



arm Cortex / RISC-V 組み込みソフトウェア 統合開発環境



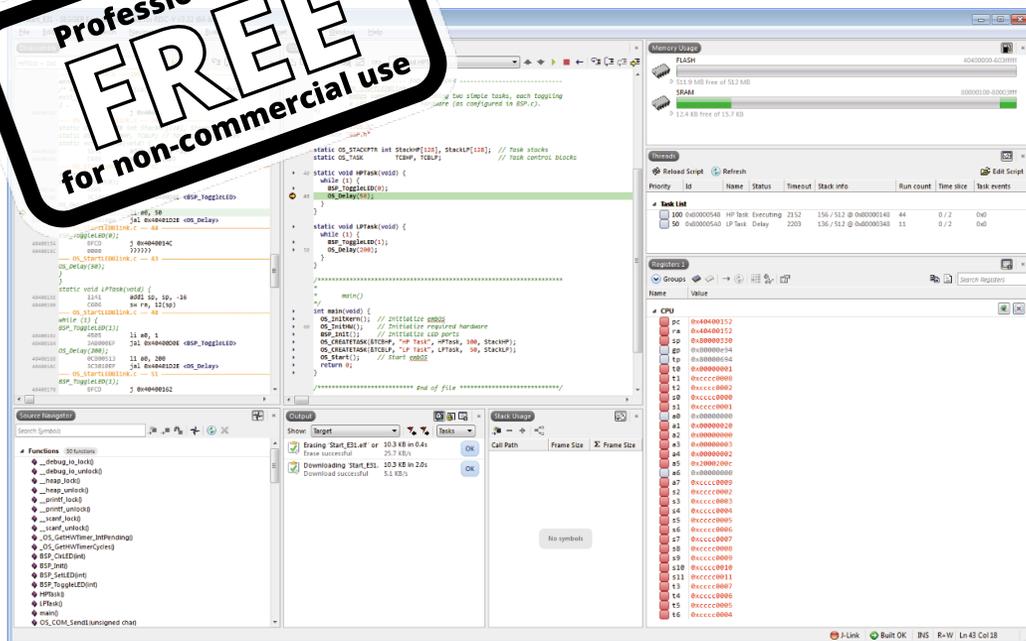
Embedded Studio

ローコストなプロフェッショナル統合開発環境
最新のデバイス対応サポートパッケージ無償配布

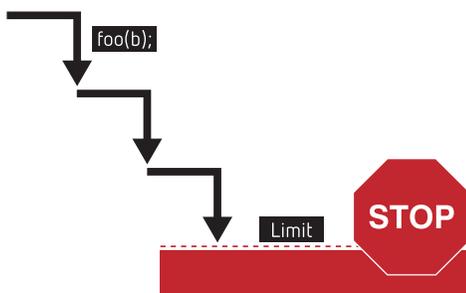
SEGGER Embedded Studio



非商用利用（評価用途）無償フル機能対応



- **クロスプラットフォーム対応**（Windows, MacOS, Linux）
- SEGGERコンパイラ、Clang/LLVM、GCC C/C++同梱
- マルチスレッドコンパイル・ビルド対応
- SEGGER社により**最適化されたCランタイムライブラリ**
- J-Linkと統合化されたデバッガ
- 高機能プロジェクトマネージャ
- CPUサポートパッケージ、簡単にプロジェクト立ち上げ



スタックオーバーフローを防ぐ独自技術

SEGGER STOP技術を利用することで、ユーザアプリケーションのスタックオーバーフローを未然に防ぐことができます。ユーザコードを改変することなく、オプション選択だけで利用頂くことができます。

Embedded Studioとは

組み込み開発用のプロフェッショナル環境をリーズナブルな提供価格で利用可能



Project Manager

プロジェクトマネージャ/IDE管理

高い生産性を実現するプロジェクトエクスプローラ / 外部ツールチェーンからのプロジェクトインポータ



Code Editor

コーディング

生産効率を上げるコードエディタ

```
void MainTask(void) {
    U32 v;
    P0_PIE = p0Pie;
    char ac[256];
    const char *svolname = "...";
    FS_XLog("Start\n");
    // Initialize file system
    FS_Init();
    // Check if low-level format is required
    // FS_FormatIfRequired(volname);
    // Check if volume needs to be high level formatted.
    if (FS_IsLowFormatted(volname) == 0) {
        FS_XLog("High-level format\n");
        FS_Format(volname, 0x1);
    }
    SEGGER_sprintf(ac, sizeof(ac), "Running sample on %s\n", volname);
    FS_XLog(ac);
    v = FS_GetVolumeFreeSpace@0(volname);
    if (v < 10000) {
        SEGGER_sprintf(ac, sizeof(ac), " Free space: %i0 bytes\n", v);
    } else {
        v >>= 10;
        SEGGER_sprintf(ac, sizeof(ac), " Free space: %i0 bytes\n", v);
    }
    FS_XLog(ac);
    SEGGER_sprintf(acFileLine, sizeof(acFileLine), "%i\\file.txt", volname);
    SEGGER_sprintf(ac, sizeof(ac), " Write test data to file %s\n", acFileLine);
    pFile = FS_Open(acFileLine, "w");
    if (pFile) {
        FS_Write(pFile, "text", 4);
        FS_Close(pFile);
    } else {
        SEGGER_sprintf(ac, sizeof(ac), "Could not open file: %s to write.\n", acFileLine);
        FS_XLog(ac);
    }
    v = FS_GetVolumeFreeSpace@0(volname);
}
```



Code Analysis

コード解析

コードの静的解析から
アプリケーション分析



静的コード解析
スタック使用量解析
外部ツール連携
メモリ使用量表示
コードアウトライン表示



Build Tools

ビルド

一般的なGCCとの比較で約20%の
生成効率の良い高性能コンパイラ
SEGGER Compiler / Linker



高性能Cランタイムライブラリ
emRun



高性能C++ランタイムライブラリ
emRun++



浮動小数点ライブラリ
emFloat



Debug

デバッグ

J-Linkと統合した高性能デバッガ
RTTインターフェース対応



J-Link PLUS以上の製品で
スタンドアロンデバッグ
ソフトウェアOZONEも利用可能



CPU support package

CPUサポートパッケージ

各種マイコンサポートファイル管理 / 各種評価ボード用BSPサンプル管理



Coding



プロジェクト管理



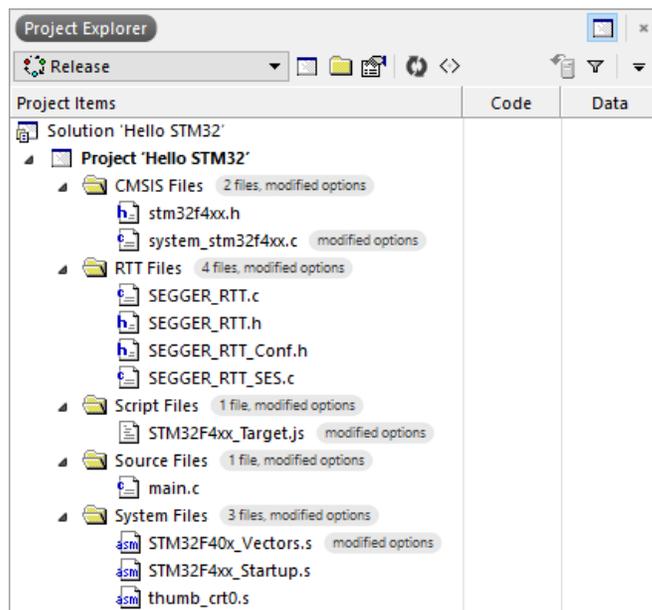
コードエディタ



静的コード解析

プロジェクト管理

生産性の高いアプリケーションのソースコード管理

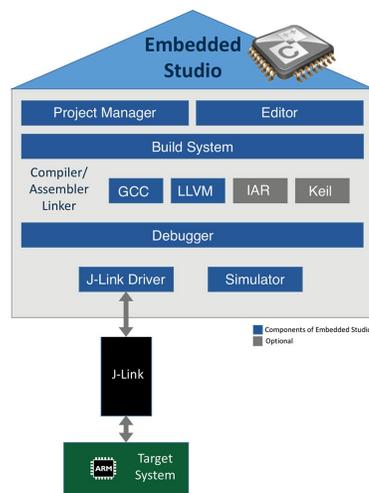


プロジェクトエクスプローラ

Embedded Studioでは、アプリケーション開発のための一元化されたソースコードファイル管理を提供します。

プロジェクトエクスプローラは、プロジェクト、ファイルを整理し、それらのプロパティを表示、必要なコマンドに素早くアクセスできます。

プロジェクトエクスプローラでは、各ファイル、フォルダの情報（フォルダ内のファイル数、コンパイルされたファイルのコードとデータサイズなど）を表示します。



プロジェクトインポータ

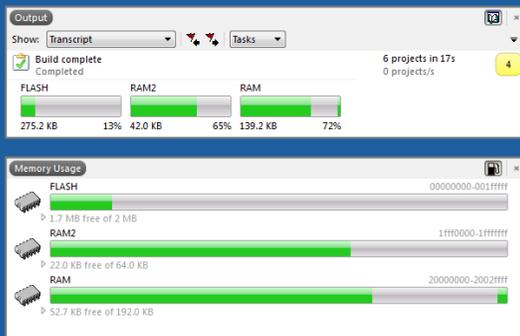
外部ツールチェーンサポートにより、Embedded Studioは、作成されたツールチェーンからの移行をスムーズに行うことができます。

※ 拡張命令やアセンブラなど、手動で変更する必要があるコード/ファイルもあります。

静的コード解析

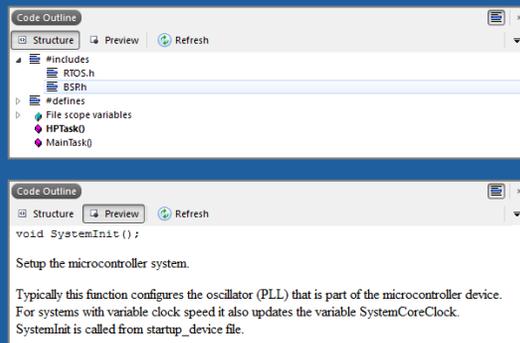
記述されたコード、コンパイル結果から静的解析を行い、リソースの使用状況、アプリケーションを解析

メモリ使用量表示



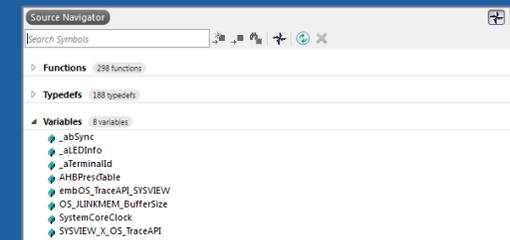
お客様のアプリケーションコードを解析することによりROM/RAM使用量を可視化して表示

コードアウトライン



大規模なアプリケーションで構造体をアウトライン表示することにより任意の要素に簡単にアクセス

ソースナビゲータ



シンボル、関数が定義されたソースコードを検索し、表示することができます。開発者は各定義ソースに簡単にアクセス可能

シンボルブラウザ

Name	Range	Size
.._data	20000000-2000008b	140
.._data_run	2000007c-2000008b	16
.._init	20000078-2000007b	4
.._stack	20000060-2000002f	48
.._heap	200010e4-2000104e3	1,024
.._text	080003c4-08004e47	19,076

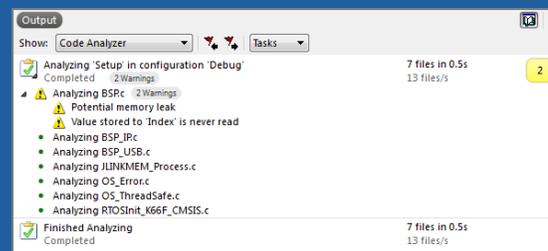
各シンボルのメモリ位置とサイズを表示します。シンボルはソースファイル、セクション、タイプ毎にグループ化して表示

スタック使用量解析

Call Path	Frame Size	Σ Frame Size
IP_UDP_OnRxIP4	48	232
IP4_OnRx	40	184
IP4_TCP_OnRx	80	328
main	24	68
BSP_Init	0	0
OS_CreateTask_R	32	32
OS_InitHW	32	44
OS_Config_SysTimer	12	12
SystemCoreClockUpdate	0	0
OS_InitKern_VFP	8	24
OS_StartASM	0	0
OS_Deactivated	16	24
OS_SignalEvent	8	32

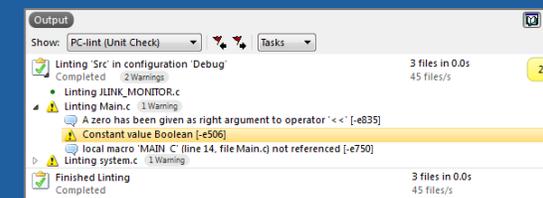
アプリケーションのスタック解析を行い、各スタックの使用量を表示

静的コード解析



静的コード解析で問題の可能性のあるコードを検索し表示

外部ツール連携



外部の構文解析ツールなどを組み込んで利用する事が可能。PC-lintなどを組み込みMISRA-Cチェックを実施できます。

Build



SEGGER Compiler

SEGGER Linker



emRun



emRun++



emFloat

SEGGER Compiler

独自開発のコンパイラ、モダンなフロントエンドを備えて、高い生成効率を実現します。



- 構成フロントエンド、C / C ++言語標準をサポート
- すべてのArmコアサポート
- Thumb-1/Thumb-2 CPU のコード生成が大幅に改善、パフォーマンスの向上とコードサイズを削減
- 言語拡張はclangと互換性があり、GCCと高い互換性

コードサイズ優先ベンチマーク比較

size optimization (-Oz)

Test Case	Code Size		
	SEGGER Compiler 18.1.0	Clang 18.1.6	GCC 13.2.1
CRC	152 bytes = 100.00 %	166 bytes = 109.2 %	164 bytes = 107.9 %
CompressToGo	2,096 bytes = 100.00 %	2,308 bytes = 110.1 %	2,124 bytes = 101.3 %
AES-CBC	1,768 bytes = 100.00 %	1,924 bytes = 108.8 %	2,120 bytes = 119.9 %
SHA-256	926 bytes = 100.00 %	978 bytes = 105.6 %	986 bytes = 106.5 %
RSA Sign	4,996 bytes = 100.00 %	5,364 bytes = 107.4 %	5,318 bytes = 106.4 %
FS Test	42,804 bytes = 100.00 %	43,544 bytes = 101.7 %	45,800 bytes = 107.0 %
coremark	2,496 bytes = 100.00 %	2,700 bytes = 108.2 %	2,734 bytes = 109.5 %
emBench	40,654 bytes = 100.00 %	48,292 bytes = 118.8 %	52,662 bytes = 129.5 %
Average	100.00 %	109.8 %	116.7 %



無償フル機能評価版で
サイズ制限なく
性能評価可能

実行速度優先ベンチマーク比較

コードサイズを抑制しながら、高速な実行速度を実現



Test Case	Run Time		
	SEGGER Compiler 18.1.0	Clang 18.1.6	highest optimization (-O3)
CRC	100.00	103.0	110.4
CompressToGo	100.00	110.3	93.2
AES_CBC	100.00	106.2	122.5
SHA256	100.00	107.0	112.7
RSA_Sign	100.00	105.8	107.5
FS_TEST	100.00	116.8	102.2
coremark	100.00	100.9	91.7
emBench	100.00	104.7	110.5
Average	100.00	106.8	106.3



並列コンパイルサポート

マルチコアを活用したコンパイルで、高速なビルドを実現します。

1 Building Thread	41 sec. (10 files/sec.)
8 Building Threads	13 sec. (31 files/sec.)

400 source files in 6 projects, Windows 7 64-bit



無償フル機能評価版で
サイズ制限なく
性能評価可能



SEGGER Linker

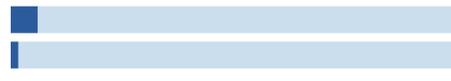
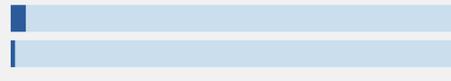
メモリアイアウトとシンボル配置の設定を容易にするだけでなく、コードサイズを大幅に縮小し、アプリケーションの実行速度を向上させることもできます。

- 大規模なアプリケーションにも対応できる超高速リンクを実現
- 柔軟性の高い配置アルゴリズムにより、固定領域の周囲にコードとデータ自動配置
- RAMベースのデータとコードのオプションの圧縮により、ROMの使用量を最小化

SEGGER Linkerは他のリンカーと比較して、より小さく、より高速なアプリケーションを生成するための機能と最適化を実現します。

SEGGER Linkerの最適化はCソースファイルからの入力だけでなく、すべてのコンパイル段階で機能します。また他コンパイラのオブジェクトコードもリンク可能です。SEGGER Linkerのほとんどの機能は、ターゲットアーキテクチャに応じて適用されます。すべてのアプリケーションでSEGGER Linkerで最適化を実現します。

高速なリンク実行時間

Project	GNU Linker		SEGGER Linker		Speedup	
Complex Project (SSL Web Server)	2.720s		0.466s		x5.8	
Medium Project (embOS Blinky)	0.164s	1.751s ¹	0.042s	0.329s ¹	x3.9	
Small Project (Simple main return)	0.088s	1.717s ¹	0.026s	0.322s ¹	x3.4	

emRun / emRun++

お客様アプリケーションを最適化するランタイムライブラリ

GCC や LLVM を使用するほとんどのツールチェーンは、newlib、newlib-nano、glibc のいずれかを使用しています。これらのライブラリには、組み込みシステム用のプロフェッショナルなランタイムライブラリに比べて大きな欠点があります。

Embedded Studioで提供されているSEGGER開発の「emRun Cランタイムライブラリ」は高度な最適化を可能にしています。これにより組み込みマイコンに最適なコード生成をサポートします。

- アセンブリコードで開発され、タイムクリティカルな性能を実現
- コードサイズ大幅な削減
- 高速または小型サイズに構成可能
- EABI互換機能
- 最小 RAM 使用量
- ヒープ要件なし

C++アプリケーション開発には、C++ライブラリ「emRun++」を利用頂けます。



SEGGER emFloat 浮動小数点ライブラリ

emFloatはデフォルトの浮動小数点ライブラリを置き換えることができ、より少ないコードでより優れたパフォーマンスを提供します。非常に高速で非常に小さいため、純粋なソフトウェアでFPUのようなパフォーマンスを提供します。emFloatはEmbedded Studioで利用頂けます。



emRun パフォーマンスベンチマーク

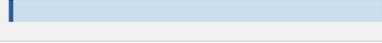
GCC / newlibとの比較で大きな改善を実現



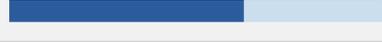
RISC-V (32bit)

Test Project ^[1]	Runtime [Cycles]		
	emRun ^[2]	gcc / newlib ^[3]	
Floating-Point Arithmetic ⓘ	1097.4	1907.0	
Floating-Point Math ⓘ	714.03	1814.36	
Integer Arithmetic ⓘ	251.2	1041.77	
String and Memory Functions ⓘ	17141	20733	

RISC-V (64bit)

Test Project ^[1]	Runtime [Cycles]		
	emRun ^[4]	gcc / newlib ^[5]	
Floating-Point Arithmetic ⓘ	826.8	1684.4	
Floating-Point Math ⓘ	734.71	1712.70	
Integer Arithmetic ⓘ	19.0	1560.6	
String and Memory Functions ⓘ	10430	17633	

Arm (32bit)

Test Project ^[1]	Runtime [Cycles]		
	emRun ^[6]	gcc / newlib ^[7]	
Floating-Point Arithmetic ⓘ	821.0	2618.8	
Floating-Point Math ⓘ	544.32	1003.35	
Integer Arithmetic ⓘ	122.3	195.7	
String and Memory Functions ⓘ	10696	17332	



Debug

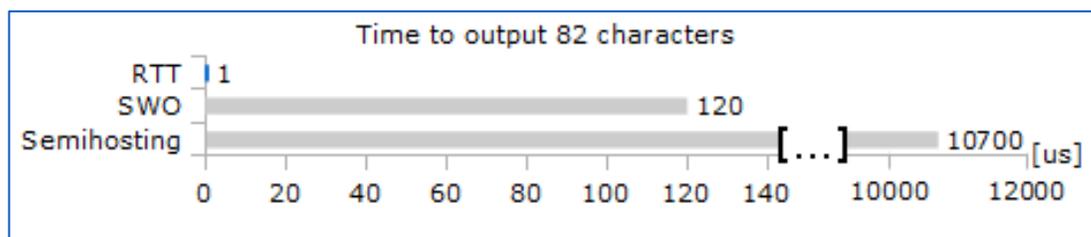
 デバッガ

RTT

J-Link独自技術、高速でマイコン負荷の少ない独自データ取得・送信インターフェース



- ターゲットアプリケーションとの双方向通信
- リアルタイム動作に影響を与えない高速データ転送
- 通信にデバッグチャンネルを使用します
- ターゲットに追加のハードウェア、ピン接続不要
- ARM Cortex-A/R/M、RISC-V、Renesas RXをサポート
- 機能性と自由度を提供する完全な実装コード

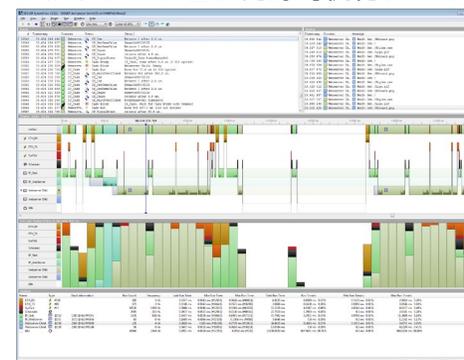


J-Link RTTを使用すると、ターゲットマイクロコントローラから情報を出力することができます。ターゲットのリアルタイム動作に影響を与えることなく、入力情報をアプリケーションに高速送信します。

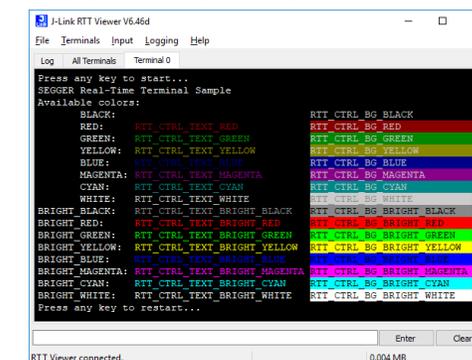
Embedded Studio以外のソフトウェアツールでもRTT利用し、アプリケーション解析が可能

リソース	使用量
ROM	500 Byte以下
RAM	24Byte + バッファ24Byte /チャンネル

SystemView
アプリケーション記録可視化ツール



RTTViewer
実行中のデバッグ セッションと並行して利用



Embedded Studio デバッガ

J-Link, J-Link RTTと統合して生産性の高いデバッグ作業を実現



ソースコードエディタ

```

Main.c
int main()
46 /*
47 *
48 *   main()
49 *
50 *   Function description
51 *   Application entry point
52 */
53 int main(void) {
54     OS_InitKern();           /* Initialize OS */
55     OS_InitHW();           /* Initialize Hardware for OS */
56     BSP_Init();           /* Initialize LED ports */
57     BSP_SetLED(0);        /* Initially set LED */
58     /* You need to create at least one task before calling OS_Start() */
59     OS_CREATETASK(&TCB0, "MainTask", 100, Stack0);
60     OS_Start();           /* Start multitasking */
61     return 0;
62 }
63
  
```

デバッグモードで利用するソースコードエディタ。アプリケーション停止位置の表示。ブレイクポイントマークをユーザは簡単に設定することができます。アプリケーションを停止した後のソースコード編集もデバッグモード中に可能

ターミナル

```

Debug Terminal
Hello World from

Embedded Studio

*****
SEGGER Microcontroller GmbH
The Embedded Experts
www.segger.com
*****
  
```

デバッグターミナルに、printf()やRTTで出力されたアプリケーション結果が出力されます。ANSIエスケープシーケンスを使用し、出力されるテキストや背景色などを変更することができます。J-Link RTT, SWO, DCC及びセミホスティングからのデータ出力をサポート

シンボルウォッチ

Expression	Value	Address	Size	Type
GLOBAL_IPRCnt	0xffff76e	0x200073d8	4	int
GLOBAL_Q_License	<struct>	0x200073a0	40	OS_Q_STRUCT
GLOBAL_SDCardPresent	0x00000001	0x200073d4	4	int
GLOBAL_Time	<struct>	0x200073cc	8	IP_NTP_TIMESTAMP
Seconds	3715339249	0x200073cc	4	long unsigned int
Fractions	381768024	0x200073d0	4	long unsigned int
index_file	"<!DOCTYPE HTML	0x0006a6c3	128	const unsigned char[7903]
[0]	0x3c	0x0006a6c3	1	unsigned char
[1]	0x21	0x0006a6c4	1	unsigned char
[2]	0x44	0x0006a6c5	1	unsigned char
[3]	0x4f	0x0006a6c6	1	unsigned char
[4]	0x43	0x0006a6c7	1	unsigned char

変数の監視に利用されるウォッチウィンドウローカルウィンドウ：現在の関数とローカル変数、パラメータを表示することができます。

グローバルウィンドウでは、グローバル変数オートウィンドウでは、コンテキストで重要な値を表示可能

Embedded Studio デバッガ

J-Link, J-Link RTTと統合して生産性の高いデバッグ作業を実現



スレッド

Pri	Id	Name	Status	Timeout	Stack info	Run
Task List						
0x2000B904	IP_Task	Wait for Task Event with Timeout	26053	940 / 1280 @ 0x2000B95C	2867	
0x200087F0	WEBS_Task	Wait for Event Object with Timeout	29123	888 / 3072 @ 0x20008848	7	
0x20009448	SMTPC_Task	Wait for Queue (Get) with Timeout	29123	448 / 6144 @ 0x200094A0	7	
0x2000ACA4	MTP_Task	Delay	26134	1572 / 3072 @ 0x2000ACFC	730	
0x20006748	MainTask	Executing	26043	2168 / 3072 @ 0x200067A0	152	

RTOSのスレッド表示をすることが可能です。アプリケーションのタスクスレッドを表示する事が可能です。スレッドをダブルクリックすることで、利用しているレジスタとコールスタックを表示する事が出来ます。SEGGER embOSとFreeRTOSはデフォルトでサポートされていますが、JavaScriptを利用して未対応のRTOSの情報を表示することも可能

レジスタ

Name	Value
sp(r13)	0x20007378
lr(r14)	0x20006618
pc(r15)	0x0000b71a
apsr	0x21000000
GE	0x0
Q	0 False
V	0 False
C	1 True
Z	0 False
N	0 False
SysTick	
SYS_CSR	0x00010007
SYS_RVR	0x0002903f
SYS_CVR	0x000267e2
SYS_CALIB	0x00000000

CPUレジスタ並びにメモリマップレジスタを表示する事ができます。レジスタは様々な数値形式で表示、直接レジスタの値を変更することも可能です。EmbeddedStudioでは、最大4つのレジスタウィンドウを表示する事ができます。

シンボルウォッチ

Function	Call Address	Stack Pointer	Frame Size	Stack Used
void FS_FindVolume(char * pFullName=mmc0; ...	0x00020F22	0x200072C8	56 bytes	216 bytes
int FS_GetVolumeStatus(char * sVolume=mmc0; ...	0x0002E122	0x20007300	32 bytes	160 bytes
int _FSCheckSD()	0x0000BAE5	0x20007320	64 bytes	128 bytes
long unsigned int _FSCheckPeriodic(long unsi...	0x0000E8D2	0x20007360	24 bytes	64 bytes
void _MainTask()	0x0000B72E	0x20007378	40 bytes	40 bytes
OS_StartTask()	0x0000E8E4	0x200073A0	0 bytes	0 bytes

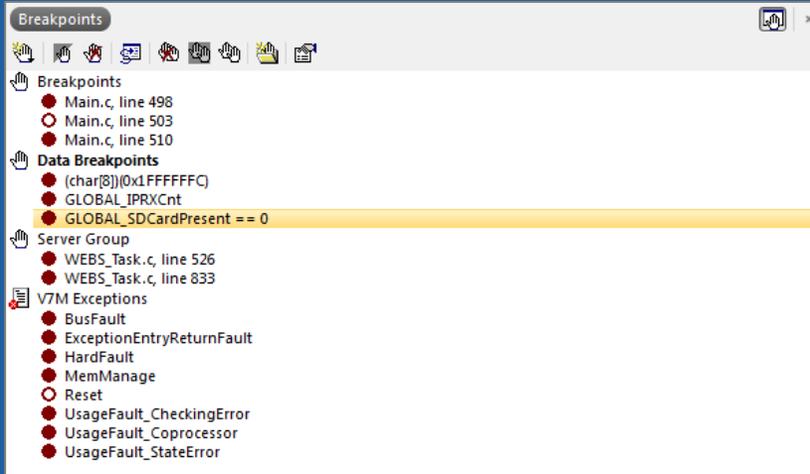
コールスタックウィンドウでは、アプリケーションが停止時の機能とその呼び出し元が表示されます。各コールをダブルクリックして、コールの正確な位置情報を取得することが可能

Embedded Studio デバッグ

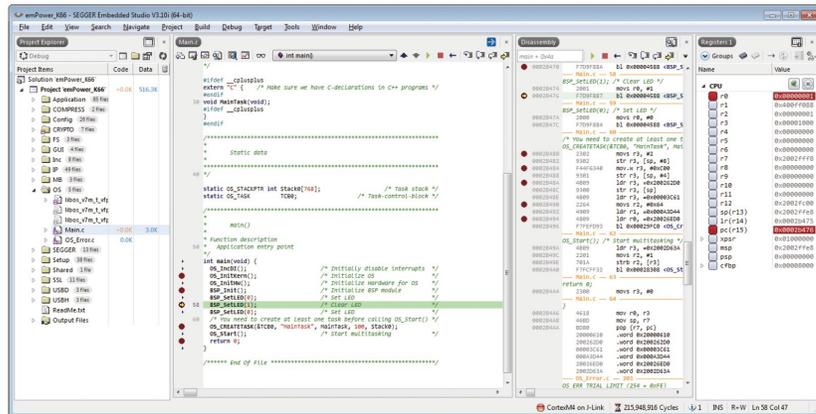
J-Link, J-Link RTTと統合して生産性の高いデバッグ作業を実現



ブレイクポイント



RTOSのスレッド表示をすることが可能です。アプリケーションのタスクスレッドを表示する事が可能です。スレッドをダブルクリックすることで、利用しているレジスタとコールスタックを表示する事が出来ます。SEGGER embOSとFreeRTOSはデフォルトでサポートされていますが、JavaScriptを利用して未対応のRTOSの情報を表示することも可能



J-Link Flash BP

マイコンのブレイクポイント数を無制限に設定可能な拡張機能 (J-Link PLUS 以上)

「J-Link Unlimited Flash-BP」拡張機能を使うことで、ユーザーはフラッシュメモリでデバッグするとき、無制限のブレイクポイントを設定できます。この機能がないデバッガでは、フラッシュに設定できるブレイクポイント数は、CPUのデバッグユニットでサポートされているハードウェアブレイクポイントの数に制限されます。(ARM 7/9 では2、Cortex-M では4-6)

対応する統合開発環境・デバッグソフトウェア

SEGGER 社製 : J-Link Debugger 「Ozone」 / SEGGER Embedded Studio

他社製 : J-Link DLL、J-Link RDI 接続をサポートする統合開発環境 (IAR EWARM / Arm MDK-ARM など)



License



製品ライセンスモデル

Embedded Studio arm

Arm License

対応CPU : ARM7、ARM9、Cortex-M0/M0+/M1/M3/M4/M7/M23/M33、Cortex-A、Cortex-R

Cortex-M License

対応CPU : Cortex-M0/M0+/M1/M3/M4/M7/M23/M33

RISC-V

RISC-V License

対応CPU : RV32I、RV32IMA、RV32IMAC、RV32IMAF、RV32IMAFIC、RV32G、RV32GC

必要なRTOS/ミドルウェアをパッケージ



Embedded Studio PRO for Cortex-M

Cortex-M License

対応CPU : Cortex-M0/M0+/M1/M3/M4/M7/M23/M33

【同梱ハードウェア】
J-Link PLUSデバッグプローブ
J-Link 19-pin Cortex-M Adapter
emPower評価ボード (Cortex-M4コア)

【同梱ソフトウェアモジュール】
Embedded Studio - Cortex-M edition
embOS : リアルタイムOS
emCrypt PRO : 組込用暗号ライブラリ
emFile PRO : ファイルシステム
emNet PRO : TCP/IPネットワーク
emSSH : セキュアシェール、セキュアログイン
emSSL : セキュア通信 (TLS1.2)
emUSB-Device PRO : USBファンクション
emUSB-Host PRO : USBホスト
emWin PRO : GUIソリューション
IoT Toolkit : IoTソリューション

Embedded Studio ライセンス

柔軟な運用を可能にするライセンスモデル

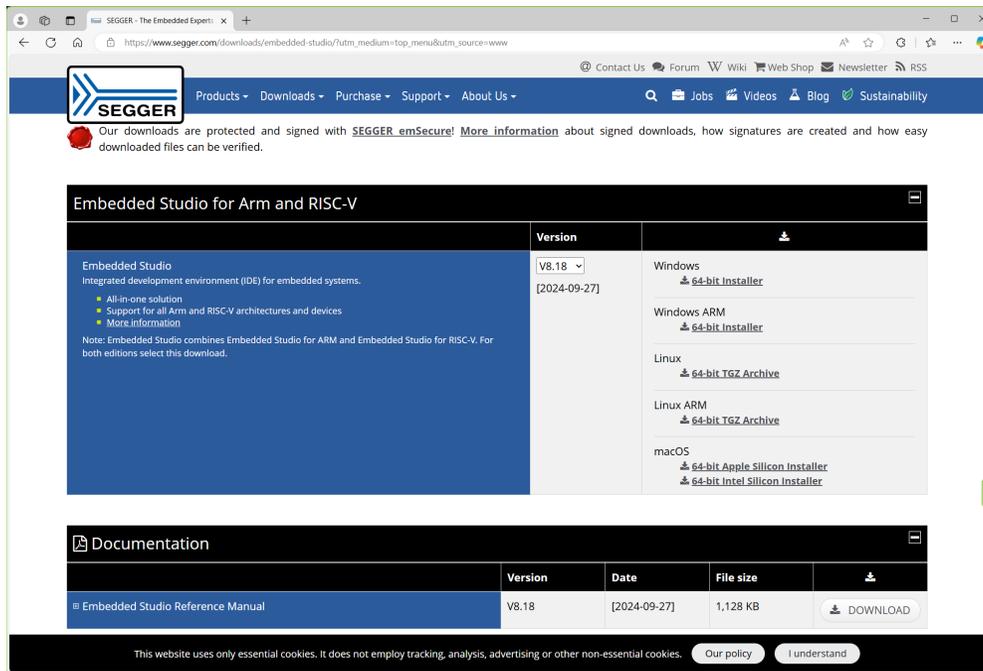
フェーズ	ターゲットボード	利用可能デバッガ	機能制限	ライセンス
評価	各種評価ボード	J-Link LITE J-Link on-board J-Link BASE (Compact) J-Link PLUS (Compact) J-Link ULTRA+ J-Link PRO J-Trace PRO	フル機能	無償評価版
商用開発	各種評価ボード 各種開発ボード 自社開発量産ボード	J-Link BASE (Compact) J-Link PLUS (Compact) J-Link ULTRA+ J-Link PRO J-Trace PRO	フル機能	商用開発 ライセンス



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。

利用期間制限なし・機能制限なし無償評価版
非商用利用・学術用途利用は無償のままご利用頂けます。

Embedded Studio

The screenshot shows the SEGGER website's download page for Embedded Studio. It features a navigation bar with links for Products, Downloads, Purchase, Support, and About Us. A main heading reads 'Embedded Studio for Arm and RISC-V'. Below this, there is a table with columns for Version and Date, showing V8.18 and [2024-09-27]. To the right, there are download links for various operating systems: Windows (64-bit Installer), Windows ARM (64-bit Installer), Linux (64-bit TGZ Archive), Linux ARM (64-bit TGZ Archive), and macOS (64-bit Apple Silicon Installer and 64-bit Intel Silicon Installer). A 'Documentation' section at the bottom shows a table with columns for Version, Date, File size, and File name, listing the Embedded Studio Reference Manual (V8.18, [2024-09-27], 1,128 KB) with a 'DOWNLOAD' button.

<https://www.segger.com/downloads/embedded-studio/>

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

ユーザー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

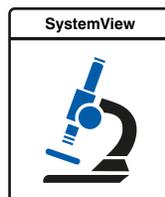
J-Linkで利用頂けるソフトウェアツールライセンス

デバッグソフトウェア

**OZONE(オゾン)**

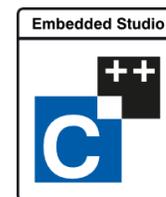
GCC/各種商用コンパイラで生成したELFファイルを利用可能

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/>

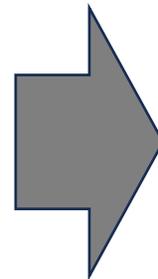
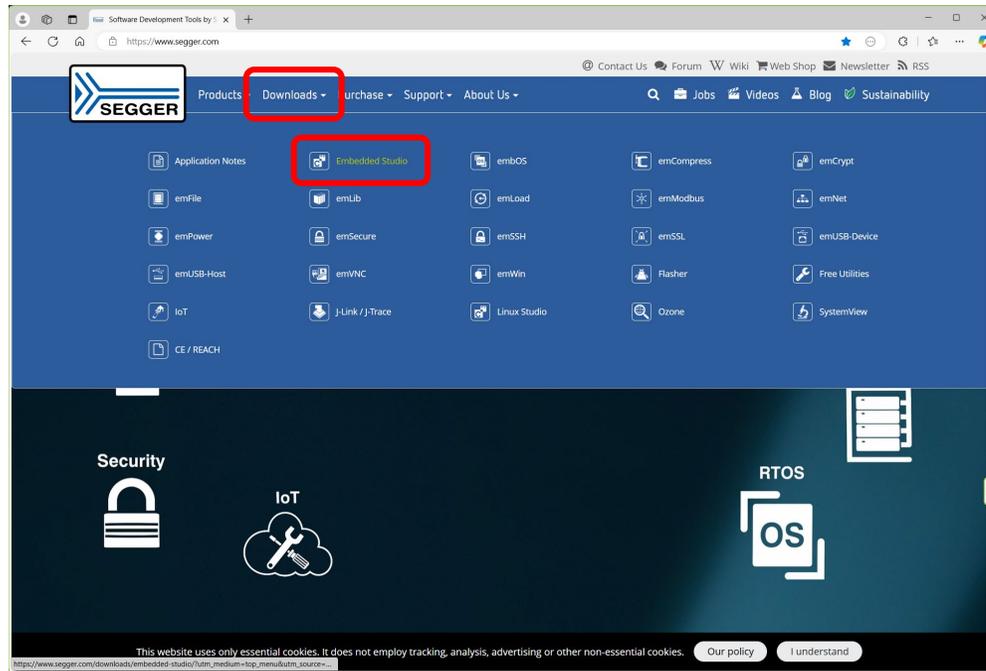
教育・学術・趣味でのご利用はライセンス費用は不要



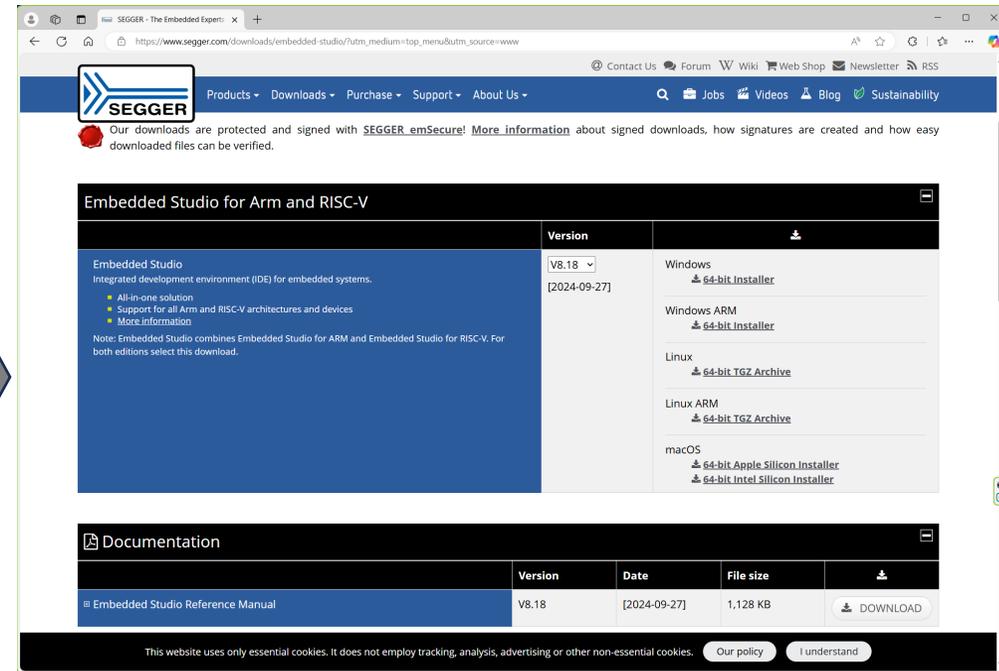
無償評価版の入手とインストール

Embedded Studio評価版ダウンロード

SEGGER社ウェブサイトへアクセス
グローバルメニューで [Download] > [Embedded Studio]を
選択



Embedded Studio インストーラをダウンロード
ご利用のPCに合わせたバージョンを選択

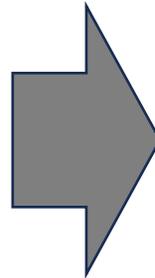
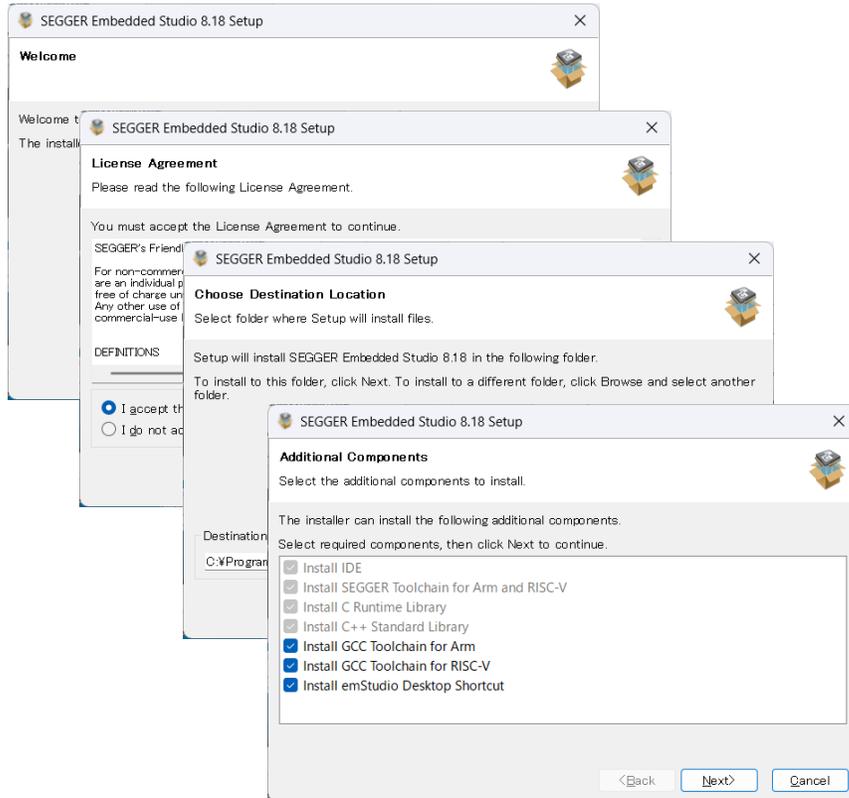


<https://www.segger.com/>

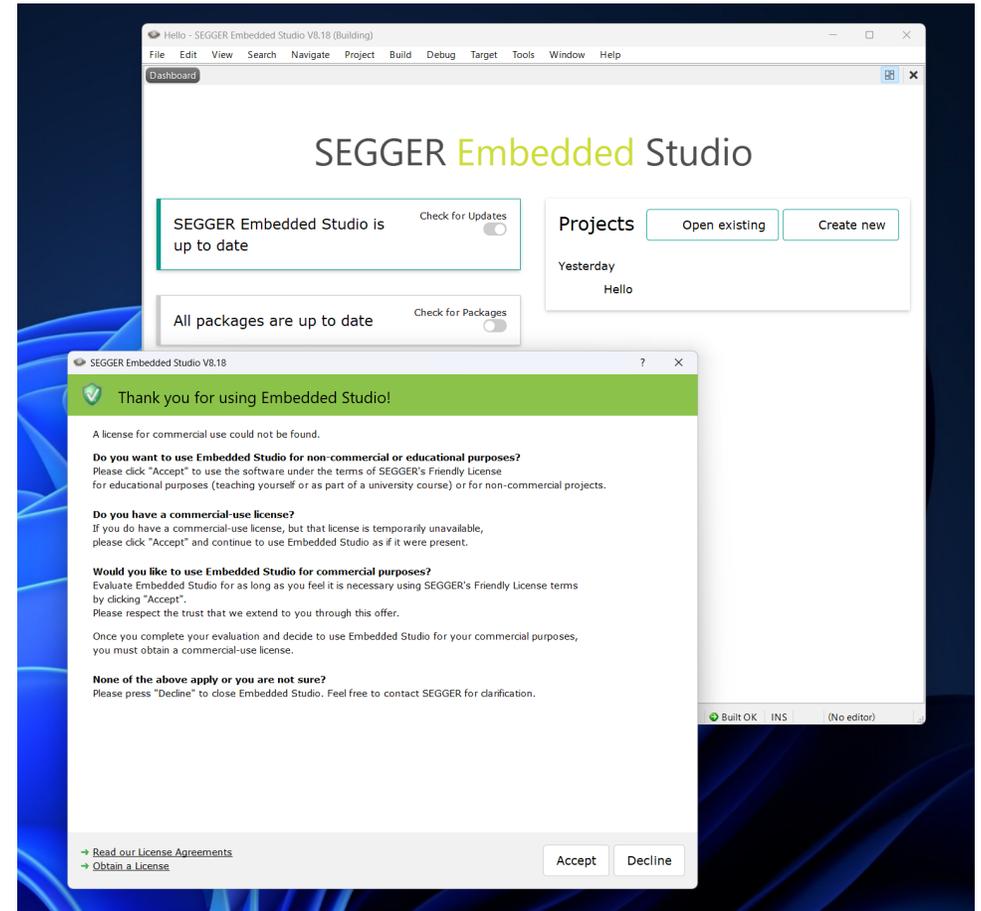
<https://www.segger.com/downloads/embedded-studio/>

Embedded Studio評価版インストール

インストーラの指示に従い、
Embedded Studio をインストール

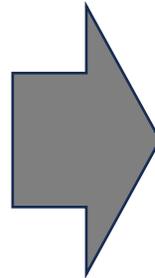
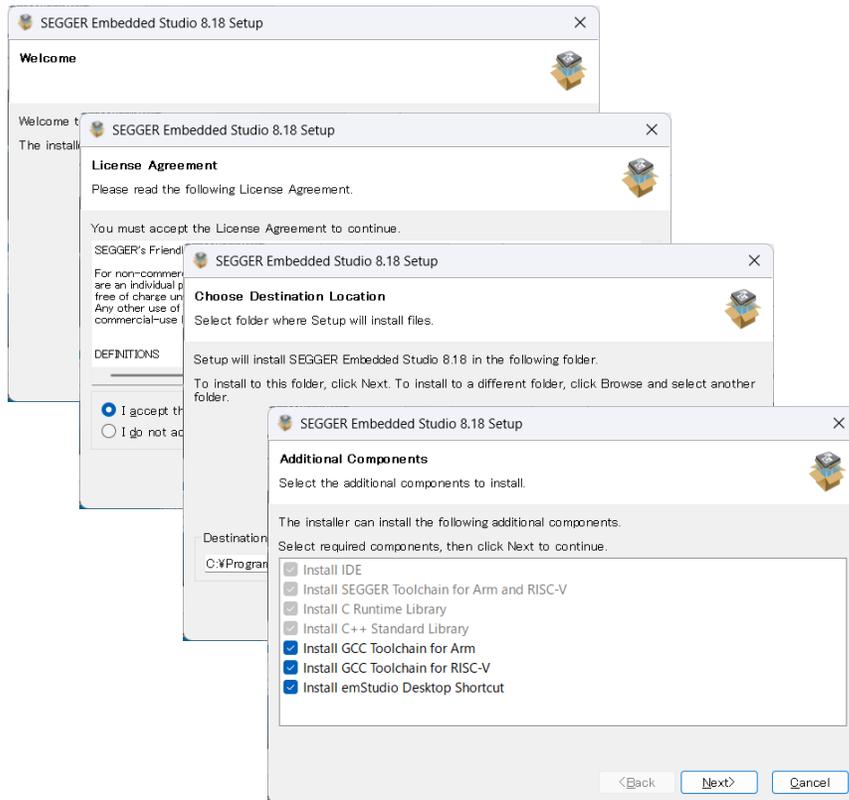


Embedded Studio を起動
SEGGER's Friendly Licenseの同意で利用可能に

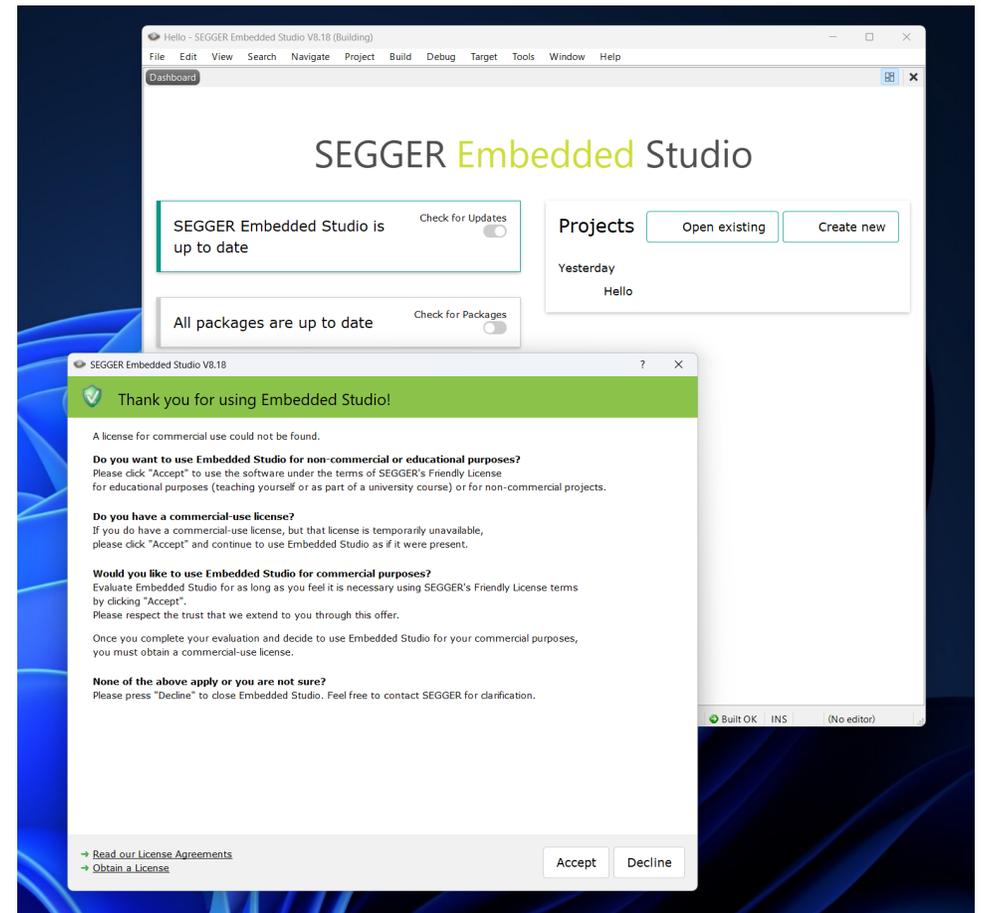


Embedded Studio評価版インストール

インストーラの指示に従い、
Embedded Studio をインストール



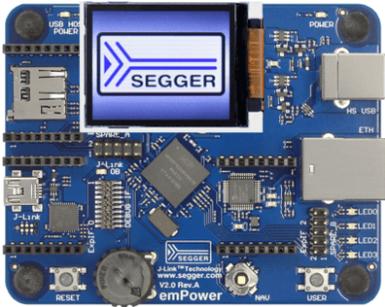
Embedded Studio を起動
SEGGER's Friendly Licenseの同意で利用可能に



SEGGER emPowerや各デバイスメーカー評価ボード

■ SEGGER emPower

Embedded Studioを利用して、すべてのソフトウェアコンポーネントを試用評価できるソリューションボード

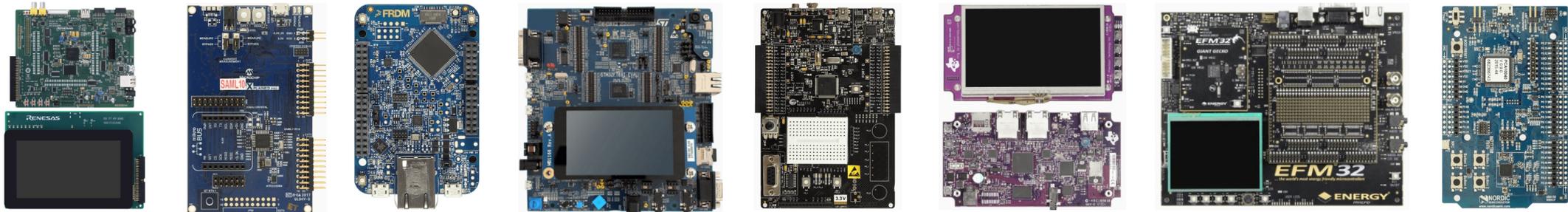


Cortex-M4 コアマイコン搭載

<https://www.embitek.co.jp/product/segger-evb.html>

■ 各デバイスメーカー評価ボード

各CPUメーカーより提供されている評価ボードでEmbedded Studioを活用し、様々な対応ソフトウェアコンポーネントを利用する事が出来ます。



<https://www.embitek.co.jp/download/evalsamples/>



SEGGER emPower評価ボード利用

ターゲットハードウェアなしでもシミュレータ利用できます。

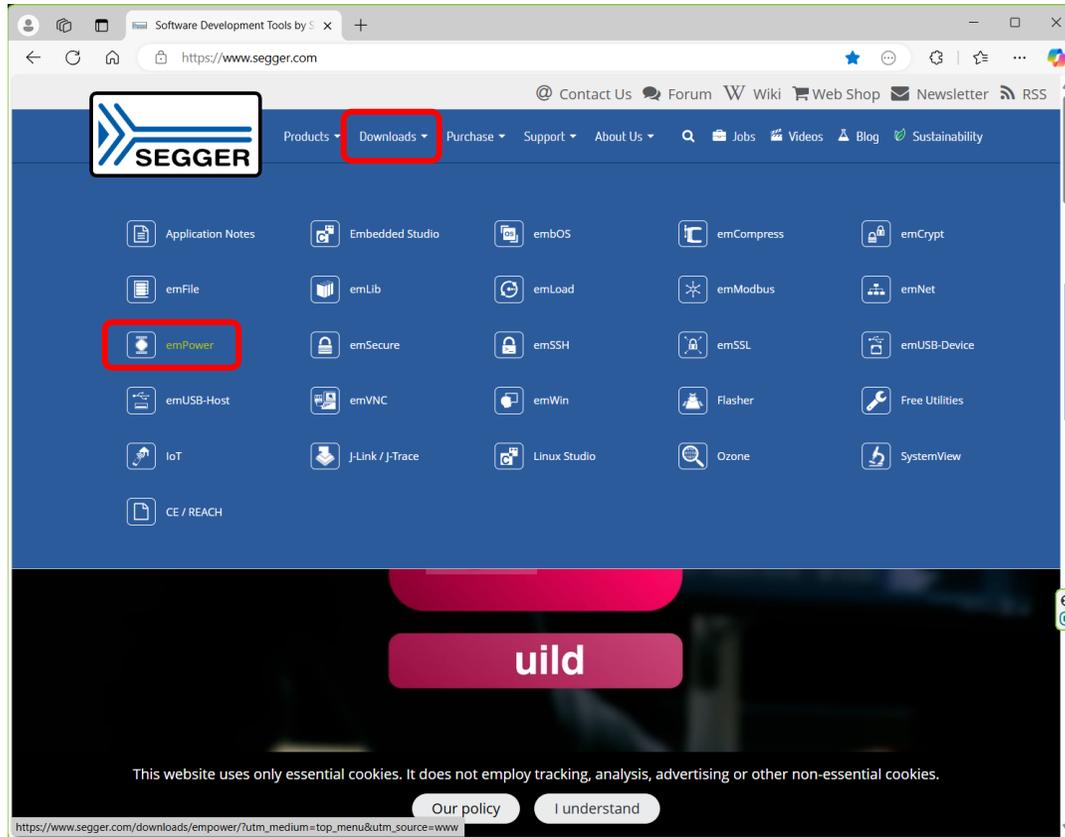


emPowerを利用するための簡単な準備作業

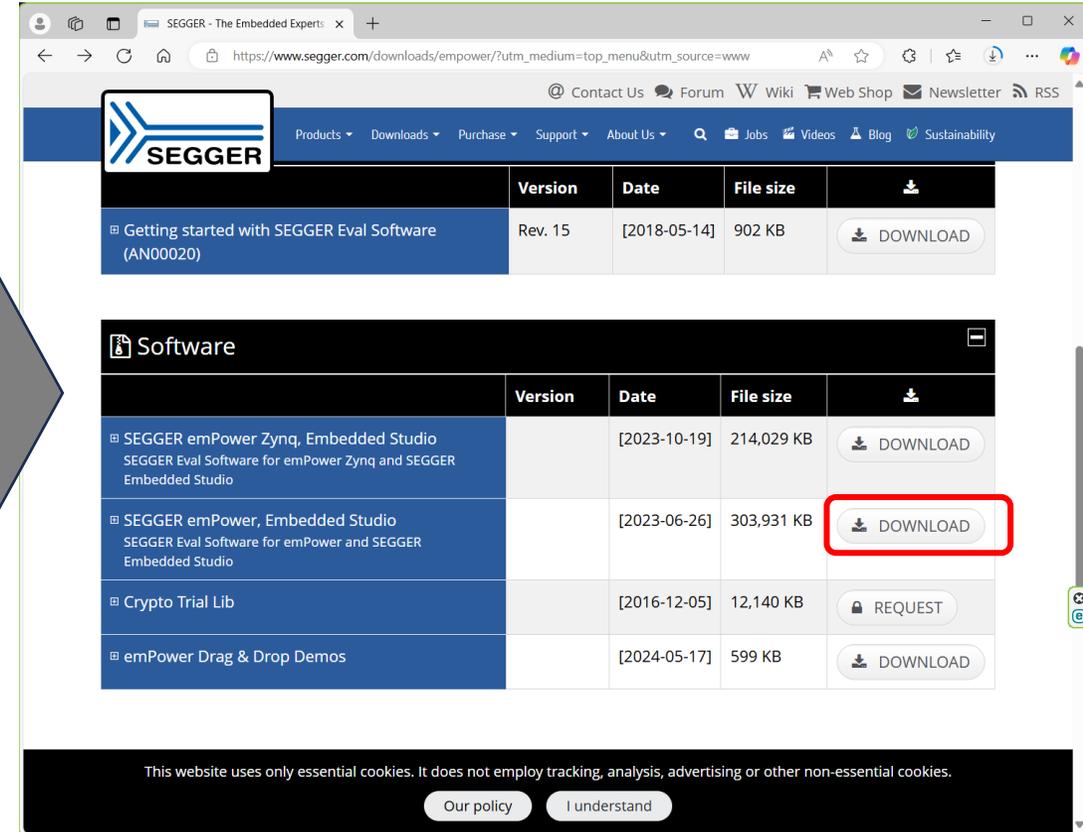
- emPowerの接続 (J-Link OB接続準備)
- SEGGER Embedded Studioインストール・立ち上げ
- emPowerサンプルソフトウェアのダウンロード

SEGGER emPowerサンプルアプリケーションをダウンロード

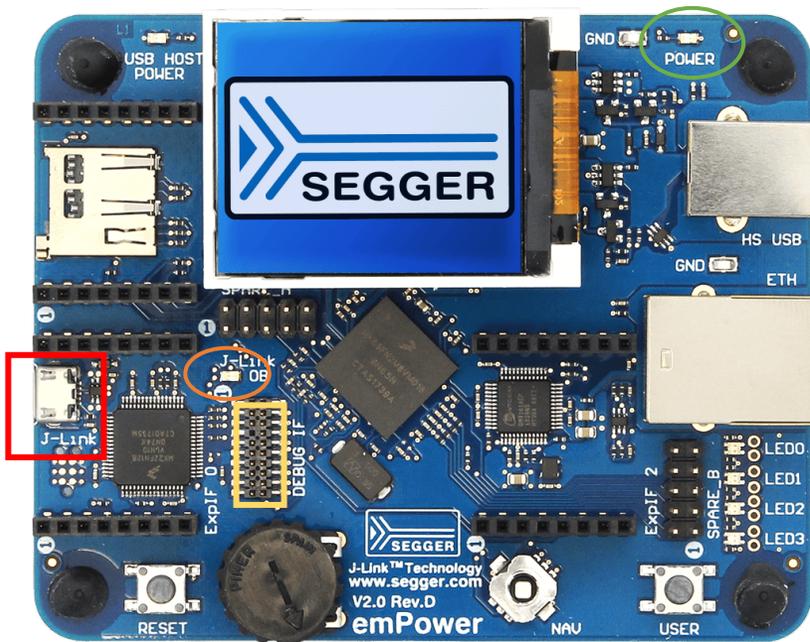
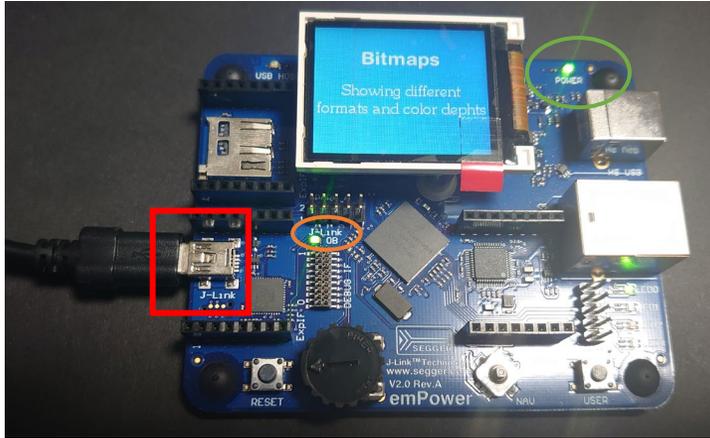
SEGGER社ウェブサイトアクセス
グローバルメニューで [Download] > [emPower]を選択



emPowerサンプルアプリケーションのダウンロード



emPower PC接続準備



- 付属のUSBケーブルでPCと接続してください。
- emPowerボードにUSB経由で電源が供給されている場合、[POWER]LED緑点灯します。
- ご利用のPCにJ-Linkドライバがインストールされている場合、[J-Link OB]LED緑点灯します。（通常SEGGER Embedded Studioのインストール時にインストールされます）

[J-Link OB]LEDが点灯しない場合は、J-LinkドライバがPCにインストールされていないので、以下URLの「J-Link ソフトウェア、JTAG/SWDコネクタ仕様、変換アダプタ：仕様説明書」の[1. J-Link ソフトウェアインストール]を参照頂き、ドライバをインストールしてください。

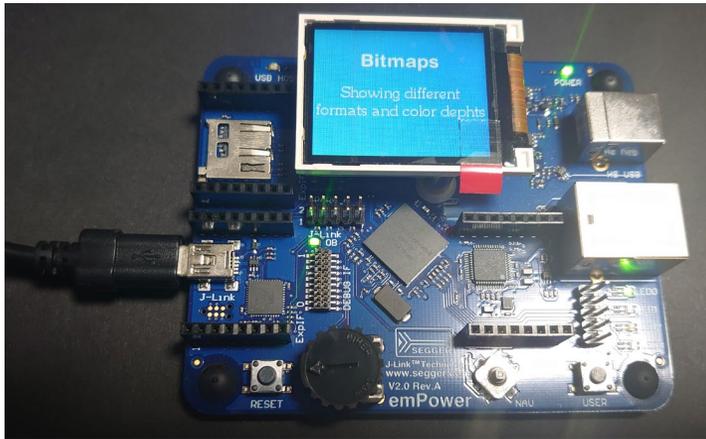
https://www.embitek.co.jp/technote/jlink/EUM_JP_JLink-Adapter.pdf

[POWER LED]と[J-Link OB]が緑点灯すれば、PCとの接続準備は完了です。

□ 製品版のJ-Linkをお持ちであれば、ハーフピッチ19pinコネクタを利用して、接続することも可能です。

emPower 「SEGGERDEMO」 サンプルアプリケーションを動かす

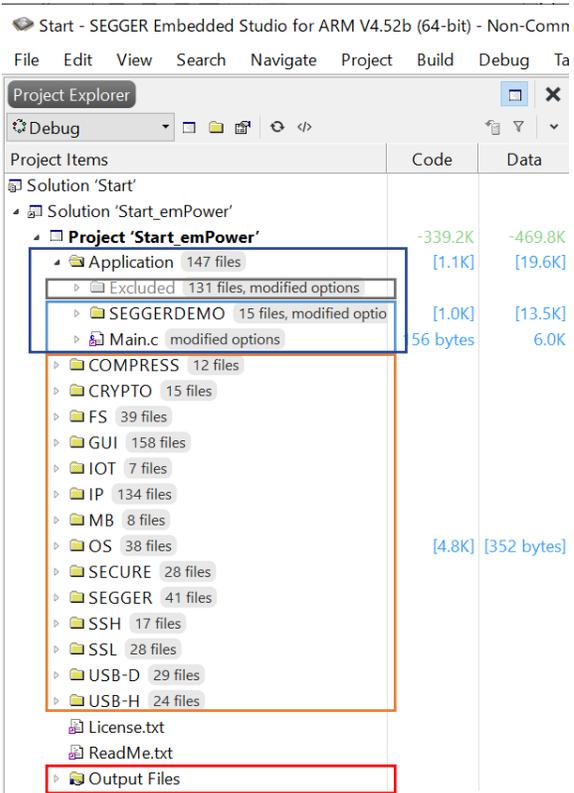
emPowerサンプルアプリケーションでデフォルト設定されている「SEGGERDEMO」サンプルは、RTOS,GUI,Filesystem,TCP/IP, USB-Host/Deviceなどが機能する統合型のサンプルになっています。



LCD液晶表示、
PCとのUSB接続、LANポート接続してのネットワーク接続などの
すべてが同時に機能しています。

まずは、このSEGGERDEMOサンプルをemPowerターゲットボードへダウンロード実行します。

emPower 「SEGGERDEMO」 サンプルアプリケーションについて



サンプルアプリケーションソースコード

- 「Main.c」, 「SEGGERDEMO」フォルダ →デフォルトでアクティブソースコード
- 「Excluded」フォルダ →デフォルトで非アクティブなソースコード（グレーアウトされています）

ソフトウェアライブラリ

- SEGGER製品の評価コンポーネント

(ライブラリはオブジェクトでの提供、各種設定ファイル・ヘッダファイル)
製品版では、オブジェクト提供されているものがソースコードで提供されます。

Output Files

- ビルド後の実行ファイル・マップファイルなどが出力されます。

まずは、「SEGGERDEMO」サンプルアプリケーションを動かしますので、
ファイル設定はこのままビルド・デバッグへ進みます。

emPower 「SEGGERDEMO」 サンプルアプリケーションを開く

① [File]→[Open Solution...]をクリックし、c:¥emPower¥Start.emProjectを指定します。

P.30でダウンロード解凍したフォルダになります。

Project Explorer (プロジェクトマネージャ) にサンプルプロジェクトが表示されます。

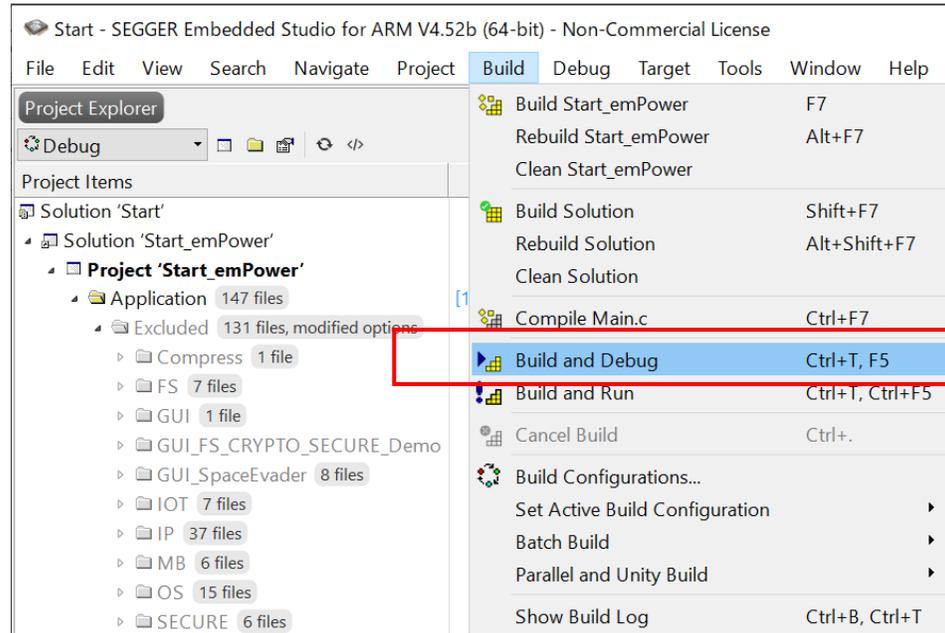
emPowerサンプルプロジェクトでは、3つのコンフィギュレーションが用意されています。

まず[Debug]を選択してください。

- Debug
- Release
- Release_SystemView
- <Edit Build Configurations...>

- Excluded 131 files, modified options
 - Compress 1 file
 - FS 7 files
 - GUI 1 file
 - GUI_FS_CRYPTO_SECURE_Dem
 - GUI_SpaceEvader 8 files
 - IOT 7 files
 - IP 37 files
 - MB 6 files
 - OS 15 files
 - SECURE 6 files
 - SSH 1 file
 - SSL 3 files
 - USB 10 files
 - USBH 7 files
 - SEGGERDEMO 15 files
 - Main.c
 - COMPRESS 12 files
 - CRYPTO 15 files

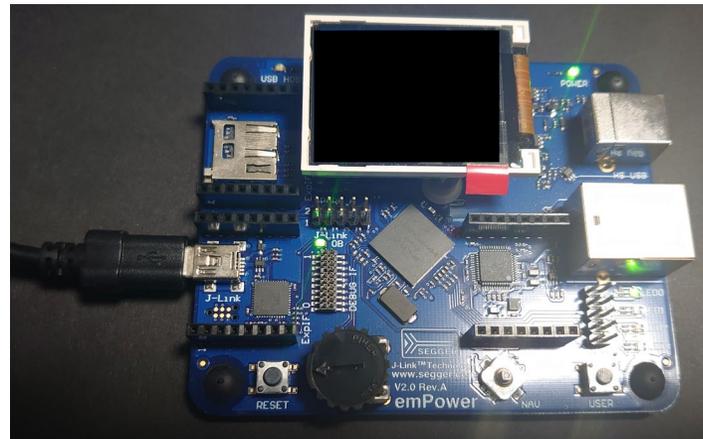
emPower 「SEGGGERDEMO」 サンプルアプリケーションをダウンロード



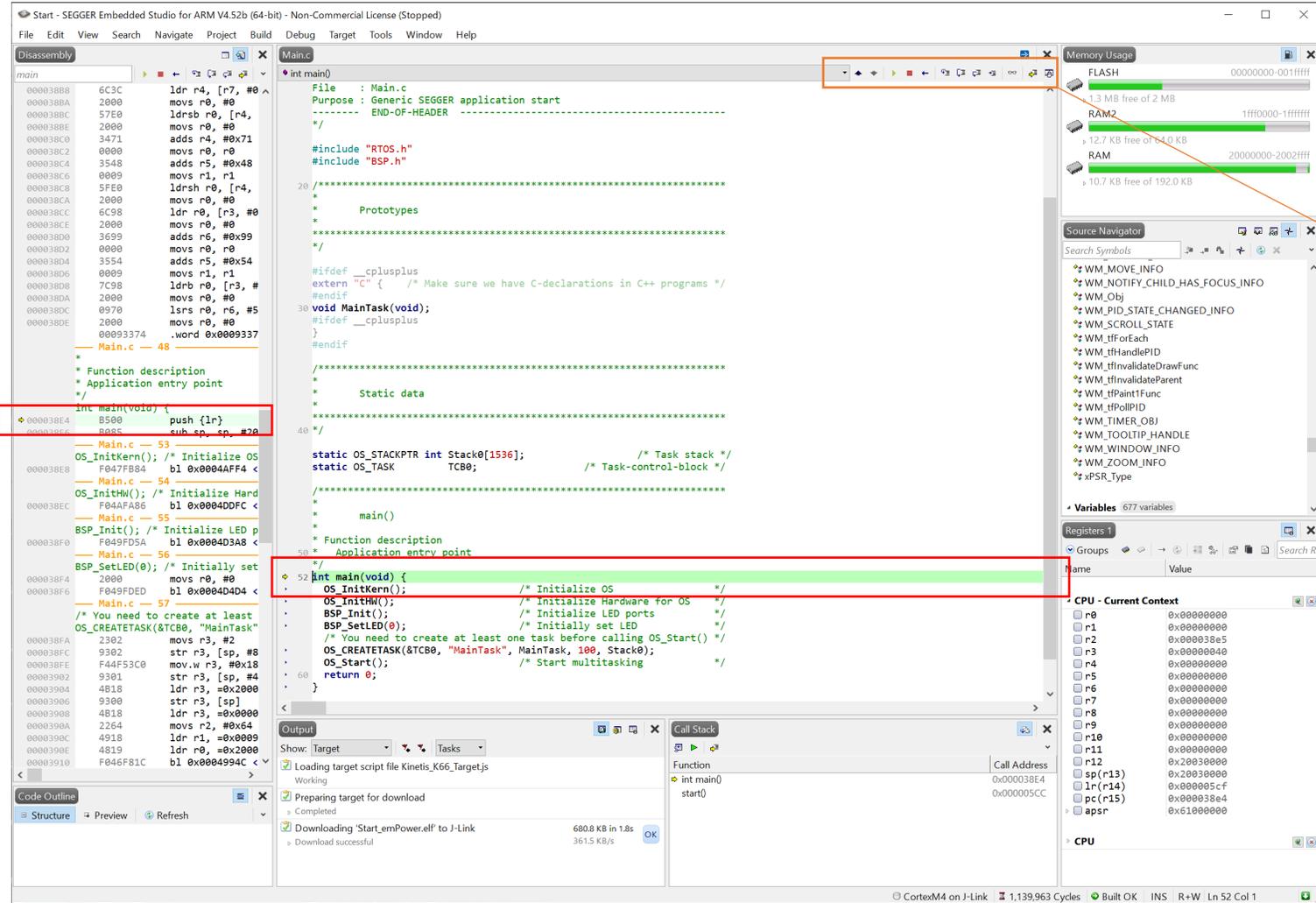
[Build]メニューから[Build and Debug]を選択することにより、アクティブになっているソースコードのコンパイル、リンク、USB接続をしたemPowerボードへJ-Link OB経由でダウンロードまで一括で行います。

まずはデフォルト設定から設定変更することなくemPowerボードへの書込まで問題なく進むことを確認してください。

ここでエラーが出る場合は、emPowerのUSB接続ができていない、J-Linkドライバが適切にインストールされていないなどが原因となります。



emPower 「SEGGGERDEMO」 サンプルアプリケーションをダウンロード



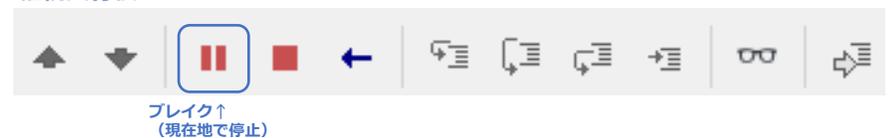
ビルドが完了し、デバッガが起動します。「Main.c」ファイルが開かれ、次に行われるMain関数（プログラムカウンタの位置）にマーク表示されます。

同様に[Disassembly]ウインドウでもプログラムカウンタの位置にマークされます。

デバッガの基本操作



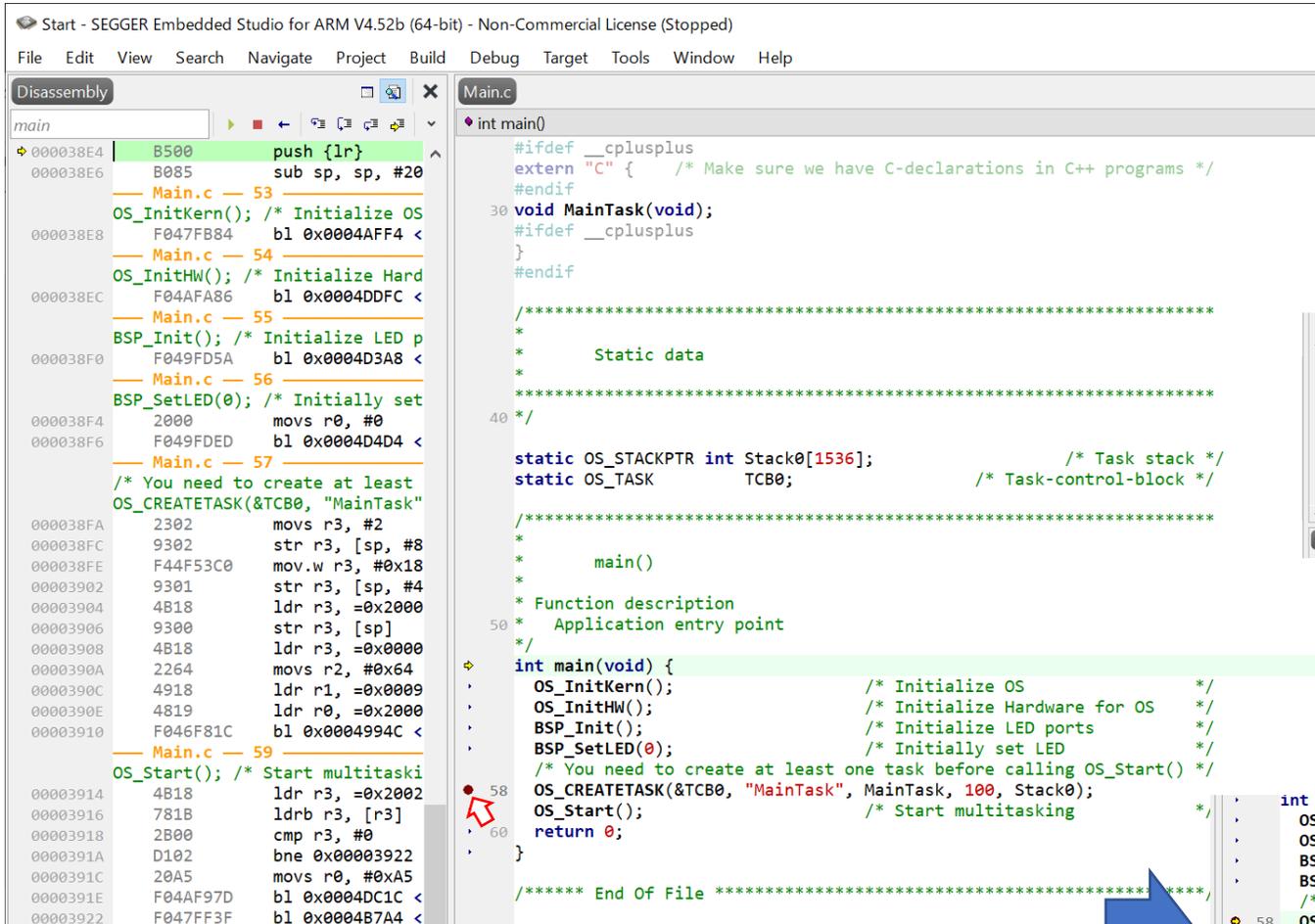
継続実行後



プログラムの実行（継続実行）すると emPowerでSEGGERデモが動作します。



ブレイクポイント

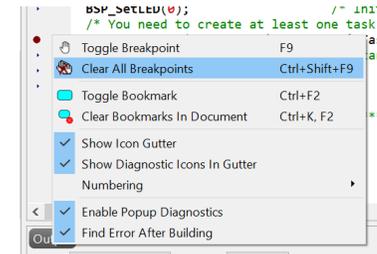


ブレイクポイントを設定した状態で「プログラムの実行（継続実行）」すると指定したポイントで停止します。



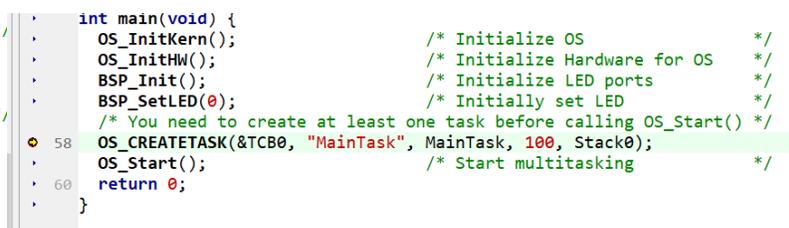
■ ブレイクポイントの設定

アプリケーション停止中に止めたいポイントでブレイクポイントを設定します。
該当行数⁵⁸をクリックすることで●マークされ、ブレイクポイントが設定されました。

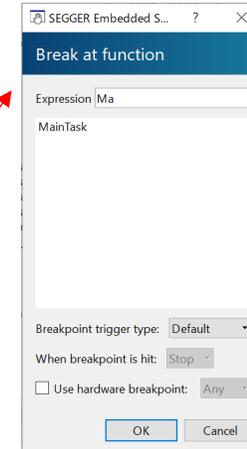
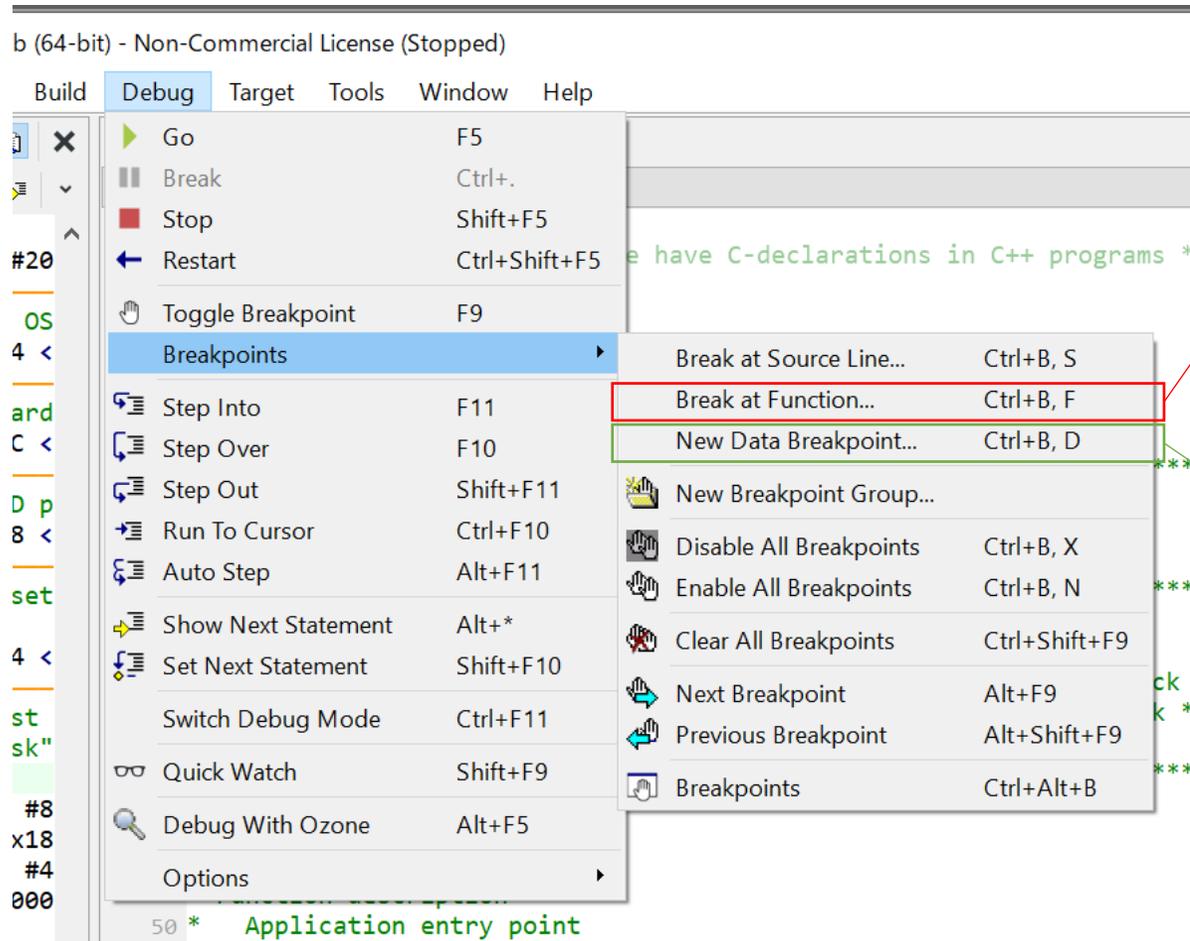


■ ブレイクポイントの解除

●マークをダブルクリックすることで、ブレイクポイントを解除できます。
※全てのブレイクポイントを解除したい場合は、右クリックのコンテキストメニューから「Clear All Breakpoints」で、一括解除できます。

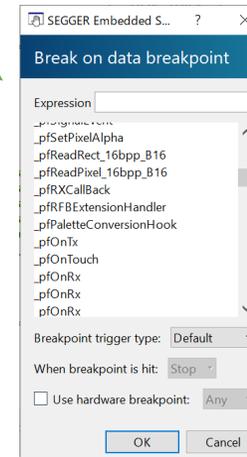


関数名・特定データでブレイクポイントを設定



関数名でブレイクポイント設定

プロジェクト内のファイルから該当する関数を検索し、表示します。ブレイクポイントを設定したい関数を選んで設定できます。

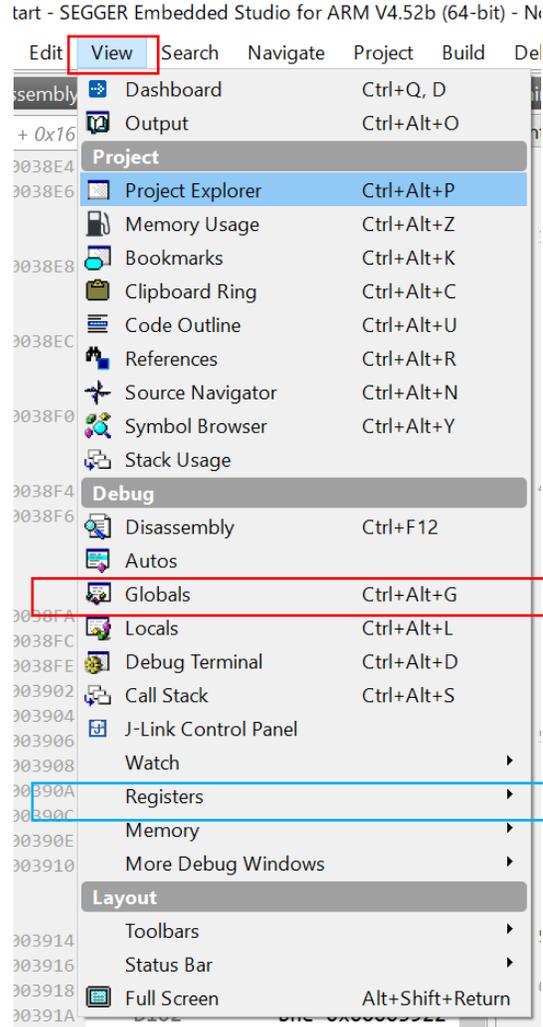


特定のデータでブレイクポイント設定

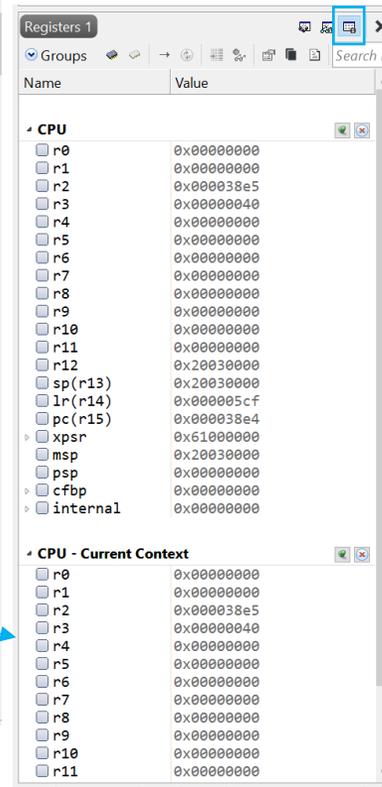
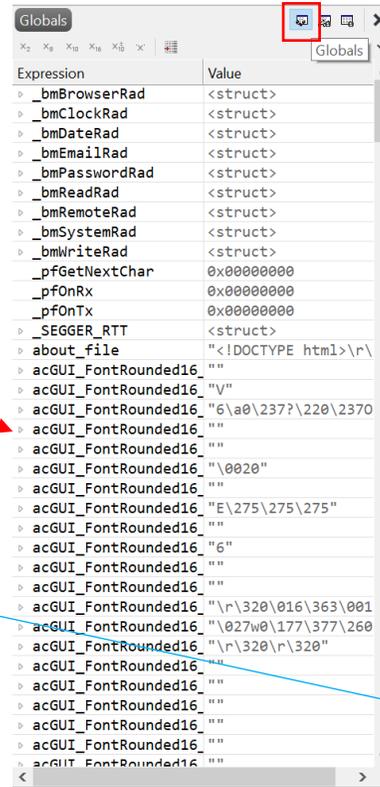
特定のデータで、ブレイクするように設定できます。

他、ブレイクポイントを設定したまま、一時的に無効化[Disable All Breakpoints] / 有効化する[Enable All Breakpoints]などの設定を本メニューから対応できます。

様々なデータを表示

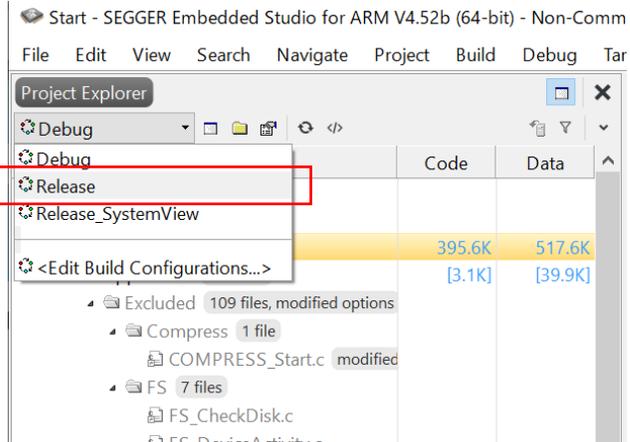


デバッガの様々な機能は、[View]メニューから呼び出します。呼び出した情報表示ウィンドウは、IDEの任意の位置に配置し、開発者様の利用しやすいようにカスタマイズできます。



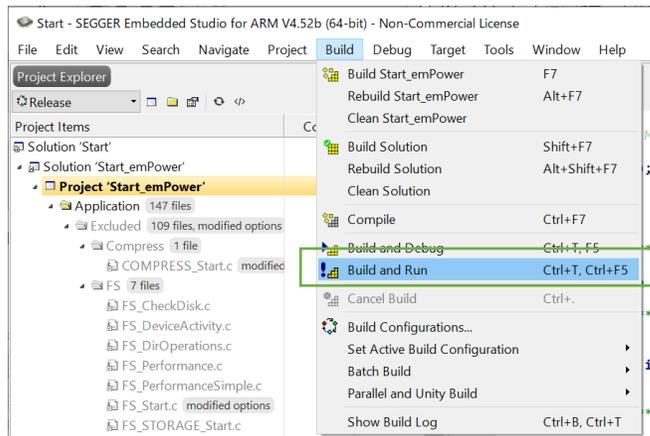
■ スタックされた情報表示画面の切り替え
 SEGGER Embedded Studio IDEで、情報表示ウィンドウを一つのエリアにスタックすることができます。スタックされたウィンドウを切り替える場合は、それぞれのアイコンをクリックすることで、切り替えることができます。また  アイコンをドラッグして、エリアの移動や独立したウィンドウ表示に切り替えられます。

プロジェクトマネージャでプロジェクトコンフィギュレーション設定

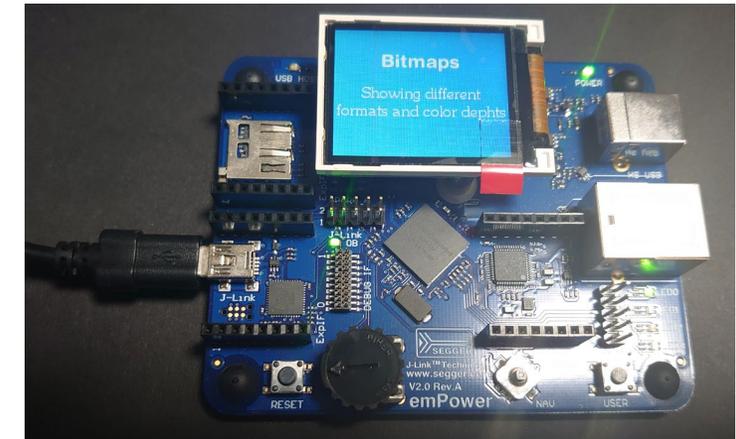


プロジェクトマネージャ（Project Explorer）で複数のプロジェクト設定を持つことができます。本サンプルプロジェクトでは、[Release]を選択するとリリースモードのプロファイルでビルドできるようになります。このプロファイルで、ビルド・ターゲットボードへのダウンロードを行うとデバッガを介することなく、電源投入によりプログラムが実行されます。

[Release]プロファイルでビルド・ダウンロードしたプログラムもデバッグ接続は可能ですが、デバッグ機能に制限が出るプロファイルとなります。



[Release]プロファイルでは、デバッガを立ち上げずに[Build]メニューから「Build and Run」を選択するとビルド・ダウンロードを実行し、ターゲットボードのプログラムがスタートします。





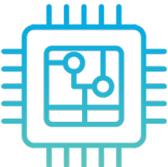
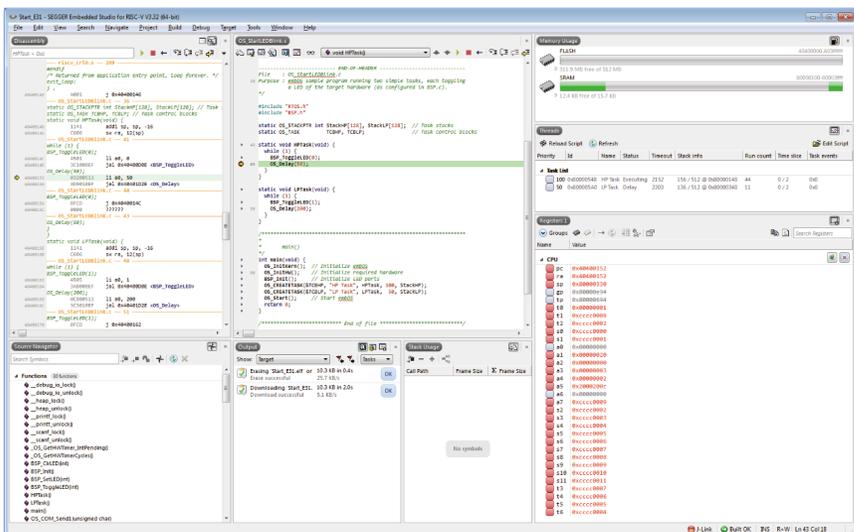
Embedded Studio PRO

統合開発環境

RTOS・ミドルウェアソフトウェアパッケージ



開発プロジェクト無制限で利用可能なソフトウェアプラットフォーム



Cortex-M

自由に変更可能



Object Code

高品質ソフトウェアモジュール

利用マイコン変更可能

高性能コンパイラ/リンカ同梱

無償ですべての機能を評価利用可能

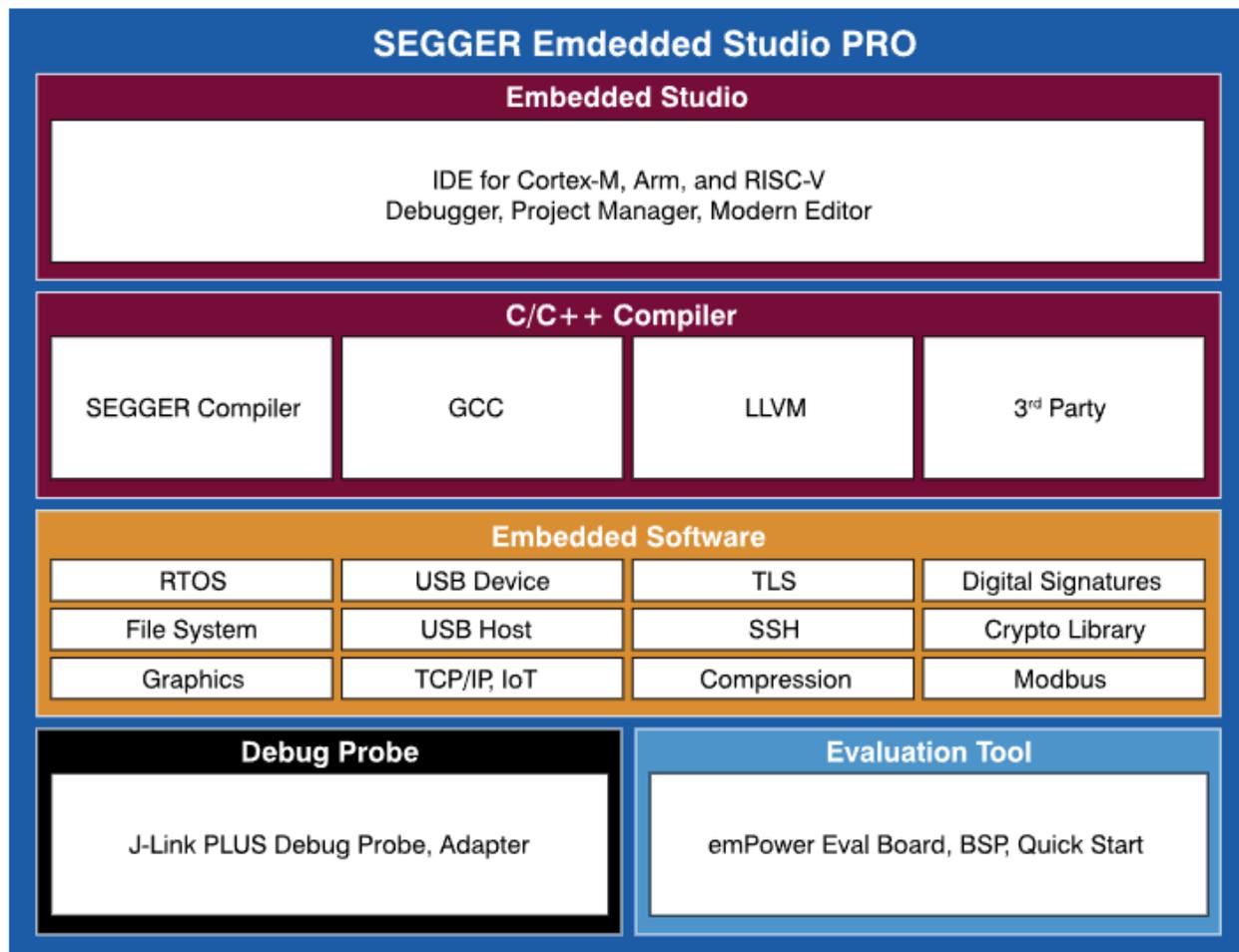
永久ライセンスで追加コスト不要

(任意年間保守でバージョンアップ対応)





RTOS, ファイルシステム, GUI, TCP/IP, 暗号セキュリティすべてを開発環境に統合
アプリケーションに合わせて選択実装可能



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

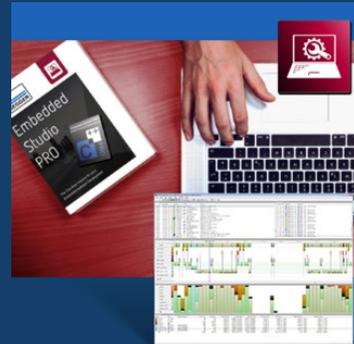
本社 : モーンハイム・アム・ライン (ドイツ)

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



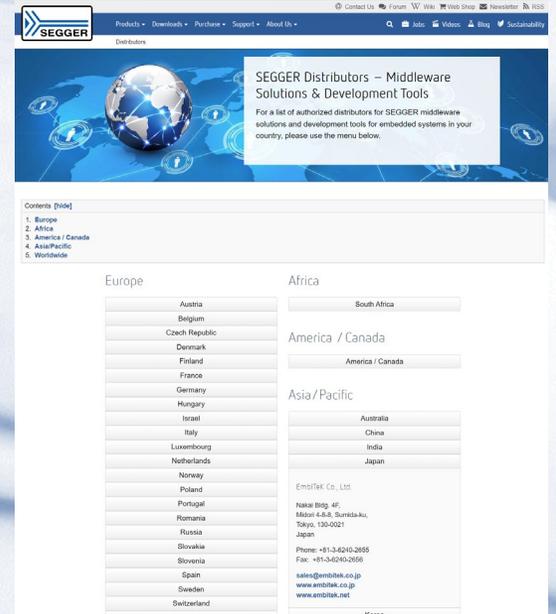
IDE



デバッグツール



書き込みツール



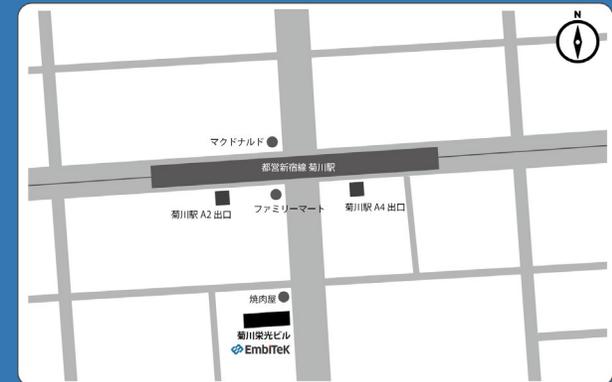
お客様の要件に合わせて、様々なシナリオで適合できる最適なソフトウェア開発環境ソフトウェアコンポーネントを提供します。

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



RTOS



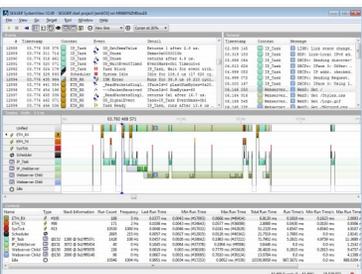
embOS

MPU 対応



機能安全認証
IEC61508 SIL3
IEC62304 class C

デバッガ
開発ツール



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 Sntp 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>