

最新の IoT と組み込みテクノロジーを利用したスマートな開発

インテル® System Studio で優れたアイデアを優れた製品に

Noah Clemons インテル コーポレーション 並列プログラミング製品 テクニカル・コンサルティング・エンジニア

優れたアイデアを優れた製品にするには多くの作業を行う必要がありますが、作業に適したツールを利用すると、そのプロセスは非常に簡単になります。インテルは、組み込みシステム開発のニーズをすべて満たすツールスイートとして、インテル® マイクロコントローラー、IoT (Internet of Things) デバイス、組み込みプラットフォームのコーディング、解析、デバッグ用の **ソフトウェア開発ツール** を統合した、インテル® System Studio およびインテル® System Studio for Microcontrollers を提供しています。

この記事では、次の項目について説明します。

- インテル® System Studio とインテル® System Studio for Microcontroller のコンポーネントの概要
- すべてのプラットフォームでコンポーネントがどのように動作するか
- 新しいマイクロコントローラー・プラットフォームを提供するためにコンポーネントを変更した場所
- インテル® System Studio がどのように IoT のニーズと組み込み開発に対応しているか

インテル® System Studio の対象ユーザー

インテル® System Studio の対象ユーザーは次のとおりです。

- プラットフォーム用の正しいツールを必要としているデバイスメーカー
- 既存のプラットフォームに依存し、完全なシステム・ソフトウェア・スタックを確立する必要があるシステム・インテグレーター
- 専用アプリケーションをビルドおよび最適化する必要がある組み込みソフトウェア開発者

優れた製品のビルド

インテル® System Studio は、インテル® ハードウェアで動作する製品を開発するために必要なすべてのツールを提供することを目的としており、インテル® Quark™ SoC X1000 やインテル® Atom™ プロセッサ・ベースの IoT ゲートウェイから、インテル® Core™ プロセッサおよびインテル® Xeon® プロセッサ・ベースのサーバーまで、さまざまな組み込みプラットフォームをサポートしています。

インテル® System Studio for Microcontrollers は、インテル® Quark™ マイクロコントローラー D1000/D2000 およびインテル® Quark™ SE マイクロコントローラー向けにカスタマイズされた開発環境を提供するため特別に用意されたツールスイートです。

BLOG HIGHLIGHTS

デバイスの選択

ALEX KATRANOV >

この記事は、インテル® スレッディング・ビルディング・ブロック (インテル® TBB) 4.4 Update 2 以降で利用可能な新しいノード `opencl_node` について説明するシリーズのパート 3 です。このノードは、インテル® TBB のフローグラフで OpenCL* デバイスの利用と連携を可能にします。このシリーズのパート 1 は[こちら](#)からご覧いただけます。

パート 2 では、基本インターフェイスを説明しました。パート 3 では、カーネルの実行に使用するデバイスの選択について述べます。

この記事の続きは[こちら](#)でご覧になれます。 >

表 1 に、これらのスイートで提供される機能の一部を示します。

	インテル® System Studio	インテル® System Studio for Microcontrollers
プラットフォーム	インテル® Xeon® プロセッサ、インテル® Core™ プロセッサ、インテル® Atom™ プロセッサ、インテル® Quark™ SoC U シリーズ	インテル® Quark™ マイクロコントローラー D シリーズ
ソフトウェア開発環境	Eclipse* ベースの統合開発環境 (IDE)、コマンドライン、インテル® Graphics Performance Analyzers	Eclipse* ベースの IDE、コマンドライン
ホストシステム	Linux*、Windows*、OS X* ¹	Linux*、Windows*
コンパイラー	インテル® C++ コンパイラー	インテル® C++ コンパイラーまたは GNU* C コンパイラー ²
ターゲット・プラットフォーム OS	Linux*、Android*、Windows*、FreeBSD*、VxWorks*	ベアメタル、リアルタイム OS
ターゲット・プラットフォーム・ソフトウェア	サンプル、デバッガー、プロファイラー対応ドライバー	ボード対応パッケージ、インテル® Quark™ マイクロコントローラー・ソフトウェア・インターフェイス、サンプル
ライブラリー	インテル® MKL、インテル® IPP、インテル® TBB (画像、信号処理、データ処理、マルチスレッド)	C ランタイム、浮動小数点エミュレーション、DSP ライブラリー
アナライザー	インテル® VTune™ Amplifier for Systems、インテル® Energy Profiler、インテル® Inspector for Systems (メモリー・アナライザー)	電力アナライザー ³
デバッガー	アプリケーションおよび OS、WinDbg カーネルデバッガー、インテルの GDB 拡張、インテル® System Debugger、JTAG、JTAG over USB、UEFI Agent	アプリケーションおよび OS、Embedded System Registers View、MCU フラッシュ用インテルの GDB 拡張、OpenOCD* ベースの JTAG

1. インテル® System Studio は OS X* 機能の一部をサポートしています。
 2. インテル® Quark™ マイクロコントローラー D1000 は LLVM ベースのインテル® コンパイラーでサポートされています。インテル® Quark™ マイクロコントローラー D2000 およびインテル® Quark™ SE マイクロコントローラーは GCC でサポートされています。
 3. マイクロコントローラー向け電力アナライザーは近日提供予定です。

表 1. インテル® System Studio とインテル® System Studio for Microcontrollers の機能

インテル® System Studio

インテル® System Studio では、コマンドラインまたは GUI ベースのツール (Eclipse* または Microsoft* Visual Studio* 統合環境を含む) を選択できます。これらのツールは、ターゲットとして Windows*、Linux*、VxWorks*、Wind River Linux*、FreeBSD*、Android* をサポートしています。また、インテル® Quark™ プロセッサ、インテル® Edison プラットフォーム、インテル® Atom™ x3 プロセッサ (開発コード名 SoFIA)、インテル® Atom™ x5/x7 プロセッサ (開発コード名 Cherry Trail)、第 6 世代インテル® Core™ プロセッサ (開発コード名 Skylake) を含む、最新バージョンのインテル® プロセッサをサポートしています。

インテル® System Studio の重要なコンポーネントは、システムを解析してアーキテクチャーに対応したコードを生成する、最適化コンパイラー (インテル® C/C++ コンパイラー) と C++11 および C++14 (-std=c++14) 機能をサポートするライブラリーです。3 つのメイン・ライブラリーは、インテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP)、インテル® マス・カーネル・ライブラリー (インテル® MKL)、インテル® スレッディング・ビルディング・ブロック (インテル® TBB) です。

インテル® IPP は、ソフトウェア関数の広範なライブラリーにより、画像、信号、ストリング処理、データ圧縮、暗号化、コンピューター・ビジョン向けのパフォーマンス・ビルディング・ブロックを提供します。これらのライブラリーは、インテル® System Studio およびインテル® System Studio for Microcontrollers の両方でサポートされており、インテル® Quark™ プロセッサ、インテル® Atom™ プロセッサ、インテル® Core™ プロセッサ向けに追加の最適化を行います。

コンパイラーの最適化に関する詳細は、最適化に関する注意事項を参照してください。

これらの関数はロイヤルティ・フリーで、インテル® ストリーミング SIMD 拡張 (インテル® SSE) 命令セットおよびインテル® アドバンスド・ベクトル・エクステンション (インテル® AVX、インテル® AVX2、インテル® AVX-512) 命令セットを使用して高度に最適化されているため、最適化コンパイラ単独で生成した場合よりも高速なアプリケーションを生成できます。

パフォーマンスと電力効率を最適化	インテルにより開発され、将来にわたって開発期間を短縮	広範なクロスプラットフォームと OS 機能
高度にチューニングされたルーチン	現在および以前のプロセッサ向けに最適化	高度に最適化された数千の信号、データ、メディア関数
インテル® SSE、インテル® SSSE3、インテル® SSE4、インテル® AVX、インテル® AVX2、インテル® AVX-512 命令セットを使用して高度に最適化	開発、デバッグ、保守期間を短縮	広範なドメインをサポート
最適化コンパイラ単独で生成したよりも高速	一度記述したコードで将来の最適化にも対応	インテル® Quark™ プロセッサ、インテル® Core™ プロセッサ、インテル® Xeon® プロセッサ、インテル® Xeon Phi™ コプロセッサ・プラットフォームをサポート

表 2. インテル® System Studio の利点

インテル® MKL は、科学、工学、金融アプリケーションの演算処理を高速化します。バージョン 11.2 では、クラスター用並列直接法スパースソルバー、BLAS および LAPACK の verbose モード、小行列における S/C/Z/DGEMM の向上、SVD および固有値ソルバーのパフォーマンスの向上、このライブラリーをさらに拡張する機能と最適化が追加されました。

インテル® TBB は、タスクの並列化に広く使用されている C++ テンプレート・ライブラリーです。並列アルゴリズムとデータ構造、スレッドと同期プリミティブ、スケーラブルなメモリー割り当てとタスク・スケジューリング用のテンプレートを提供します。バージョン 4.3 では、メモリー・アロケータの改良 (改良された `tbbmalloc` によりマルチスレッド・アプリケーションのパフォーマンスとスケーラビリティを向上)、インテル® トランザクショナル・シンクロナイゼーション・エクステンション (インテル® TSX) のサポートの強化 (`tbb::speculative_spin_rw_mutex` により読み取り/書き込みロックを使用するアプリケーションでインテル® TSX をさらに活用)、C++ 11 標準規格との互換性の向上、タスク領域の改良 (新しいクラス `tbb::task_arena` によりワークロードの分離と並行性レベルをより細かく制御)、最新のインテル® アーキテクチャーのサポートが行われました。(ハードウェアのサポートについては、インテル® TBB [リリースノート](#)を参照してください。)

インテル® VTune™ Amplifier for Systems は、パフォーマンス・プロファイリング/電力解析ツールです。IoT ゲートウェイやモデムベースのプラットフォームでの電力解析の実行など、さまざまな解析タイプで必要なチューニング・データを迅速かつ容易に入手することができます。

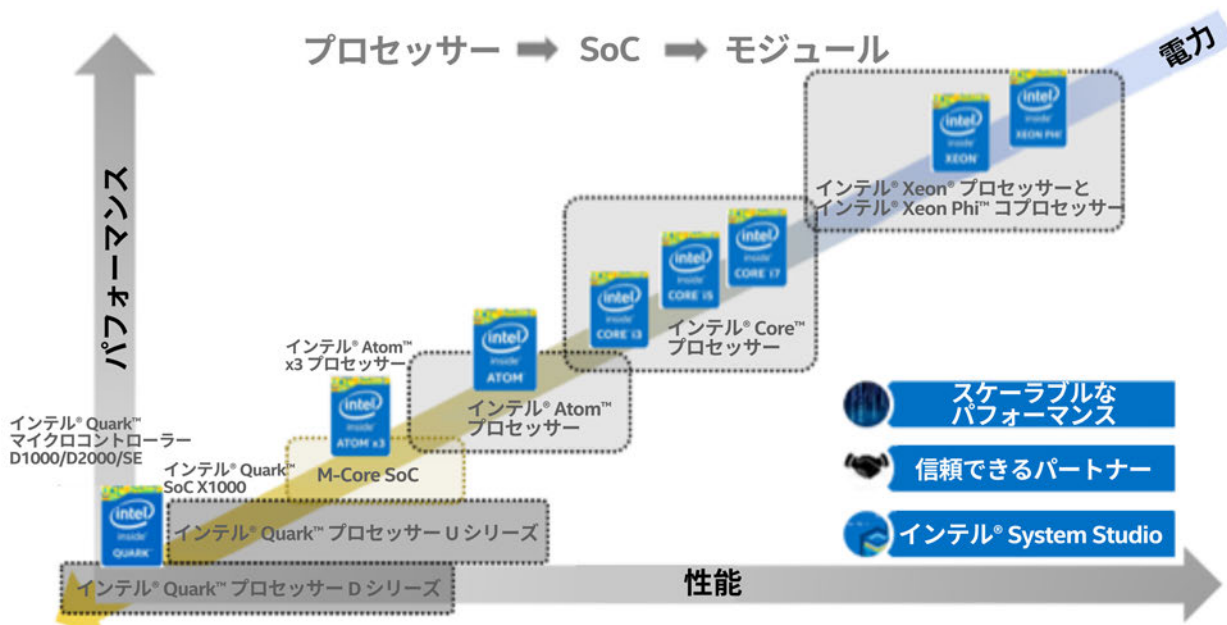
インテル® System Debugger は、USB 接続による JTAG デバッグをサポートし、デバッグのニーズを満たす多くのツールを提供します。最初のツールは Microsoft* WinDbg カーネルデバッガーの JTAG デバッグおよび命令トレースです (インテル® System Studio for Windows* ターゲットバージョン)。ボード起動中の発見が困難な Windows* ドライバーの問題を特定するのを支援します。また、複雑なランタイム問題の切り分けに役立つインテル® Processor Trace (インテル® PT) を WinDbg カーネルデバッガーでサポートしています。

コンパイラの最適化に関する詳細は、[最適化に関する注意事項](#)を参照してください。

インテル® Trace Hub は、システム全体のハードウェアおよびソフトウェア・イベントをキャプチャーするシステムトレースを実行します。(第 6 世代インテル® Core™ プロセッサをサポートしています。) これにより、ハードウェアとソフトウェア間の複雑な相互作用を素早く理解できます。また、タイムスタンプに関連したトレース情報を取得できます。

Closed Chassis Debug は、低コストの USB 接続による JTAG ベースのデバッグおよびトレースです。デバッグポートを利用することなく、製品要件に合ったフォームファクターで開発できます。デバッグポートの代わりに、USB ポートを利用して、デバッグに必要なすべての処理を行います。

図 1 は、ツールの IoT プロジェクトにおけるエンドツーエンドのスケラビリティを示しています。



1 IoT エンドツーエンドのスケラビリティ

インテル® System Studio for Microcontrollers

インテル® System Studio for Microcontrollers は、インテル® Quark™ マイクロコントローラー D1000/D2000 およびインテル® Quark™ SE マイクロコントローラーをサポートする新しいツールスイートです。このスイートは、Eclipse* ベースの統合開発環境を提供します。開発者は、make ユーティリティを使用して、コマンドラインからインテル® System Studio for Microcontrollers のツールを実行することもできます。

インテル® System Studio for Microcontrollers は、ベアメタルシステムや選択したリアルタイム・オペレーティング・システム (インテル® Quark™ マイクロコントローラー D2000 およびインテル® Quark™ SE マイクロコントローラー) で実行するコードの作成をサポートしています。また、ブートストラップ・コードを記述することなく、I/O 機能を単純化できるボード・サポート・パッケージ (BSP) が含まれています。

インテル® System Studio for Microcontrollers には、インテル® Quark™ マイクロコントローラー D2000 およびインテル® Quark™ SE マイクロコントローラー向け BSP に加えて、ほとんどの I/O インターフェイス (GPIO、アナログ入力、I2C、SPI、UART など) の実装とカスタマー・リファレンス・ボード (Bosch* BMC150 加速度計など) の周辺機器用のデバイスドライバーを含む、インテル® Quark™ マイクロコントローラー・ソフトウェア・インターフェイス (QMSI) パッケージが含まれています。

インテル® System Studio for Microcontrollers には、浮動小数点エミュレーションと DSP ライブラリーが含まれており、DSP ライブラリーは、パフォーマンス、精度に優れており、低消費電力で、コードサイズは高度に最適化されています (通常は関数あたり 1KB 未満)。一般的な CMSIS-DSP* ライブラリー・ベースの DSP ライブラリーには、基本的な演算、高速演算、複雑な演算、統計、変換、補間、および行列関数が含まれています。インテルが最適化した LibM には、sqrtf、expf、logf、sinf、cosf、sincosf、tanf、asinf、acosf、atanf、floorf、ceilf、truncf など、一部の使用頻度の高い単精度関数が含まれています。GNU* 標準 C 数学ライブラリーに比べて、パフォーマンスは最大で 10 倍になり、コードサイズは最小で 5 分の 1 になります。

インテル® System Studio は、オープンソース GDB および OpenOCD* ソフトウェアを使用して、デバッガーとマイクロコントローラー・ファームウェア・フラッシュ・サポートを統合します。ハードウェア側の JTAG インターフェイスは、単純でコスト効率に優れた FTDI FT232H および FTDI FT2232H USB から JTAG/UART アダプターを使用して行われます。

クロスプラットフォームのツール

インテル® System Studio には、ターゲット・プラットフォームに関係なく、さまざまな組込み、モバイル、ウェアラブル、IoT、マイクロコントローラー・プラットフォームでシームレスに動作する、明確に定義されたツールのセットが用意されています。小規模なアプリケーションから、大規模なメニーコア・アプリケーションまで、これらのツールを使用することができます。この記事ではコンパイラーとライブラリーを取り上げていますが、同等のプラットフォームで動作する解析ツールも理解する価値があります。

詳細

- [インテル® System Studio](#)
- [インテル® System Studio for Microcontrollers](#)
- [インテル® MKL](#)
- [インテル® IPP](#)
- [インテル® System Debugger](#)
- [インテル® TBB](#)
- [インテル® VTune™ Amplifier for Systems](#)