



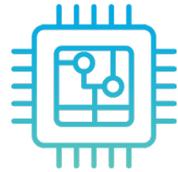
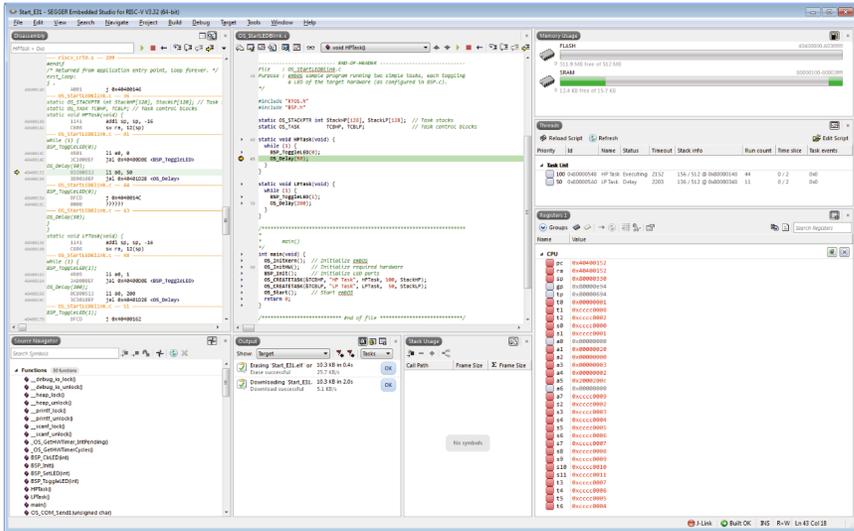
SEGGER Embedded Studio PRO

Cortex-M用ソフトウェア開発プラットフォーム

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

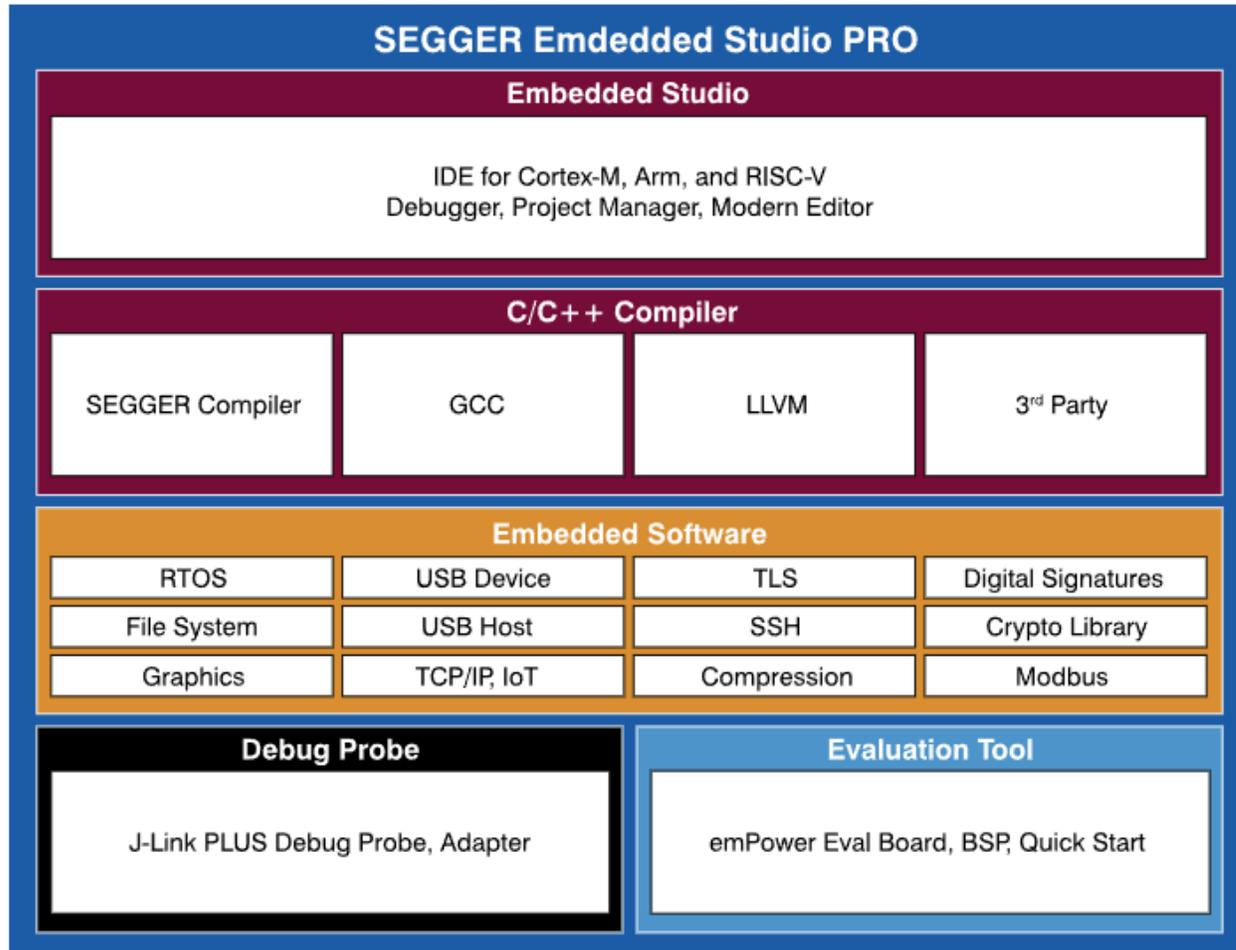
永久ライセンスで追加コスト不要

(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



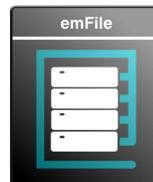
同梱するソフトウェアモジュール



Embedded Studio

IDE

- クロスプラットフォーム対応 (Win, MacOS, Linux)
- Clang/LLVM、GCC C/C++
SEGGER独自コンパイラ
- コンパクト標準ライブラリ
- J-Linkと統合化されたデバッグ
- CPUサポートパッケージ、
簡単にプロジェクト立ち上げ



emFile PRO

ファイル
システム

機器の外部とのデータファイルのやりとりに



基本対応機能

FATファイルシステム / NAND Flash Device Driver /
NOR Flash Device Driver / SD/SDHC/MMC Device
Driver / IDE/Compact Flash Device Driver / Long
File Name add-on / **ジャーナリングアドオン** / 暗号
化アドオン / **RAID1 / RAID5**



embOS

RTOS

機器のマルチタスクを実現
省電力対応
高性能・低フットプリント
使いやすいAPI



emWin PRO

GUI

Cortex-Mに最適なコンパクトGUI

豊富な市場実績

日本語・多言語対応

※PROパッケージに含まない
モジュールは別途オプション

フォントコンバータ別途オプション



※赤字：別途オプション

同梱するソフトウェアモジュール



emUSB-Host PRO

■ USB1.0/2.0対応 モジュール構造

USB
Host

基本対応機能・対応クラス

USB-Host Core / MSD / HID / Pinter /
CDC / FTDI UART / LAN / MTP / CCID /
MIDI / Audio / CP210x UART



emUSB-Device PRO

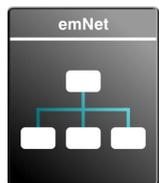
多数のクラスドライバ対応
IP-over-USB対応



USB
Device

基本対応機能・対応クラス

USB-Device Core / HID / MSD / CDC / Printer / MTP /
virtualMSD / Bulk/ CCID / **MSD-CDROM / CDC-ECM**
/ CDC-NCM / IP-over-USB(RNDIS / CDC-ECM) / MIDI
/ Audio / Video / DFU

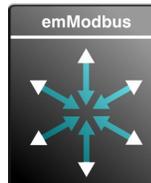


emNet PRO IPv4/IPv6 Dual Stack

基本対応機能・プロトコル

IPv4 / IPv6 / Wifi support / ACD / AutoIP / ARP /
CoAP client/server / DHCP client / DHCP server /
mDNS server / FTP client / FTP server / ICMP /
Loopback device / Multicast / MQTT client / TFTP /
NetBIOS Name Service / RAW Sockets / NTP client /
PPP/PPPoE / PTP OC client / Tail tagging / SMTP
client / **SNMP Agent** / SNTP client / TCP / UDP /
UPnP / VLAN / WebSocket / Web server

TCP/IP



emModbus Master / Slave

マスター・スレーブ対応
ASCII / RTU / TCP対応
産業機器通信で利用される
「Modbus」規格に基づく通信
を実装



※赤字：別途オプション

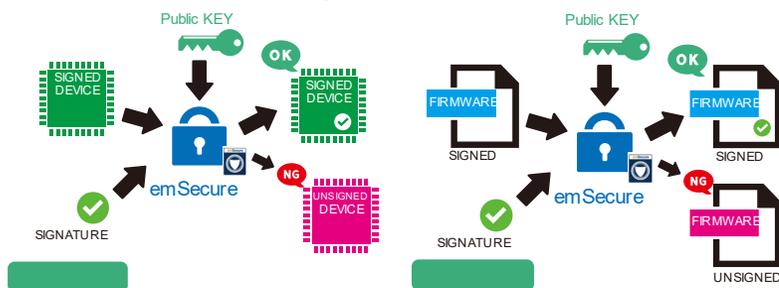
同梱するソフトウェアモジュール



emSecure RSA / ECDSA

不正改造や不正量産を防ぐ
デュアルシグニチャ認証

セキュア
認証



emCrypt PRO

多数のアルゴリズム対応
非GPL / 非オープンソース
改ざん・データ保護のための暗号化ライブラリ
各種暗号・ハッシュアルゴリズム
認証プロトコル
鍵生成アルゴリズムなどを
ユーザアプリからAPI利用

USB
Device



emSSL

通信経路におけるデータ流出を防ぐ
コンパクトなSSL
クライアント認証対応
非GPL / 非オープンソース

SSL



emSSH

SSHセキュアログイン認証

SSH

オプション：
SSH-SCPサーバ機能対応



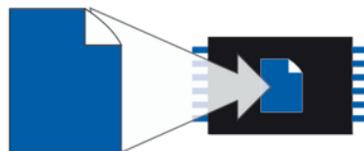
同梱するソフトウェアモジュール



emCompress PRO
emCompress Embed
emCompress ToGo

圧縮
解凍

複数CODECを付属
データサイズの削減でストレージ効率運用
データ通信の高速化
ファームウェア転送効率化



IoT Toolkit

JSON
Parser
HTTPc

JSONパーサー

JSON (JavaScript Object Notation) データ処理を簡単かつメモリ効率の良いParserライブラリ

HTTPクライアント

RESTインタフェースAPIリクエストを実行し、返されたデータを処理します。

テスト用ハードウェアプラットフォーム

- 評価ボード (Cortex-M4, 180MHz)
- J-Link PLUS
- J-Link用19-Pin変換アダプタ
- USBケーブル

ソフトウェアツールライセンスを同梱

- 専用デバッグツールOZONE
- フラッシュ書込み専用J-Flashツール
- SPIフラッシュライターJ-Flash SPI
- 無制限ブレイクポイント





各種ソフトウェアモジュールは10-30年の市場実績を持つソフトウェアライブラリ
 統合開発環境から必要なオブジェクトコードを利用します。
 ユーザハードウェア実装、アプリケーション組込を簡単に行う事ができます。

- アプリケーションレイヤ (OSレイヤ)
- ソフトウェアオプションモジュール
- ソフトウェアコアモジュール
- デバイスドライバ
- ボードサポートパッケージ (BSP)



通常同梱するembOS (オブジェクト) を利用する場合はそのまま利用可能。ユーザアプリケーションを迅速に開発



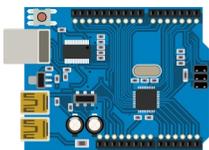
通常ライセンスではソースコードは提供しますが、ユーザ様でコード編集する必要はありません。
 (SEGGER社提供品をそのまま利用)



SEGGER Embedded Studio PROではオブジェクト提供



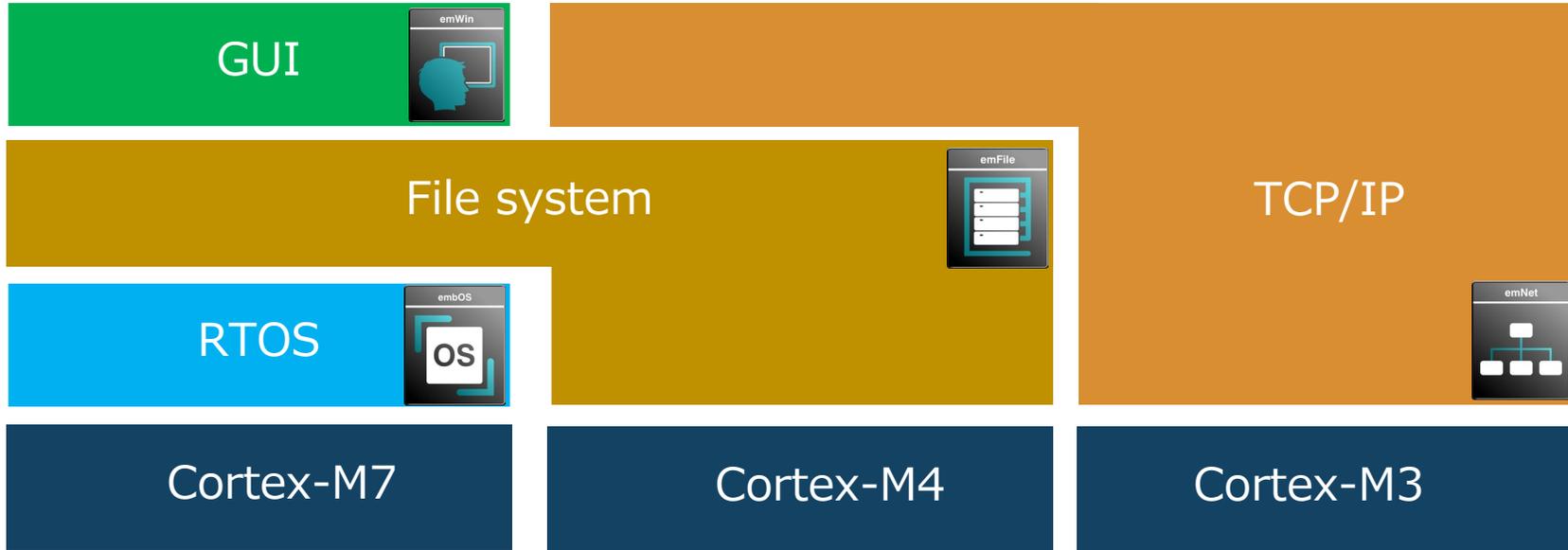
評価ボード用サンプル提供されているものはそのまま動作
 ユーザハードウェアに合わせて編集可能



Cortex-Mマイコン



アプリケーションに必要なソフトウェアのみ取捨選択、導入可能



購入したライセンスを製品ラインナップやデバイス・RTOSに依存することなく利用可能

一つの製品開発の予算ではなく基盤ソフト開発予算として計上できれば、導入障壁は低くなります。



ユーザ視点：
一度購入してしまえば、
試作開発や開発予算のない
プロジェクトでも気兼ねなく
使える！





RTOS/ミドルウェアオブジェクトを簡単に導入

SEGGER Embedded Studio V8.12 - Options

Project 'MyProject' Options

Debug Search Options Show Modified Options Only

Option	Value
embOS	
• Add embOS	Yes inherits
• embOS Library Configuration	Debug build with profiling (dp) modified
• Enable Erratum 837070 workaround	No inherits
emCompress	
• Add emCompress library	Yes inherits
• emCompress Library Configuration	Debug build (d) modified
emCrypt	
• Add emCrypt	Yes inherits
• emCrypt Library Configuration	Debug build (d) modified
emFile	
• Add emFile	Yes inherits
• emFile Library Configuration	Debug build (d) modified
• emFile Locking Level	Use single lock inherits
emModbus	
• Add emModbus library	Yes inherits
• emModbus Library Configuration	Debug build (d) modified
emNet	
• Add emNet	Yes inherits
• emNet Library Configuration	Debug build (d) modified
emSecure	
• Add emSecure	Yes inherits
• emSecure Library Configuration	Debug build (d) modified

Add embOS

Add embOS to your application.

- `#define USE_EMBOS=1`
- `#include $(PackagesDir)/ESPRO/OS/Inc/`

OK Cancel

ユーザアプリケーションに必要なRTOS/
ミドルウェアのオブジェクトライブラリを選択、

利用したいもののみ呼び出して利用する事が
できます。



Embedded Studio

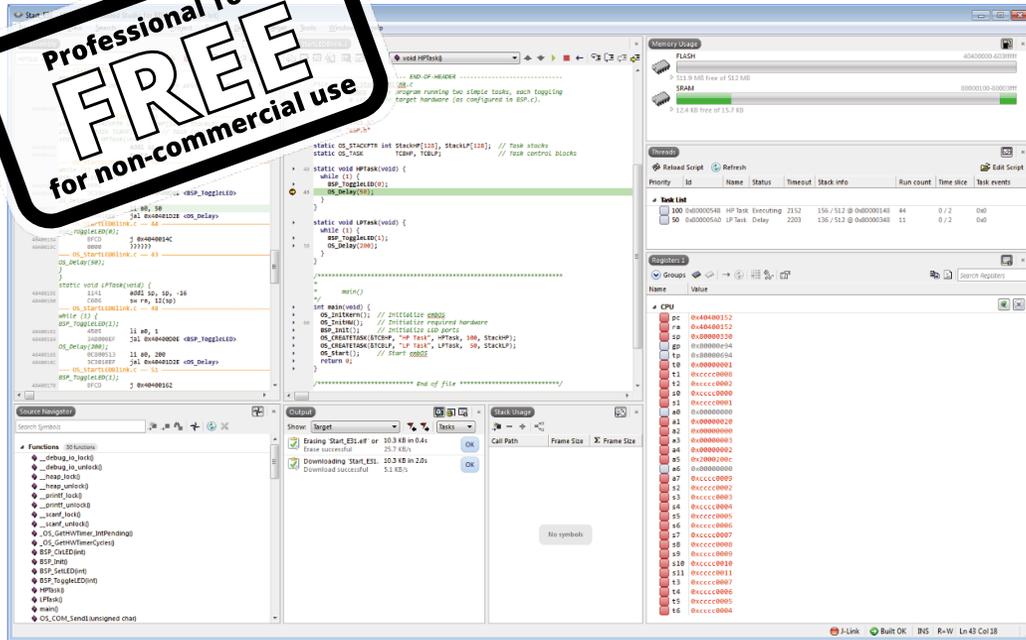
統合開発環境 コンパイラ・デバッガ

J-Linkシリーズとネイティブに連携する統合開発環境

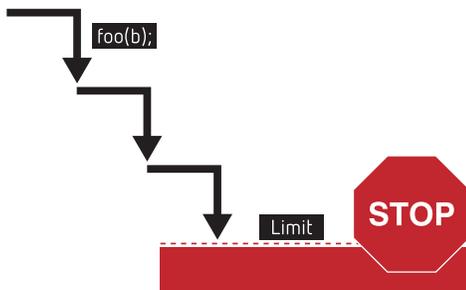


C/C++プログラミング用プロフェッショナルIDE

Professional Tools
FREE
for non-commercial use



- **クロスプラットフォーム対応** (Windows, MacOS, Linux)
- Clang/LLVM、GCC C/C++コンパイラ同梱
- 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
- **非商用利用 (評価用途) 無償フルパッケージ**



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。



C/C++プログラミング用プロフェッショナルIDE



コードエディタ



プロジェクト
マネージャ



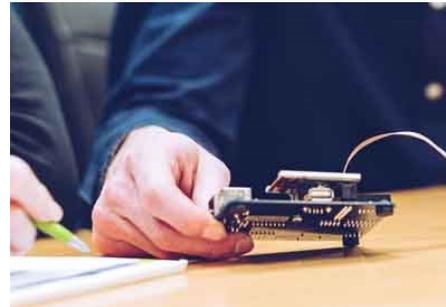
コンパイラ



コード解析



プロジェクト
インポータ



RTT



デバッガ



アプリケーション開発効率化するコードエディタを同梱



コード補完機能

```

char ac[256];
char acFileName[32];
const char * sVolumeName = "";

FS_X_Log("Start\n");
// Initialize file system
80 FS_Init();
// Check if low-level format is required
84 FS_FormatLL;
// FS_FormatLLRequired
// check if volume needs to be high level formatted.
if (FS_IsHLFormatted(sVolumeName) == 0) {
    FS_X_Log("High-level format\n");
    FS_Format(sVolumeName, NULL);
90 }
SEgger_sprintf(ac, sizeof(ac), "Running sample on \"%s\"\n", sVolumeName);
FS_X_Log(ac);
v = FS_GetVolumeFreeSpaceKB(sVolumeName);
if (v < 0x8000) {
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu KBytes\n", v);
} else {
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu MBytes\n", v);
100 }
FS_X_Log(ac);
SEgger_sprintf(acFileName, sizeof(acFileName), "%s\\file.txt", sVolumeName);
SEgger_sprintf(ac, sizeof(ac), " Write test data to file %s\n", acFileName);
pFile = FS_FOpen(acFileName, "w");
if (pFile) {
    FS_Write(pFile, "Test", 4);
    FS_FClose(pFile);
} else {
110 SEgger_sprintf(ac, sizeof(ac), "Could not open file: %s to write.\n", acFileName);
    FS_X_Log(ac);
}
v = FS_GetVolumeFreeSpaceKB(sVolumeName);
if (v < 0x8000) {
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu KBytes\n", v);
} else {
    v >>= 10;
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu MBytes\n", v);
}
    
```

お客様のコードに合わせて推測補完を提案します。

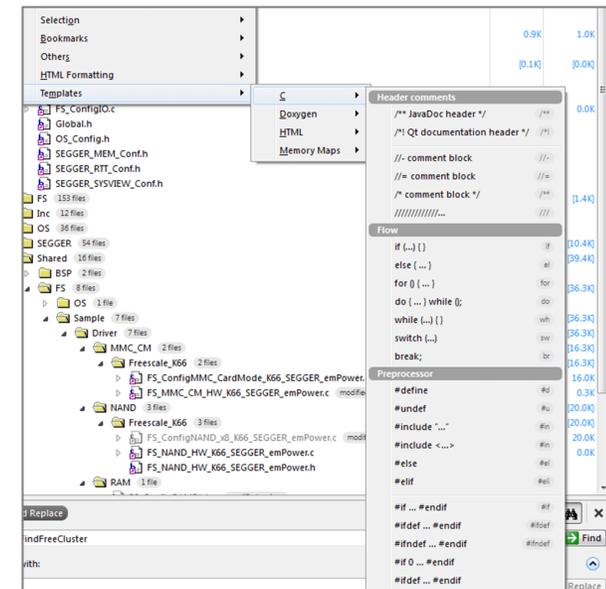
簡易ナビゲーション

```

SEgger_sprintf(ac, sizeof(ac), "Running sample on \"%s\"\n", sVolumeName);
FS_X_Log(ac);
v = FS_GetVolumeFreeSpaceKB(sVolumeName);
if (v < 0x8000) {
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu KBytes\n", v);
} else {
    v >>= 10;
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu MBytes\n", v);
}
FS_X_Log(ac);
SEgger_sprintf(acFileName, sizeof(acFileName), "%s\\file.txt", sVolumeName);
SEgger_sprintf(ac, sizeof(ac), " Write test data to file %s\n", acFileName);
FS_X_Log(ac);
pFile = FS_FOpen(acFileName, "w");
if (pFile) {
    FS_Write(pFile, "Test", 4);
    FS_FClose(pFile);
} else {
    SEgger_sprintf(ac, sizeof(ac), "Could not open file: %s to write.\n",
    FS_X_Log(ac);
}
v = FS_GetVolumeFreeSpaceKB(sVolumeName);
if (v < 0x8000) {
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu KBytes\n", v);
} else {
    v >>= 10;
    SEgger_sprintf(ac, sizeof(ac), " Free space: %lu MBytes\n", v);
}
FS_X_Lbg(ac);
FS_Unmount(sVolumeName);
FS_X_Log("Finished\n");
while (1) {
    ;
}
//***** End of file *****
    
```

ブックマーク機能により、素早くコード間移動が可能です。

コードテンプレート



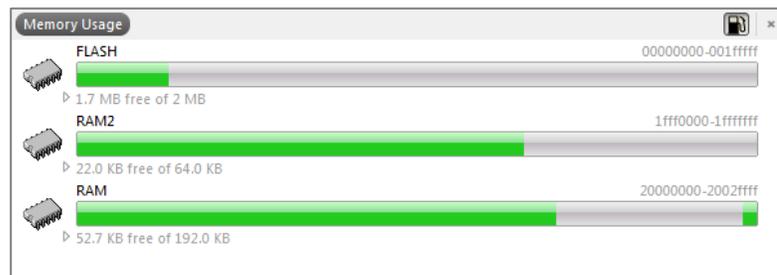
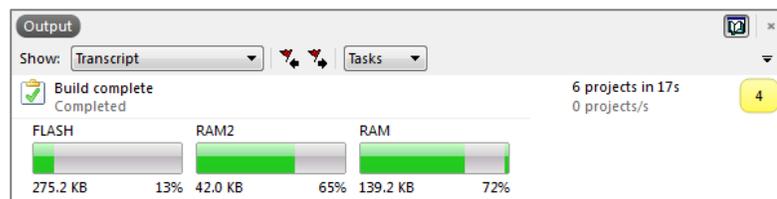
テンプレート機能により、代表的なコードを自動入力できます。



コードを静的解析し、アプリケーション開発の情報を提供

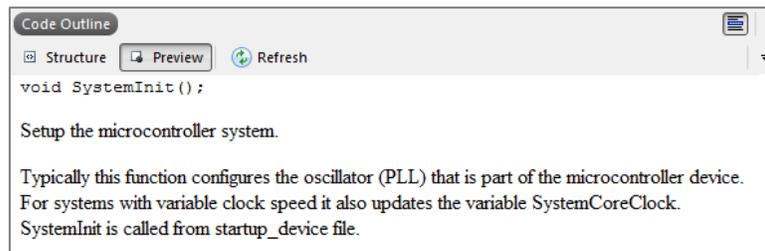
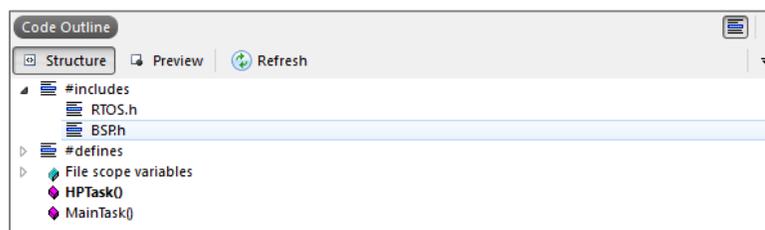
コード解析

メモリ使用量表示



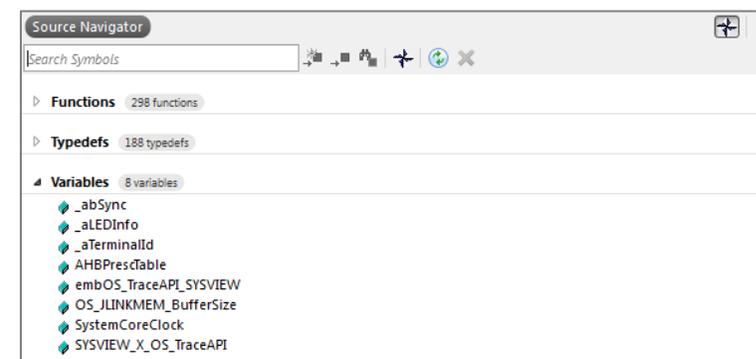
お客様のアプリケーションコードを解析することにより、ROM/RAM使用量を可視化して、表示することができます。

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することができます。

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することができます。



スタック使用量表示やコードの問題点、外部ツール連携による解析

コード解析



シンボルブラウザ

Name	Range	Size
(No section)		
.bss	20000008-200010e3	4,184
.data	20000000-2000008b	140
AHBPrescTable	2000007c-2000008b	16
OS_Global	20000030-20000077	72
SystemCoreClock	20000078-2000007b	4
_aLEDInfo	20000000-2000002f	48
__data_end__	2000000c	
__data_start__	20000000	
.data_run		
.fast_run		
.heap	200010e4-200014e3	1,024
.init	0800017c-080003c3	584
.rodata	08004e48-08004f68	289
.stack	2001fd00-2001ffff	768
.stack_process		
.tdata_run		
.text	080003c4-08004e47	19,076
BSP_ClrLED	0800004c-080004f7	52
BSP_Init	080003c4-0800048f	204
BSP_SetLED	08000490-080004c3	52

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することができます。

スタック使用量解析

Call Path	Frame Size	Σ Frame Size
IP_UDP_OnRxiP4	48	232
IPv4_OnRx	40	184
IPv4_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します。

静的コード解析

Task	Files	Warnings
Analyzing 'Setup' in configuration 'Debug'	7 files in 0.5s	2 Warnings
Analyzing BSP.c	13 files/s	2 Warnings
Potential memory leak		
Value stored to 'Index' is never read		
Analyzing BSP_IPc.c		
Analyzing BSP_USB.c		
Analyzing JLINKMEM_Process.c		
Analyzing OS_Error.c		
Analyzing OS_ThreadSafe.c		
Analyzing RTOSInit_K66F_CMSIS.c		
Finished Analyzing	7 files in 0.5s	
Completed	13 files/s	

静的コード解析で、問題の可能性のあるコードを検索し、表示します。

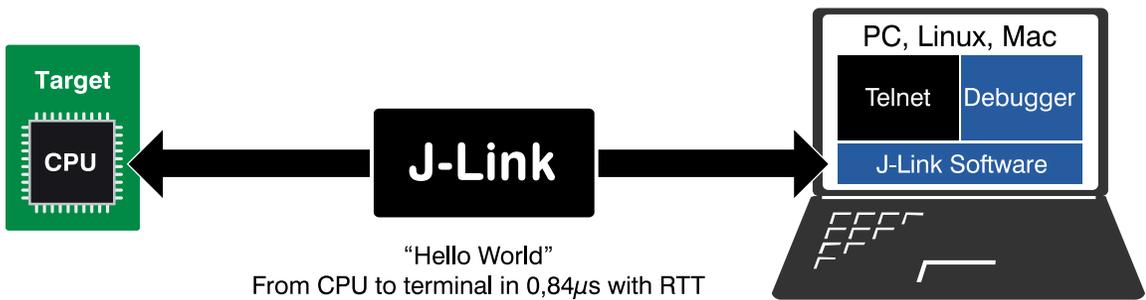
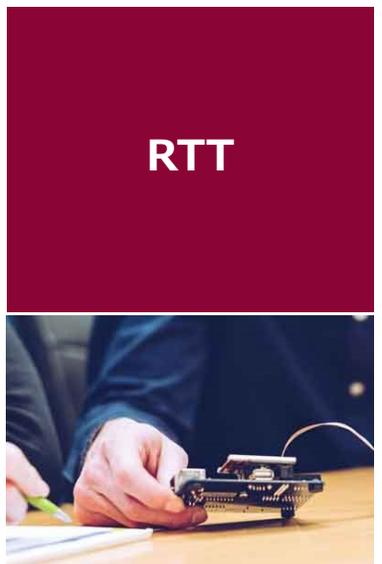
Task	Files	Warnings
Linting 'Src' in configuration 'Debug'	3 files in 0.0s	2 Warnings
Linting JLINK_MONITOR.c	45 files/s	
Linting Main.c		1 Warning
A zero has been given as right argument to operator '<<' [-e835]		
Constant value Boolean [-e506]		
local macro 'MAIN_C' (line 14, file Main.c) not referenced [-e750]		
Linting system.c		1 Warning
Finished Linting	3 files in 0.0s	
Completed	45 files/s	

外部ツール連携

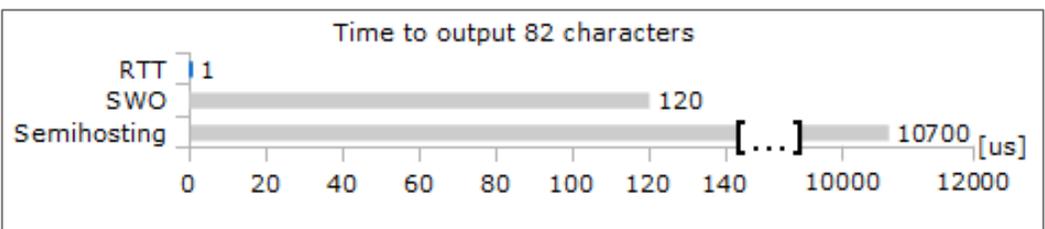
外部の構文解析ツールなどを組み込んで利用する事が可能です。PC-lintなどを組み込み、MISRA-Cチェックを実施することもできます。



高速・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



タイムラグのないデータ取得・送信
複数のチャンネルを双方向で利用可能



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

Printf()よりも高速データ転送を実現

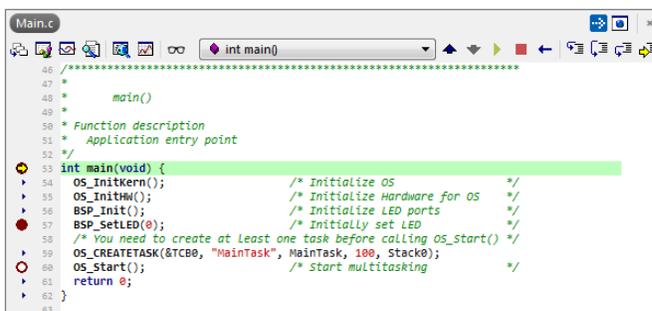
デバッガが接続されていない状態でもアプリケーションに影響しない。
リリースコードや稼働試験用コードに含まれていても問題なし。
(非ブロッキングモード利用可能)



J-Link / J-Trace PROと統合し、デバッグ環境を提供

デバッグ

ソースコードエディタ

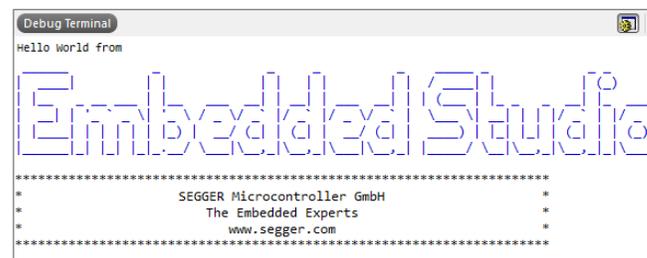


```

46
47
48 *
49 *   main()
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", 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。また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です。

ターミナル



```

Debug Terminal
Hello world from

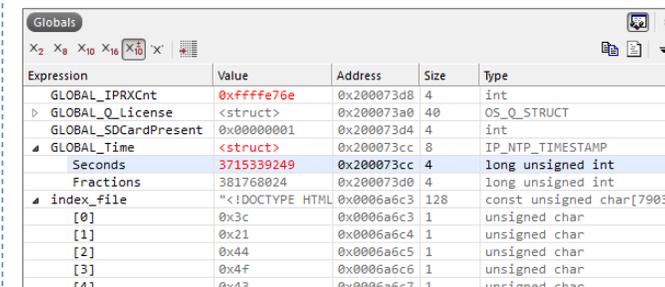
Embedded Studio

*****
*   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_IPRXCnt	0xfffffe76e	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することができます。



J-Link / J-Trace PROと統合し、デバッグ環境を提供

デバッグ

スレッド

Pri	Id	Name	Status	Timeout	Stack info	Run
Task List						
0x20008904	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です。

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	0x80000000

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。Embedded Studio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。

コールスタック

Function	Call Address	Stack Pointer	Frame Size	Stack Used
void FS_FindVolume(char * pFullName=mmc0, ...)	0x0002DF22	0x200072C8	56 bytes	216 bytes
int FS_GetVolumeStatus(char * sVolume=mmc0;)	0x0002E122	0x20007300	32 bytes	160 bytes
int_FSCheckSD()	0x0000BAE6	0x20007320	64 bytes	128 bytes
long unsigned int_FSCheckPeriodic(long unsig...)	0x0000BD2	0x20007360	24 bytes	64 bytes
void_MainTask()	0x0000B72E	0x20007378	40 bytes	40 bytes
OS_StartTask()	0x0000E8E4	0x200073A0	0 bytes	0 bytes

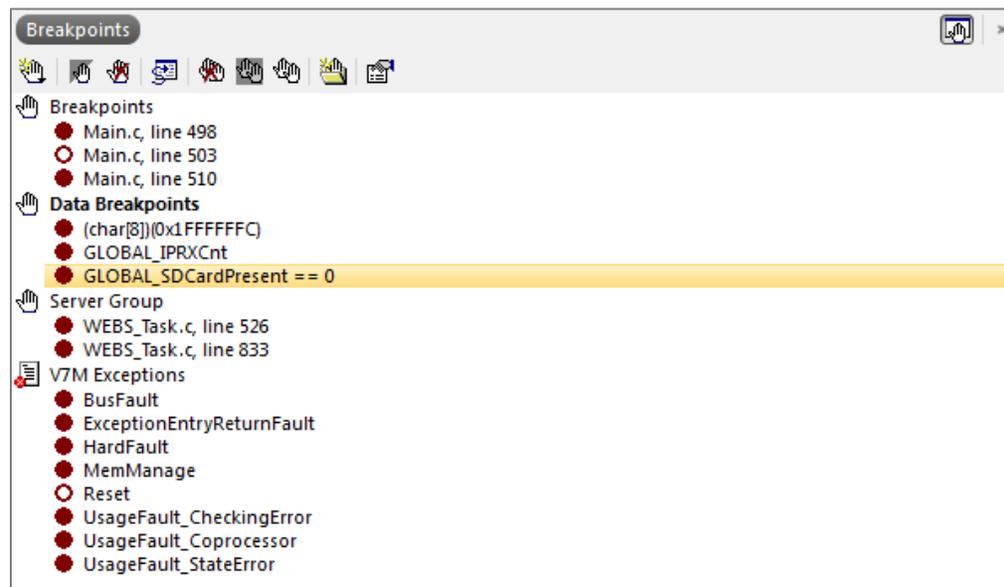
コールスタックウィンドウでは、アプリケーションが停止時の機能とその呼び出し元が表示されます。各コールをダブルクリックして、コールの正確な位置情報を取得することが可能です。



J-Link / J-Trace PROと統合し、デバッグ環境を提供

デバッグ

ブレイクポイント



Embedded Studioデバッガには、命令、関数、及びソース行のコードブレイクポイントと、メモリ内変数へアクセス時のデータブレイクポイントがあります。

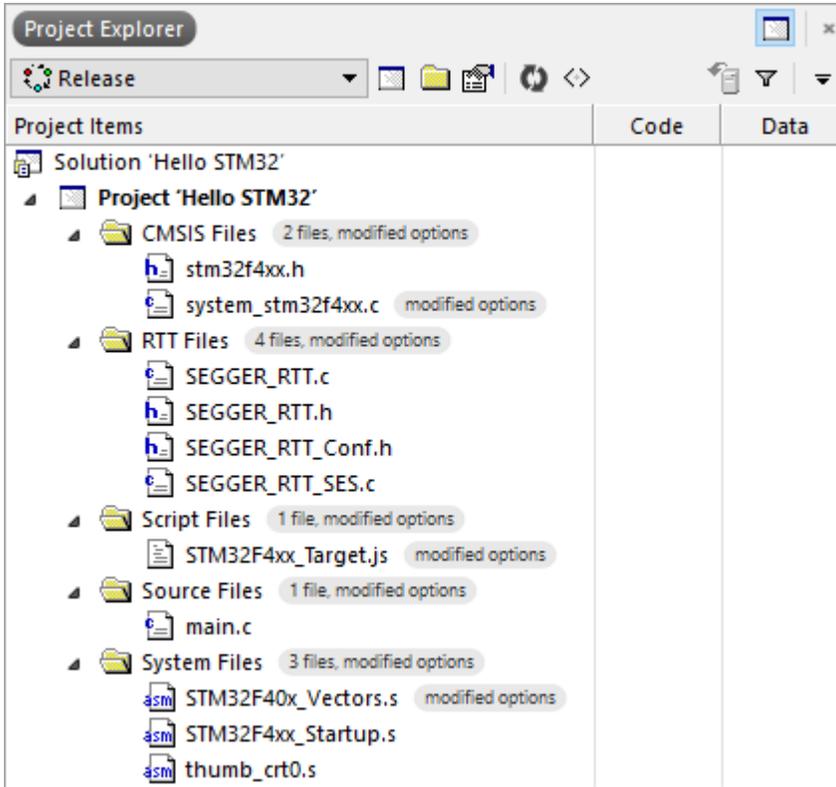
単純なC言語表現でブレイクポイントを設定できます。
例えば `(x==4)` という記述では、「x」がアクセスされ、その値が「4」の場合にブレイクポイントになります。

演算子として「<, <=, >, >=, ==, !=」が利用できます。
また「&」を利用して複合的な要素を持たせることも可能です。
さらに配列キャストを使用して、任意のメモリ範囲をブレイクポイントに指定することも可能です。

例として
`(char[256])(0x1000)` という記述では、「0x1000-0x10FF」にアクセスされた際に、ブレイクポイントとなります。また「!」を利用して、逆にメモリ範囲から外れた場合にブレイクをかけるという指定も可能です。



高い生産性を実現するプロジェクトマネージャ

プロジェクト
マネージャ

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。

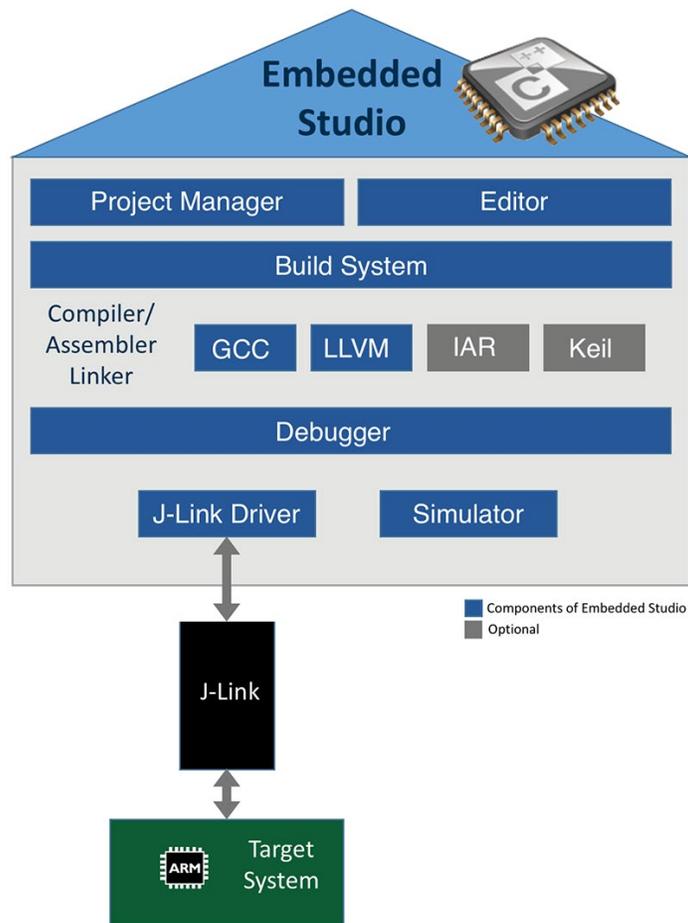
プロジェクトエクスプローラは、プロジェクト、ファイルを整理し、それらのプロパティを表示、必要なコマンドに素早くアクセスできます。

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。



高い生産性を実現するプロジェクトマネージャ

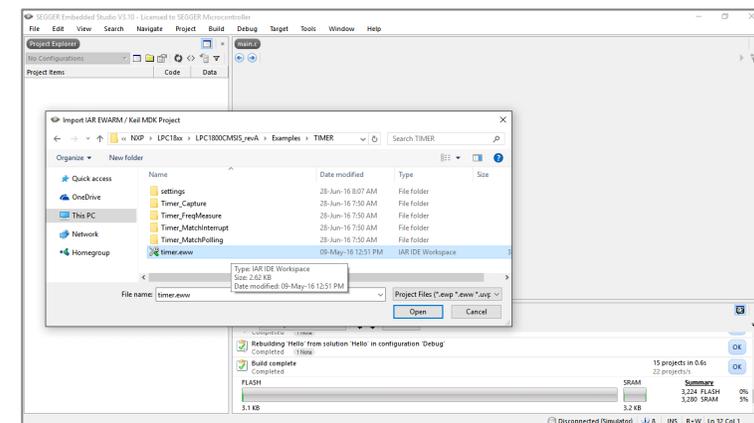
プロジェクトインポータ



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。





マイグレーション対応

プロジェクト インポータ



マイグレーション

GCCベースのプロジェクト・GCC互換のソースコードの場合、比較的スムーズに切り替えて利用することができます。

一般的なマイグレーション対応

■ Linker Script

Embedded Studioプロジェクトインポータでは、Linker Scriptのマイグレーション変更は行いません。選択したデバイスの基本的なメモリマップで配置します。ユーザ様でメモリマップを変更する場合、それに応じたメモリマップ配置、セクション配置ファイルを作成する必要があります。

■ オブジェクト提供ソフトウェア

オブジェクトライブラリは、一般的に一つのコンパイラ・IDEへの対応を前提に作成されている事が多いため、Embedded Studioで再構成できない場合があります。リンク可否などは事前にご確認ください。

■ アセンブラファイル・インラインアセンブラ

アセンブラはコード、シンタックスはツールチェーンにより、記述や制御コマンドも異なります。シンボル、ラベル、関数定義などの構文は、GNUアセンブラ用に変更する必要があります。

■ プリプロセッサ定義

ツールチェーンは、様々な定義を利用して使用されるコンパイラを識別し、コア、デバイス、エンディアン、その他の設定構成に基づいて、コンパイル条件を定義しています。そのためEmbedded Studioの定義方法と異なる定義記述がなされている場合、コードエラーが発生します。エラーは適時確認排除する必要があります。



SEGGER コンパイラ/GCC, LLVMコンパイラを同梱

コンパイラ



Output		
 Building 'IP' in configuration 'Debug' Completed	65 targets in 2.1s 30 targets/s	OK
 Building 'FS' in configuration 'Debug' Completed	99 targets in 3.4s 29 targets/s	OK
 Building 'USB' in configuration 'Debug' Completed	18 targets in 0.9s 19 targets/s	OK
 Building 'CRYPTO' in configuration 'Debug' Completed	117 targets in 3.5s 33 targets/s	OK
 Building 'SSL' in configuration 'Debug' Completed	16 targets in 0.6s 25 targets/s	OK
 Building 'emPower' in configuration 'Debug' Completed	85 targets in 3.1s 27 targets/s	OK
 Build complete Completed	6 projects in 13s	OK

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

SEGGER社で最適化されたCライブラリ

ほとんどのGCCベースのアプリケーションは、Newlibまたはより小さなNewlib nanoを標準的なライブラリとして、採用しています。しかしながら、Newlib nanoでもprintf()などの関数では、多くのスタックとヒープが必要になる場合があります。

Embedded Studioには、独自の標準Cライブラリが付属しています。このライブラリは、ゼロから開発し、組込アプリケーション向けに調整・最適化されています。



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>