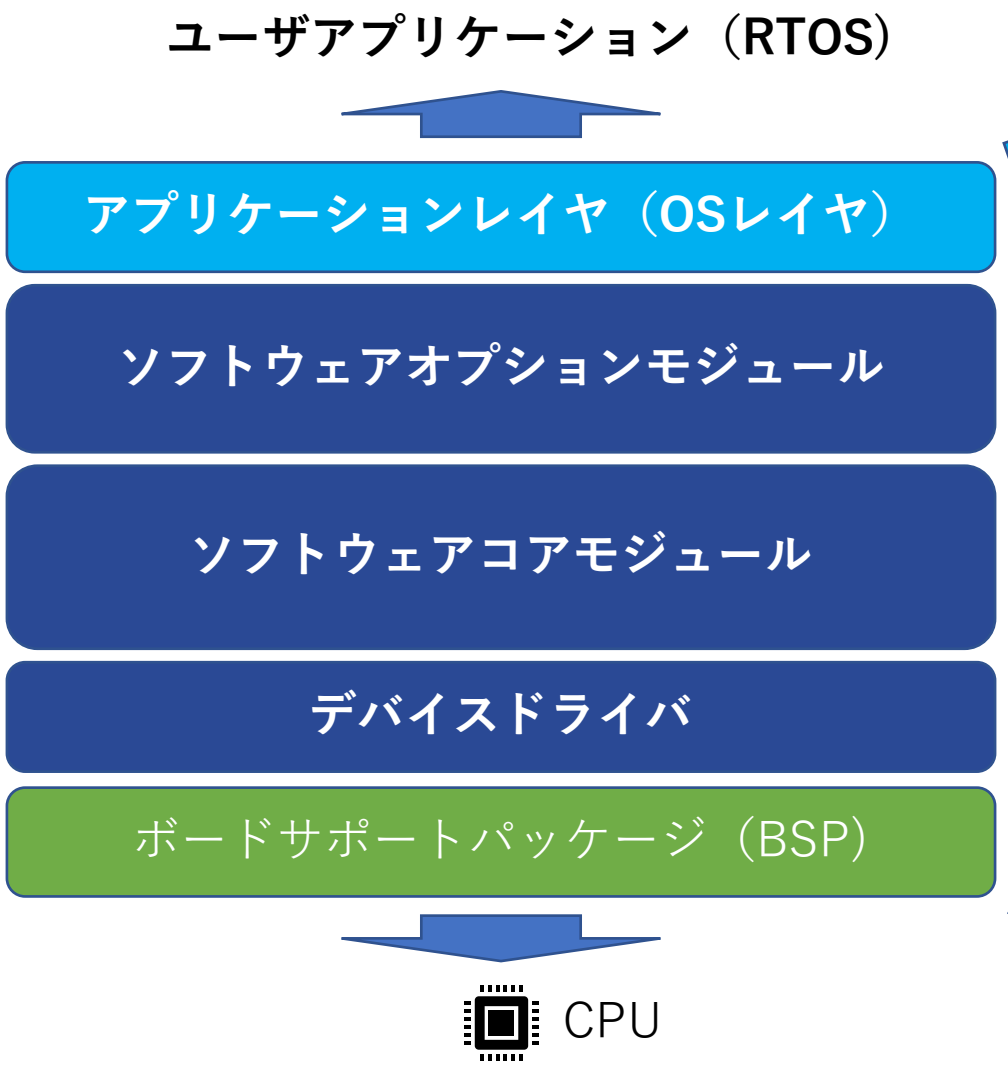
CPUベンダーツール

他社統合開発環境

今ご利用中の開発ツール・利用予定のツールで使えます。

使いたい
CPUを選べる

SEGGER社ソフトウェア製品基本コンセプト

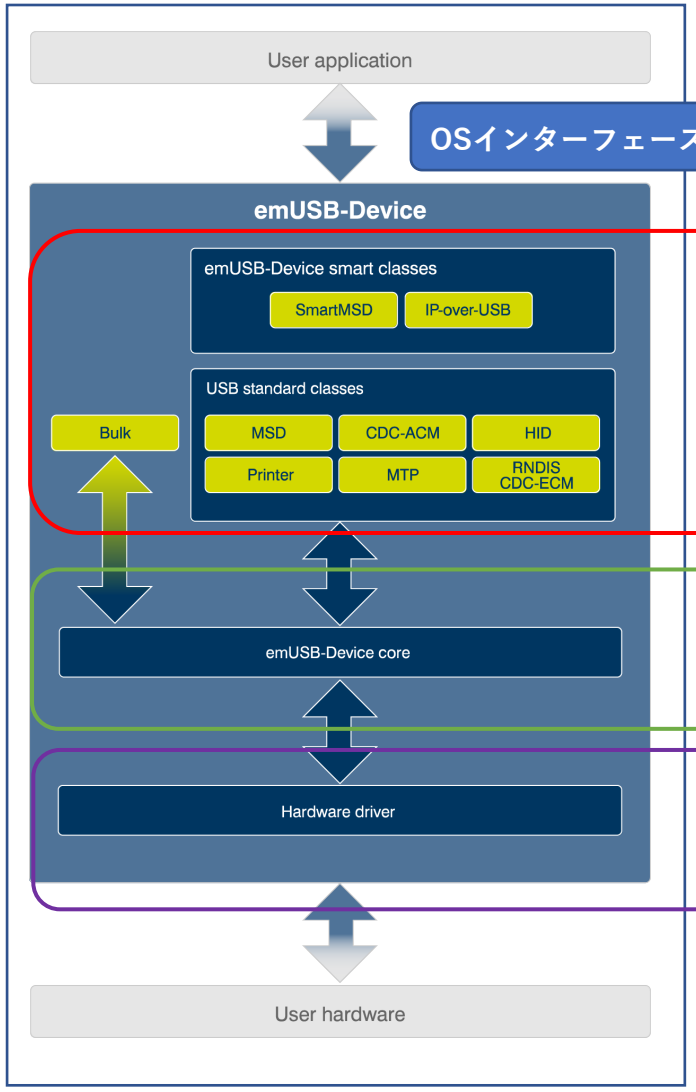


SEGGER社製embOSであれば、編集の必要はありません。
他社RTOS、non-RTOSの場合は、ユーザ編集が必要

ソースコードは提供しますが、
ユーザ様でコード編集する必要はありません。
（SEGGER社）

評価ボード用サンプル提供されているもので、そのまま動作する場合もあります。ユーザハードウェアに合わせて編集

SEGGER社ソフトウェアプロダクトの特徴



例: USB-Device構成イメージ

OSインターフェースレイヤ

利用するシステムに合わせて、OSレイヤを
ユーザで設定利用・non-RTOSベースも対応

USBクラスドライバ

USB コアモジュールから分離構成されており、
クラスの追加・削除も容易

USB Deviceコアモジュール

コアモジュールはユーザアプリケーションや
ハードウェアから独立、ソースコードを触る必要なし

ハードウェア依存部

CPU変更もデバイスドライバの変更で対応

ANSI-Cコードで開発

**コンパイラ依存の
拡張命令やプラグマ
不使用**

**ソフトウェアとしての基本設計がしっかりしているので、
CPU、コンパイラ、OSの制約なく利用できる。
既存のユーザシステムに「アドオン」できる。**

Restructure

ベースとなるソフトウェアを
新規構築

ソフトウェアとしての基本方針

新規でソフトウェアを整備するにあたり、大きな方針として、OSをどうするか？

ベアメタル
OSは使わない。

RTOSを利用しない通信実装なども
提案は可能ですので、ご相談ください。

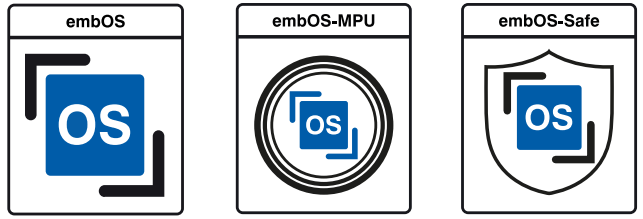
RTOSベース
各種RTOSで構築

汎用OSベース
LinuxやAndroidなどで構築

マルチOS
リアルタイム要求：RTOS
アプリケーション：Linux　などで構築

embOS概要

RTOSベース
各種RTOSで構築



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

Copyright © 2019 EmbITeK Co., Ltd. All Rights Reserved.

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

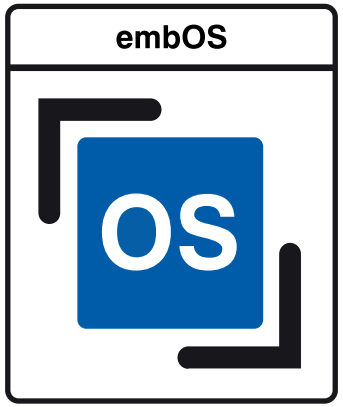
10億台以上の量産品出荷（SEGGER社調べ）

「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



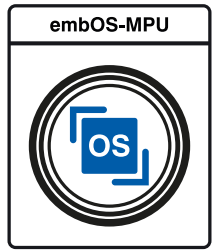
embOS基本パッケージ

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



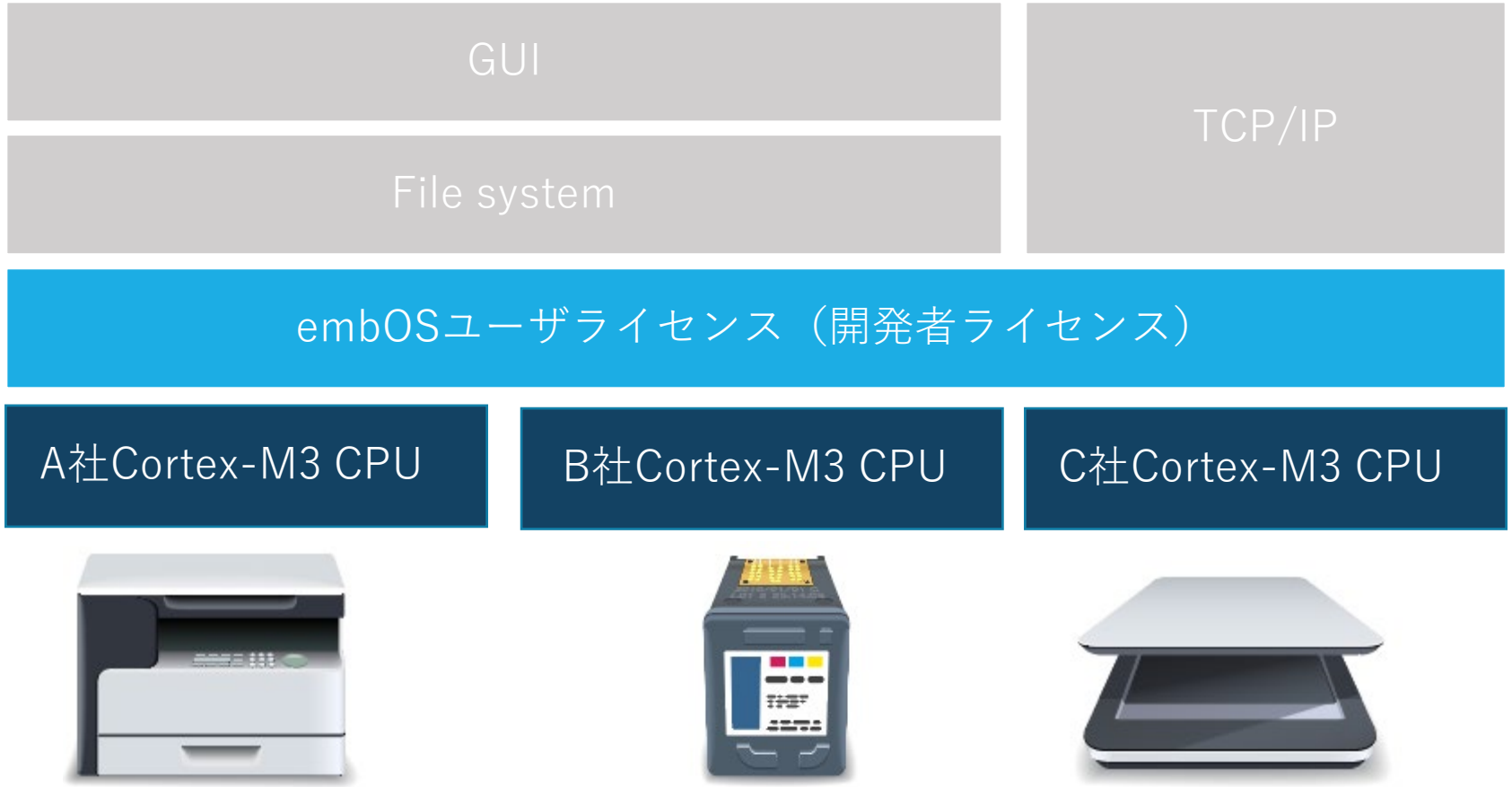
オブジェクト版

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

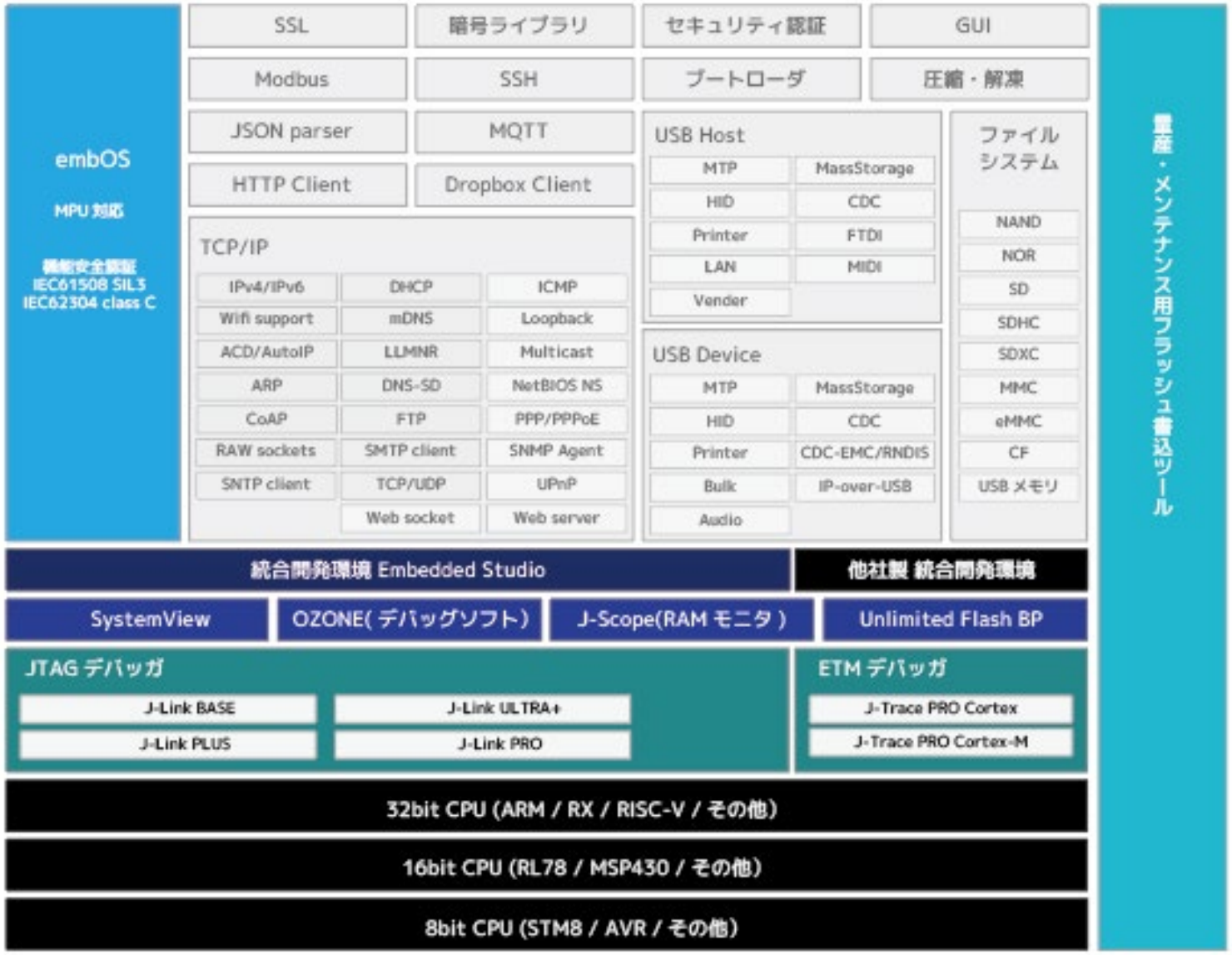
ロイヤリティフリー・開発製品無制限のユーザライセンスで提供可能

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）



１つのデバイスメーカにとらわれず、製品に合わせたCPU選定を容易に行うことができます。

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
ティックレスサポート

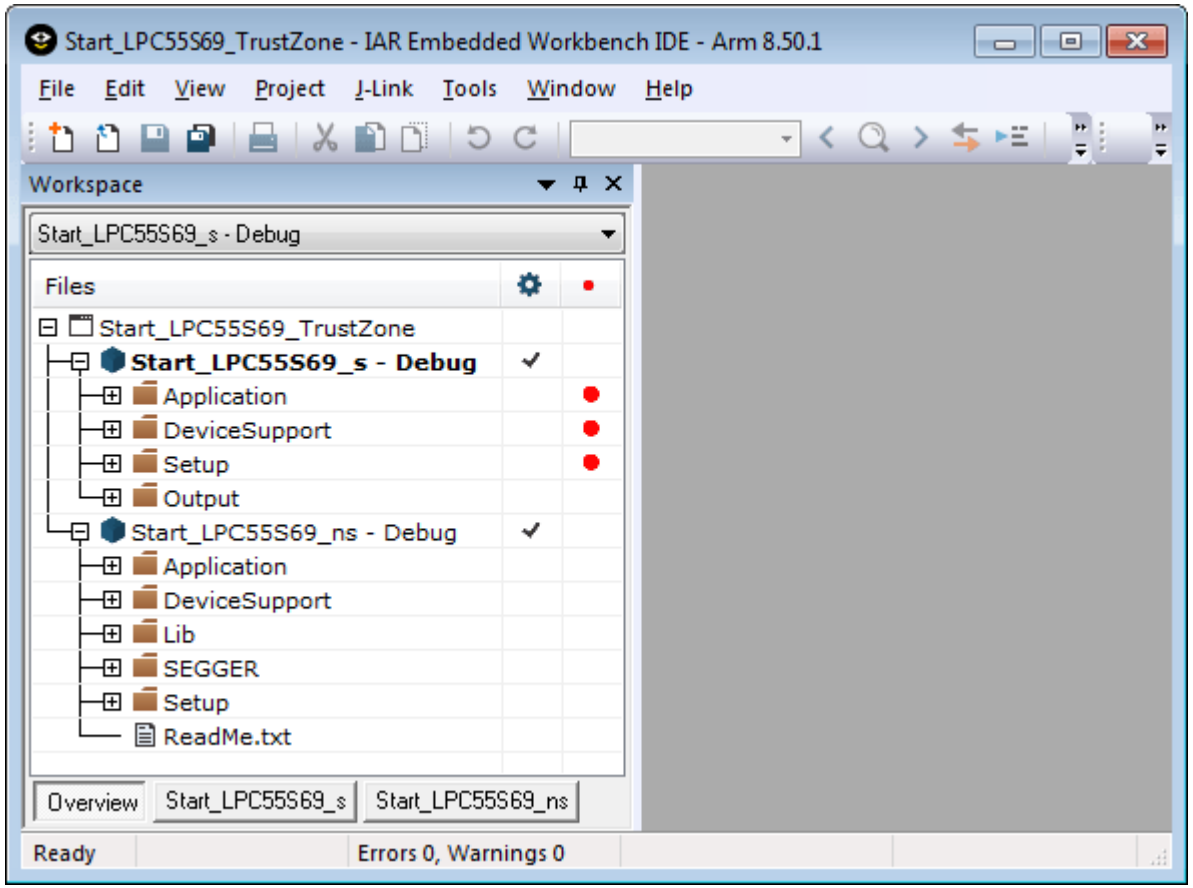
小さなフットプリントで、汎用性の高いRTOSソリューション
コンパクトながら高性能な機能を実装しています。

Copyright © 2021 EmbITeK Co., Ltd. All Rights Reserved.

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。

正確なサイクル実行時測定をサポート

製品特長

アプリケーションの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します。

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);
}
```

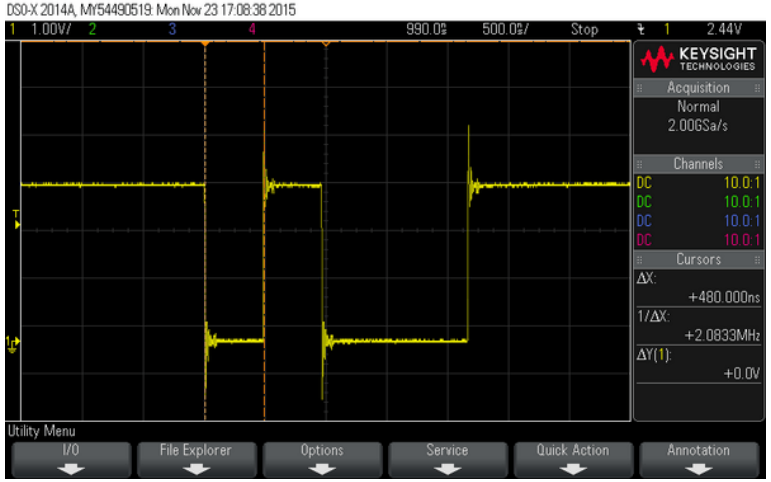
embOSの効率性：割込レイテンシー（遅延）なし

製品特長

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/

embOSの効率性：Ticklessサポート

製品特長

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の効率性：ペリフェラルの電力制御

製品特長

embOSによりペリフェラルの電力をコントロール

embOSのAPIを活用して、ペリフェラルの電力をコントロール

ペリフェラルの電力制御例（例）Peripheral A, B, Cを定義

```
#define OS_POWER_USE_A   (1 << 0)  // peripheral "A"
#define OS_POWER_USE_B   (1 << 1)  // peripheral "B"
#define OS_POWER_USE_C   (1 << 2)  // peripheral "C"
#define OS_POWER_USE_ALL (OS_POWER_USE_A | OS_POWER_USE_B | OS_POWER_USE_C)
```

次のコードで、Peripheral A, Cが初期化され、OS_POWER_UsageInc（）コールによって、使用中とマーク

```
void _InitAC(void) {
  ...
  OS_POWER_UsageInc(OS_POWER_USE_A); // Mark "A" as used
  OS_POWER_UsageInc(OS_POWER_USE_C); // Mark "C" as used
  ...
}
```

Peripheral Cの利用を止める場合、OS_POWER_UsageDec():をコールし、未使用とマーク

```
void _WorkDone(void) {
  ...
  OS_POWER_UsageDec(OS_POWER_USE_C); // Mark "C" as unused
  ...
}
```

embOSの効率性：電力制御2

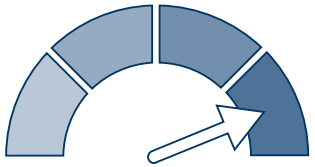
製品特長

OS_Idle（）中に、OS_POWER_GetMask（）を呼び出し、電源管理カウンタからビットマスクを取得します。
ビットマスクを使用して、制御レジスタのビット修正、使用中のペリフェラルのみ残します。

```
void OS_Idle(void) {  // Idle loop: No task is ready to execute
  OS_UINT PowerMask;
  OS_U16  ClkControl;
  //
  // Initially disable interrupts
  //
  OS_INT_IncDI();
  //
  // Examine which peripherals may be switched off
  //
  PowerMask  = OS_POWER_GetMask();
  //
  //  Store the content of CTRLREG and clear all OS_POWER_USE related bits
  //
  ClkControl = CTRLREG & ~OS_POWER_USE_ALL;
  //
  //  Set only bits for used peripherals and write them to the specific register
  //  In this case only "A" is marked as used, so "C" gets switched off
  //
  CTRLREG    = ClkControl | PowerMask;
  //
  // Re-enable interrupts
  //
  OS_INT_DecRI();
  for (;;) {
    _do_nothing();
  };
}
```

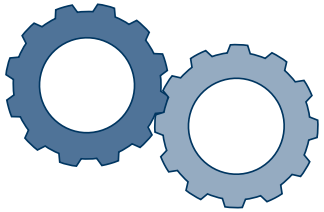
embOSの利点

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です。
組込ソフトウェア開発に求められる機能・性能・パッケージを提供します。



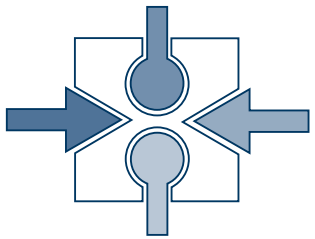
効率性

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

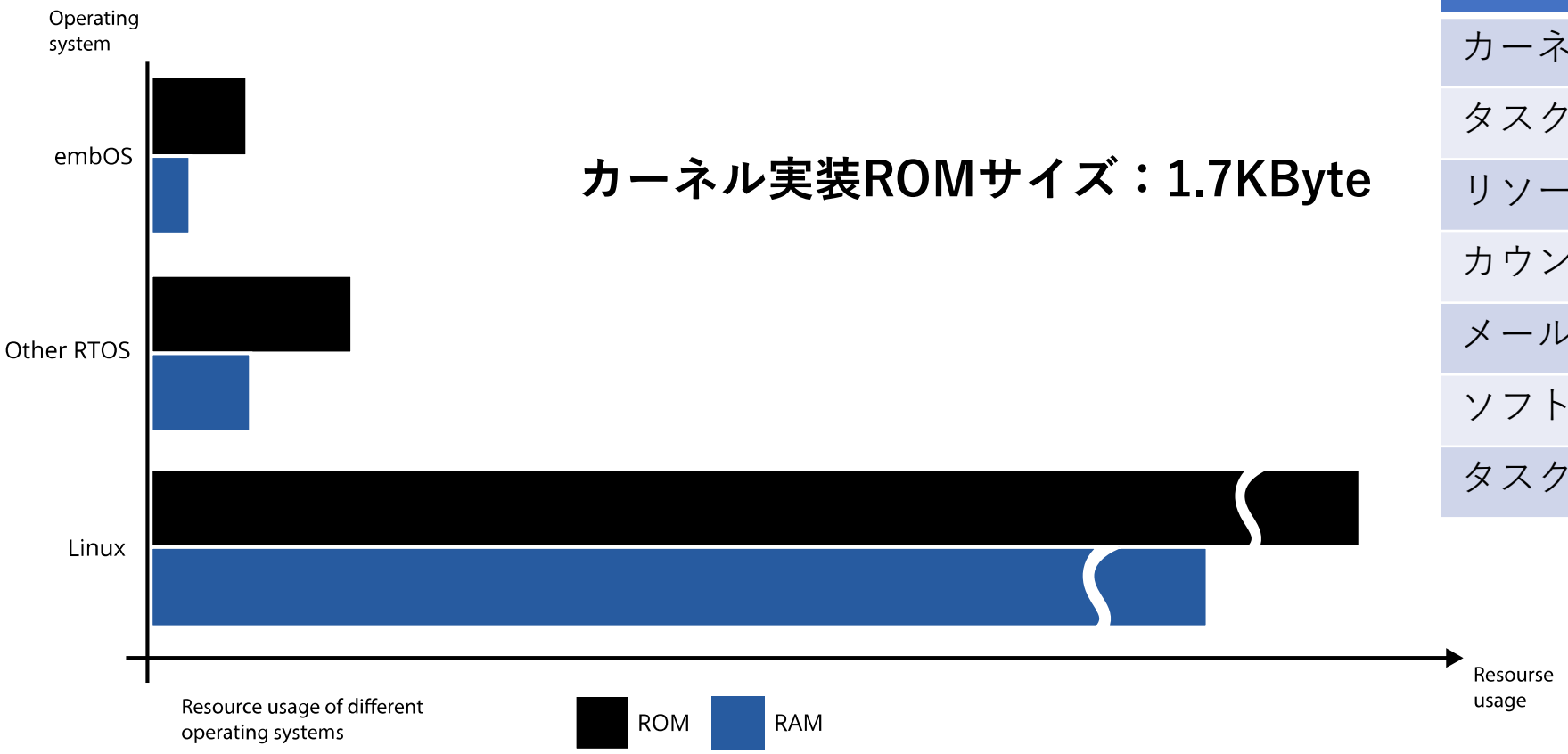
幅広いコア・コンパイラ対応で、
マイコンやツールに依存しない資産を構築

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：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。



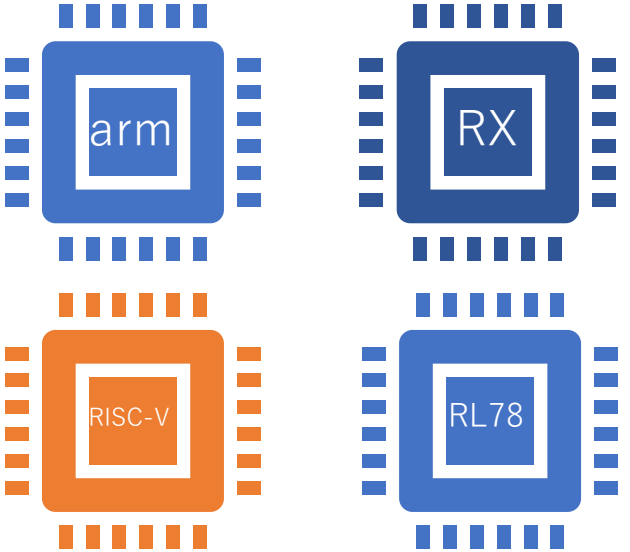
カーネル実装ROMサイズ：1.7KByte

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

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

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をサポートします。

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。

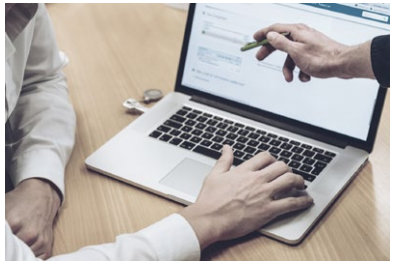
汎用的なロックメカニズムにより、プロセスの同期を行う事ができます。

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

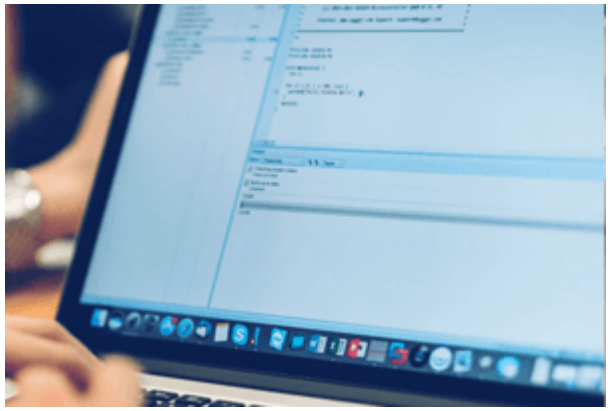
数十億台の量産製品出荷実績
リリース以来、25年にわたりお客様製品に搭載され量産出荷されています。



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。

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

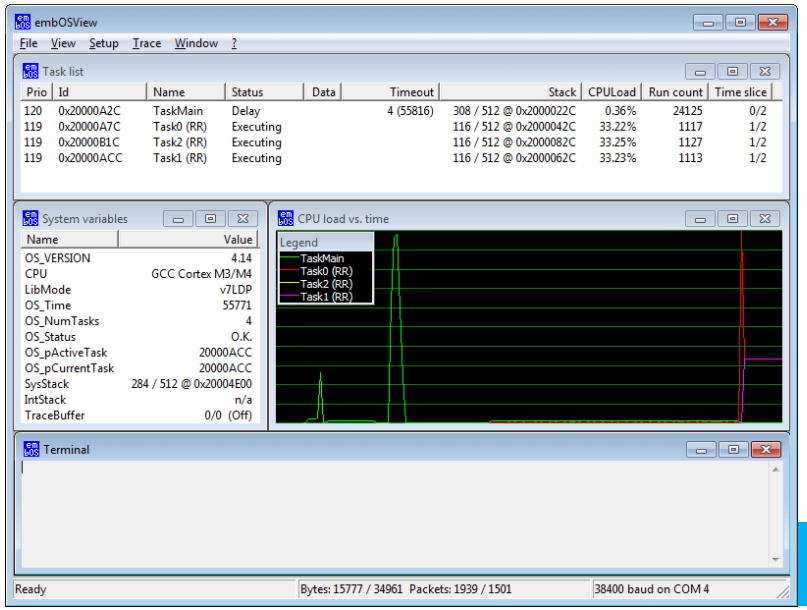
分かりやすいAPI
扱いやすいAPIを提供
初めてでも使いやすく

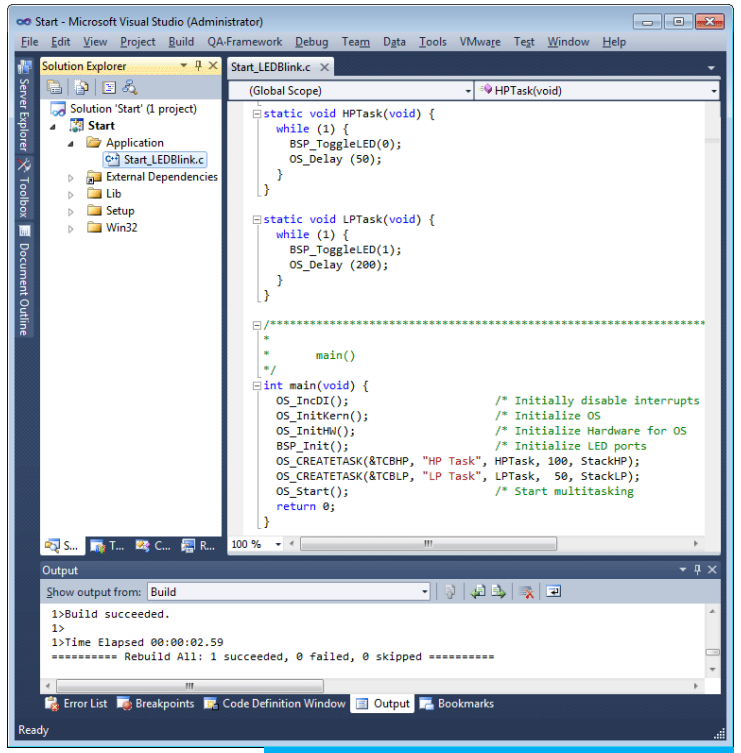
BSPサポート
500以上の評価ボードへ
BSPを提供しています。

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

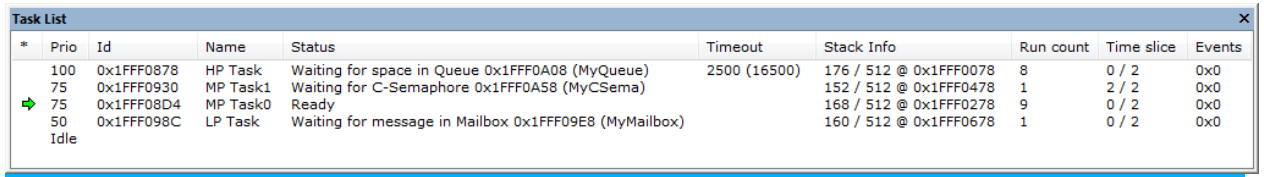
embOS開発支援ツール

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

ベースソフトウェアとしての実力は十分

実績

20年以上の販売実績と
グローバルでの製品搭載実績
日本国内でも医療機器メーカや
産業機器メーカ様中心に採用

評価

無償評価版（非商用利用）

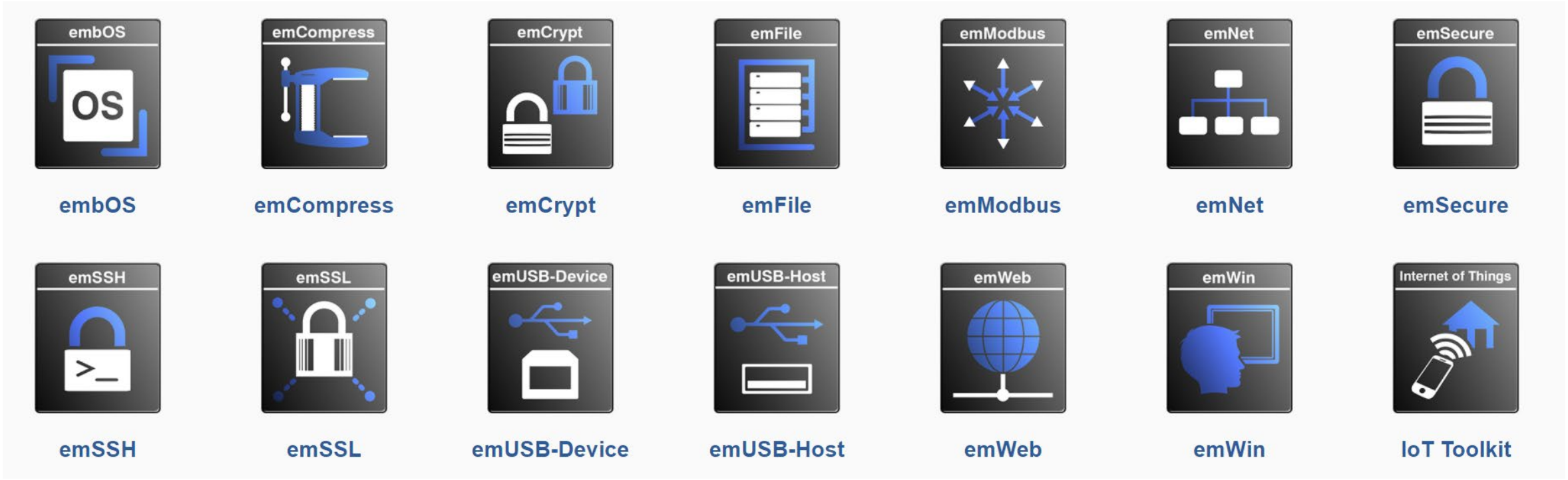
テクニカルサポート

日本語窓口対応
基本マニュアルは当社翻訳日本語版提供

SEGGER社RTOS開発チームと密に連携し、
お客様の問題解決

emPower OS

emPower OSは、RTOS/ミドルウェア製品各種をソースコード提供するパッケージになります。



IoT製品開発に要求されるソフトウェアモジュール製品を１つにパッケージングしたソリューション

ソフトウェアとしての基本方針

新規でソフトウェアを整備するにあたり、大きな方針として、OSをどうするか？

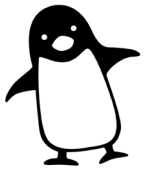
ベアメタル
OSは使わない。

RTOSベース
各種RTOSで構築

汎用OSベース
LinuxやAndroidなどで構築

マルチOS
リアルタイム要求：RTOS
アプリケーション：Linux　などで構築

組込システムに
Linuxを検討する理由



組込開発でLinuxを利用するメリット

組込み開発におけるLinux採用のメリット

オープンソースで無償

ミドルウェアが豊富

デバイスドライバが豊富

組み込みシステム開発に必要な環境がそろっている

デバッグツールが用意されている

WEBや書籍など豊富な情報が公開されている

引用元：EIPC「**組込み開発におけるLinuxシステムについて**」より
https://www.eipc.jp/embedded/linux/

コストの課題解決としてのLinux？

Linuxを利用しないと実現出来ないこと（開発手法やLinuxのみ用意されているアプリケーション利用など）もあり、必ずしもコスト解決のみがLinux選定理由になっていないと考えます。

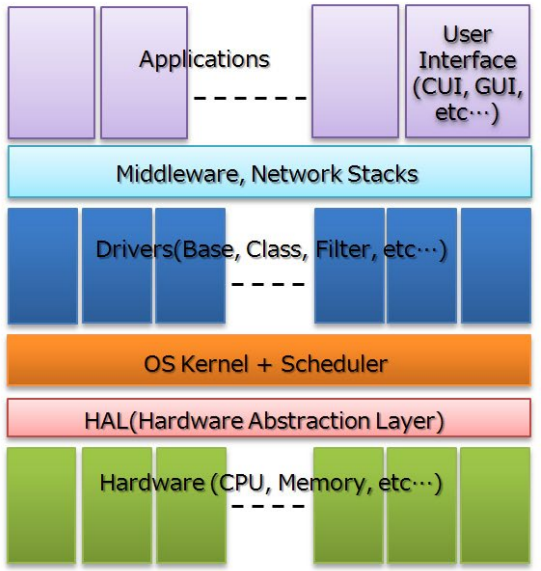
embOS vs GPOS(General Purpose OS)

GPOSがすべてOS上にアプリケーション開発するのに対して、
RTOSは、アプリケーションによりミドルウェア機能を使う、RTOSカーネル機能を使う、
ハードウェアをダイレクトにコントロールするような形で、最適な製品開発を行うことができます。

モータ制御やGPIOコントロールなど
GPOSを利用すると数百MHz以上の動作周波数が
必要な処理も最悪応答時間を決められるRTOS前提の
アプリケーションであれば、数十MHzの動作周波数で
動作可能

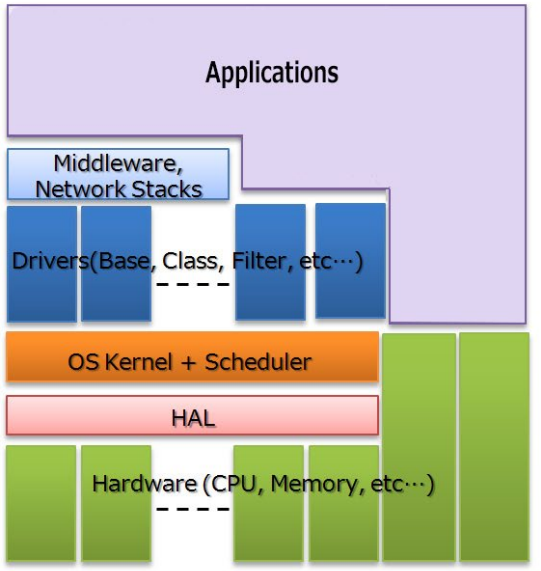
同じCPUでもリアルタイム性能を前提とした
アプリケーションであれば、RTOS制御を行うこと
で高パフォーマンスを得ることが可能

図1



■ GPOS利用アプリケーション

図2



■ RTOS利用アプリケーション

図表引用：ITmedia

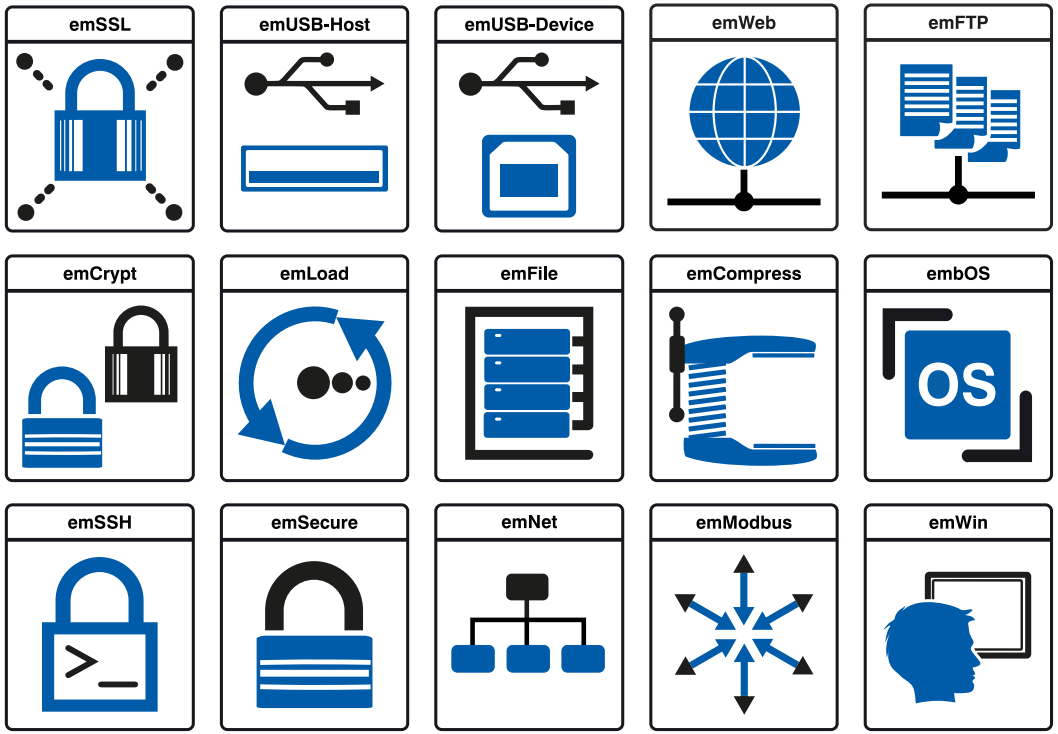
embOS vs GPOS(General Purpose OS)

	embOS +M/W	GPOS(Windows / Linux / Android …)
ROM	10 - 200KB	1 – 100 MB
RAM	KB単位	1 – 100 MB
Startup time	ミリ秒単位（1秒以下）	数秒～数十秒
RealTime Control	○	X
タイムクリティカル処理保証	クリティカルタスク処理可能	X
GPLソースコード開示義務	不要	Linuxではソースコード開示義務あり Windows は開示不要
デバッグ環境	リアルタイムデバッグ・トレース 診断可能	GDBサーバによるデバッグ・リアルタイムデバッグ不可
バージョン管理	OS/ミドルウエア 個別にバージョン管理可能	OSによるが、個別のコンポーネントごとに管理不可
プラットフォーム依存性	ローエンドマイコンからハイエンドCPUまで対応可能	基本的にハイエンドのCPU性能が必須
技術サポート	保守継続により、継続品質保証	ディストリビュータ方針による パッケージとして終了する可能性あり

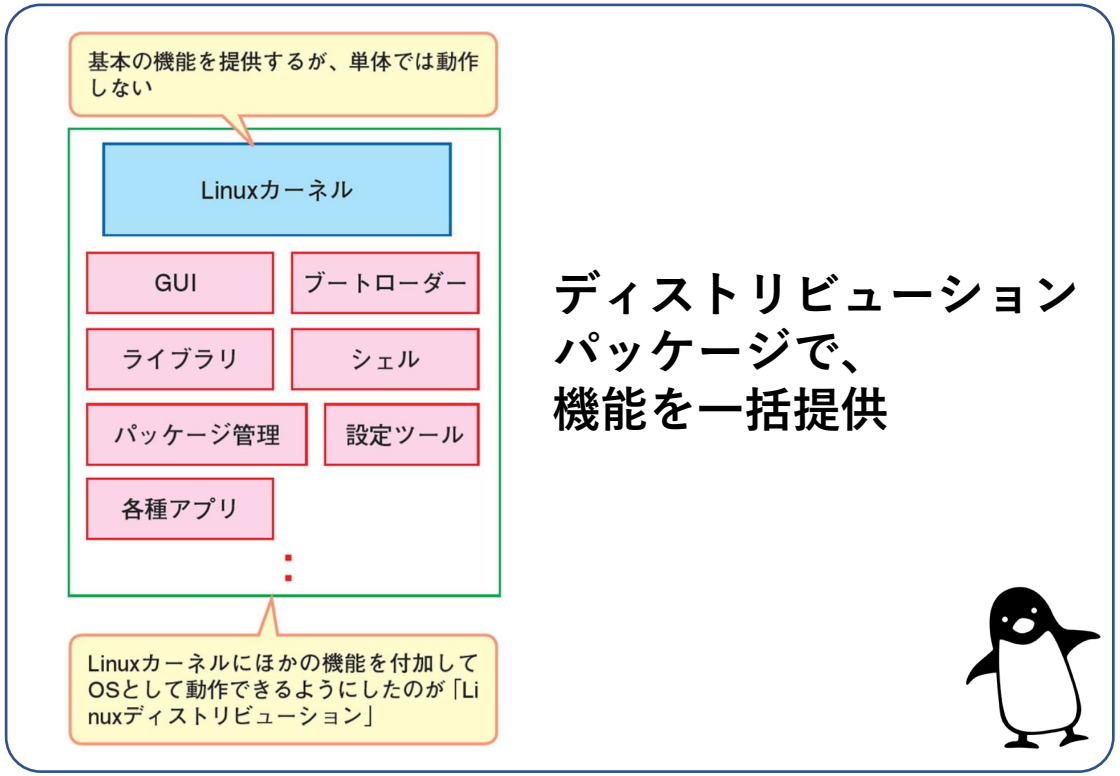
25年以上、継続提供

Windows Embedded Compact 7
2021年サポート終了など

RTOSシステムにおいて機能要件に応じてミドルウェア選定



機器の機能要件に合わせて、実装するミドルウェアを選択、取捨選択して導入。
最適なハードウェア構成を設定することができます。



ディストリビューション
パッケージで、
機能を一括提供

単体プロジェクトか全体プロジェクトか

ベースとなるソフトウェアを
新規構築

単体開発プロジェクトとしてみる

「個別製品開発プロジェクト」に最適なソリューション

最重視されるポイント
該当製品開発をいかに効率よく
ローコストに進められるか

ラピッドプロトタイピング
CPUメーカから提供のソリューションモデルを利用する。
コストダウンの手段として、マルチOS、Linuxを使う　などなど

≠

コスト効率

この後の製品開発を見越したプロジェクトとしてみる

「開発基盤の整備を前提とした開発プロジェクト」に最適なソリューション

最重視されるポイント
ソフトウェア資産価値を最大化しつつ
該当製品開発を効率良く進められるか

個別製品開発プロジェクト

■ 個別製品開発プロジェクト

目的：定義された予算（開発／量産）・スケジュール内で必要要件を満たし、信頼性の高い製品を開発する。

プロジェクトの全体計画の中で、外部商用ツール・ソフトウェアを利用を定義

目的	手法	ツールメーカが提供できるサービス・製品の要件
予算内に納める	ローコストな製品を選定	適正な価格を提示する
スケジュール内で完成	使いやすいツールを選定	ユーザ様の開発期間を短縮できる製品／サポートサービスを提供する
製品要求を満たす	商用製品を使う場合、機能要求を適切に満たせる製品を選別する	要件定義にあった製品・サービスの提供
信頼性の高い製品	評価使用を行い、実績を評価	ユーザニーズに合った評価環境を提供。実績を正確に情報提供

RTOS/ミドルウェアコンポーネント／開発ツール／量産書込ツールまで幅広く対応可能

SEGGER　EmbiTeK

ソフトウェア基盤としてのあるべき姿

「開発基盤の整備を見越した開発プロジェクト」と定義

一般的に求められる事

汎用性
TCO

- 大量生産にも使え、**テストモデル・試作開発にも使えるプラットフォーム**であること
- 開発製品に合わせて**CPU・ハードウェア選定を自由に行える**こと
- **長期的な供給・運用**ができること

初期コストと運用コストまで含めたコストメリット

利便性

- RTOS/ソフトウェア製品のソースコードが最適化され、**API定義が整然**としている。
- **ソフトウェアモジュールの設計が統一**されている。
- **開発支援ツールが充実**している。
- ソフトウェア開発の工数削減に貢献できる**開発ツールを柔軟に選べる。**

ソフトウェア基盤としてのあるべき姿

一般的に求められる事

汎用性
TCO

■ 大量生産にも使え、**テストモデル・試作開発にも使えるプラットフォーム**であること
■ 開発製品に合わせて**CPU・ハードウェア選定を自由に行える**こと
■ **長期的な供給・運用**ができること

1CPUアーキテクチャ：
「Cortex-M3ライセンス」であれば、様々なデバイスメーカのマイコンで利用可能

ユーザライセンス	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

CPUライセンス	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ライセンスについては、お客様要求に応じてライセンス内容をカスタマイズ提案可能です。
(例)開発製品数:10製品型番／利用可能開発者数：10名に限定　CPUとコンパイラを３種に変更など

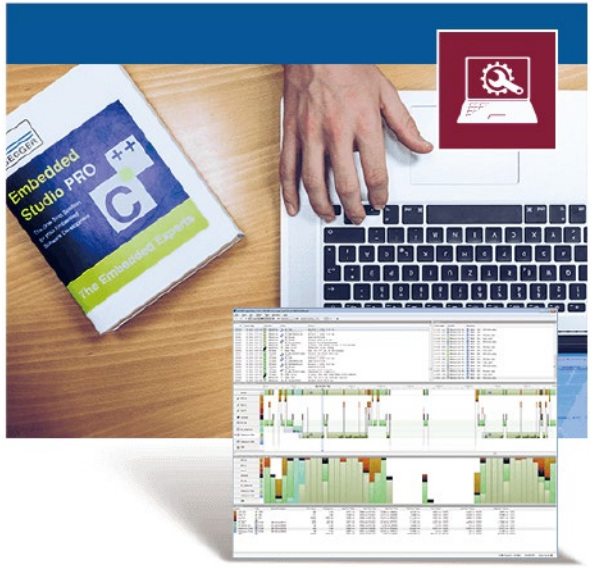
ソフトウェア基盤として

利便性

- RTOS/ソフトウェア製品のソースコードが最適化され、**API定義が整然**としている。
- **ソフトウェアモジュールの設計が統一**されている。
- **開発支援ツールが充実**している。
- ソフトウェア開発の工数削減に貢献できる**開発ツールを柔軟に選べる**。

基盤ソフトウェア開発に必要な全てを揃え、ユーザニーズに合わせて、外部ツール・ソフトウェアも利用可能

RTOS / GUI / ファイルシステム / 圧縮・解凍 / ブートローダ / IoT（JSON Parser / HTTPc）
MQTT / TCP/IP / Modbus Master / Slave
USB Host / USB Device / 暗号化ライブラリ / セキュア認証 / SSL / SSH



統合開発環境
システム可視化・記録ツール
デバッグソフトウェア

JTAG/SWDデバッガ
ETMトレースデバッガ

フラッシュ書込ツール
量産用大量書込ツール
スタンドアロンメンテナンス書込
セキュア書込ツール

とはいえ、使えるお金は無限ではない。

単体開発プロジェクトとしてみる

「個別製品開発プロジェクト」に最適なソリューション

≠

この後の製品開発を見越したプロジェクトとしてみる

「開発基盤を前提とした開発プロジェクト」に最適なソリューション

最重視されるポイント
該当製品開発をいかに効率よく
ローコストに進められるか

最重視されるポイント
ソフトウェア資産価値を最大化しつつ
該当製品開発を効率良く進められるか

SEGGER　EmbiTeK

投資コストを最低限に抑えたい。

サポート要求とコストによって選択いただけます。

ライセンス製品パッケージは同じでサポート窓口対応により、２種の価格提案をいたします。
ご購入後、ライセンスの運用状況に合わせて、更新時切り替えも可能。

ローコスト対応

SEGGERメーカサポート（英語）

ソフトウェアライセンス
（ソースコード・ツール）

日本語サポート窓口対応

EmbITeKサポート（日本語）

＋

SEGGERメーカサポート（英語）

ソフトウェアライセンス
（ソースコード・ツール）

まとめ

ユーザ様のソフトウェア開発ニーズに合わせて、提案いたします。

Add-on

これまでのソフトウェア資産を
ベースに機能追加

Restructure

ベースとなるソフトウェアを
新規構築

「単体製品開発プロジェクト」

*「開発基盤を前提とした
開発プロジェクト」*

お気軽にご相談ください。

キャンペーンサービス

新製品開発にあたって、お客様仕様を満たす開発方針、アプローチ

これまでの資産をベースに、新機能を追加する。

インターネット通信が機能追加されるので、OSを導入する必要性がある？

RTOSとLinux、Androidなど、製品に適切なOSは何がよい？

RTOSを使う場合、FreeRTOSなど無償製品を使う？

必要なミドルウェア・ソフトウェアモジュールはどういう指標で選定すべき？

ご要求に応じたソフトウェア提案をいたします。

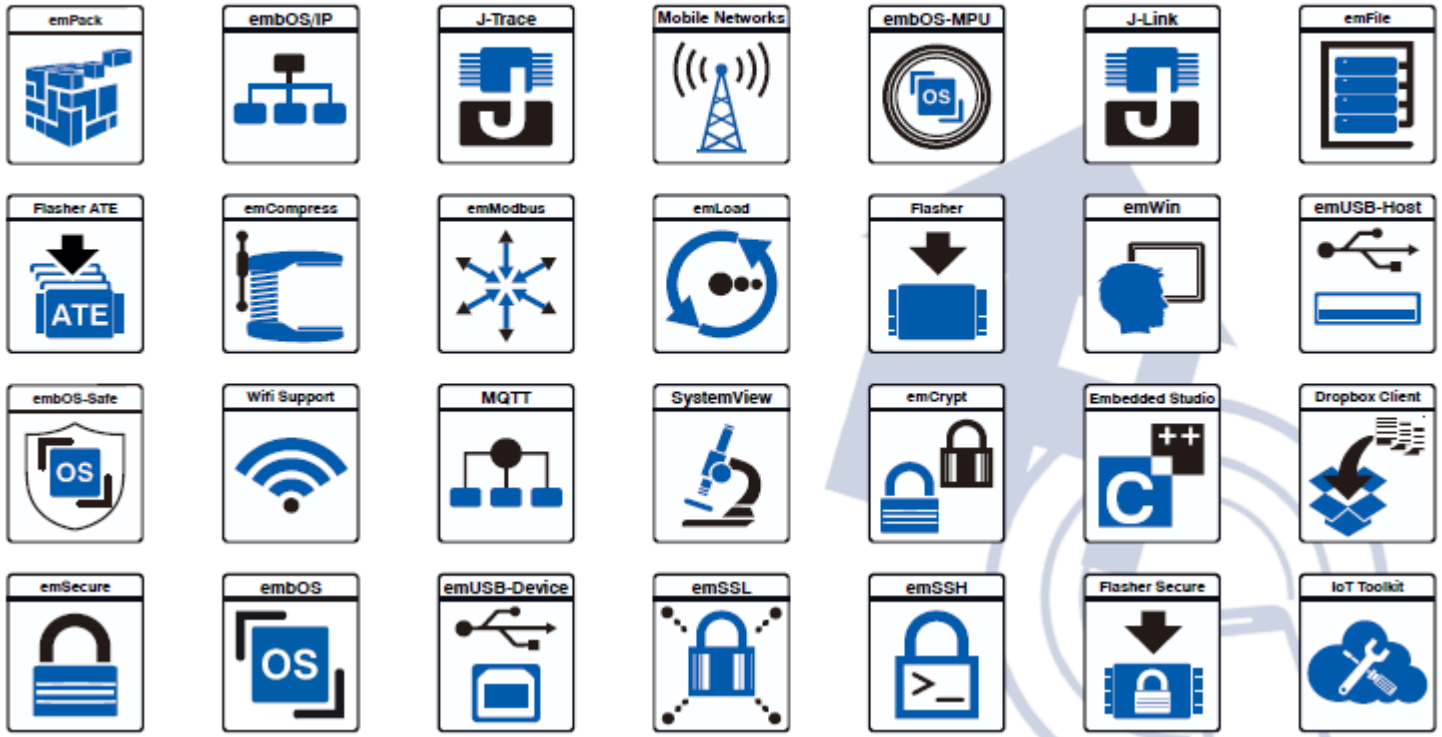
お問合せ窓口

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

LAN
USB
j-link
Pro
SEGGER
www.segger.com
Target
Flasher
The in-circuit programmer!
ARM
SEGGER
Ready
O.K.
Not ready
Fail
PROG


emPack
embOS/IP
J-Trace
Mobile Networks
embOS-MPU
J-Link
emFile
Flasher ATE
ATE
emCompress
emModbus
emLoad
Flasher
emWin
emUSB-Host
embOS-Safe
OS
Wifi Support
MQTT
SystemView
emCrypt
Embedded Studio
Dropbox Client
emSecure
embOS
OS
emUSB-Device
emSSL
emSSH
Flasher Secure
IoT Toolkit

Appendix

Catalog Stand

各種資料をダウンロード可能です。

https://www.embitek.co.jp/download/catalogs

総合カタログ

https://www.embitek.co.jp/download/MB_SolutionGuide.pdf

J-Linkカタログ

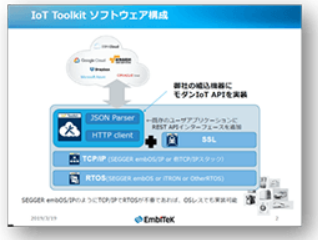
https://www.embitek.co.jp/download/MB-CTLG-JLink.pdf

Flasherカタログ

https://www.embitek.co.jp/download/MB-CTLG-Flasher.pdf

ソフトウェアライセンスについて

https://www.embitek.co.jp/download/MB_SWLicModels.pdf



IoT Toolkit製品資料

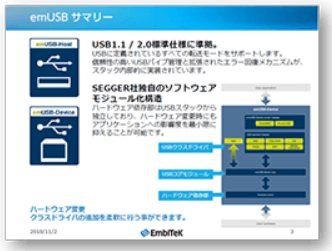
https://www.embitek.co.jp/download/ps/IoTtoolbox.pdf

圧縮解凍製品資料

https://www.embitek.co.jp/download/ps/emCompress.pdf

ファイルシステム製品資料

https://www.embitek.co.jp/download/ps/emfile.pdf



USB製品資料

https://www.embitek.co.jp/download/ps/SeggerUSBSolution.pdf