

組み込み機器向けリアルタイムOS Windows CE 6がやって来る

杉本 拓也 / 松井 敏訓



後編



Visual Studio 2005 で開発する

1. Visual Studio 2005 で アプリケーションを開発する

Windows CE は、これまで専用の開発環境として Platform Builder, eMbedded Visual C++ が必要でした。しかし、Windows CE 6 からは、開発環境が Visual Studio 2005 に統合されました。今回は、Windows CE 6 の開発環境に注目して解説を進めていきます。

Platform Builder / eMbedded Visual C++ から
Visual Studio 2005 へ

開発環境が Visual Studio 2005 になったことで、通常の Windows アプリケーション以外に、以下のような開発が可能となりました(図1)。

▶ Windows CE 6 の OS 開発

これまで Platform Builder で行っていた作業を Visual Studio 2005(図2)上で行えます。OAL, ドライバなどの OS を構成する基本モジュールの開発, および Windows CE 6 で提供するコンポーネントの抽出や選択が可能となります。

▶ Windows CE 6, Windows Mobile 上のアプリケーション開発

従来は eMbedded Visual C++, Visual Studio 2003 で行っていたアプリケーション開発が可能となりました。

開発言語としては、Visual C#, Visual Basic.NET を利用するマネージド・コード(コラム1を参照)のアプリケーション開発以外に、これまで、eMbedded Visual C++ でしか行えなかった C/C++ によるネイティブ・コード・アプリケーションの開発も行えます。

組み込みシステム開発における 製品ラインナップの注意点

Visual Studio 2005 の製品ラインナップでは、Professional Edition, Team System が Windows CE のアプリケーション開発をサポートしています。開発言語としては、Visual C#, Visual Basic, Visual C++ を利用できます。

Platform Builder との違い

Windows CE 5 までの開発環境である Platform Builder との大きな違いは、Shell にあります。これまでは、専用のユーザ・インターフェースによる開発であり、エンジニアはアプリケーションと OS の開発環境を切り替えて開発する必要がありました。しかし、今回の Windows CE 6 からは、OS とアプリケーションを同一のユーザ・インターフェースで開発できます。そのため、複数の開発環境を導入する必要がなくなりました。

Visual Studio .NET 2003 との違い

旧バージョンの Visual Studio .NET 2003 との大きな違いの一つとして、ツールの構成が挙げられます。

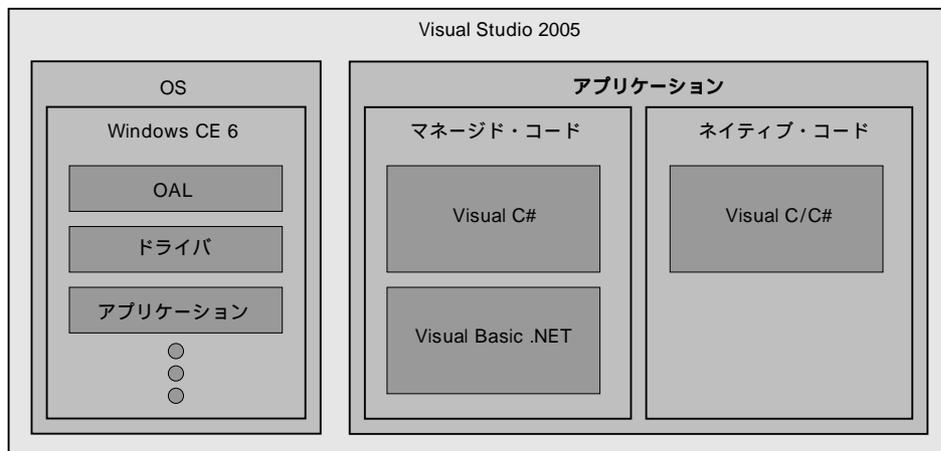


図1
Visual Studio 2005 で開発を行う場合の
概念図

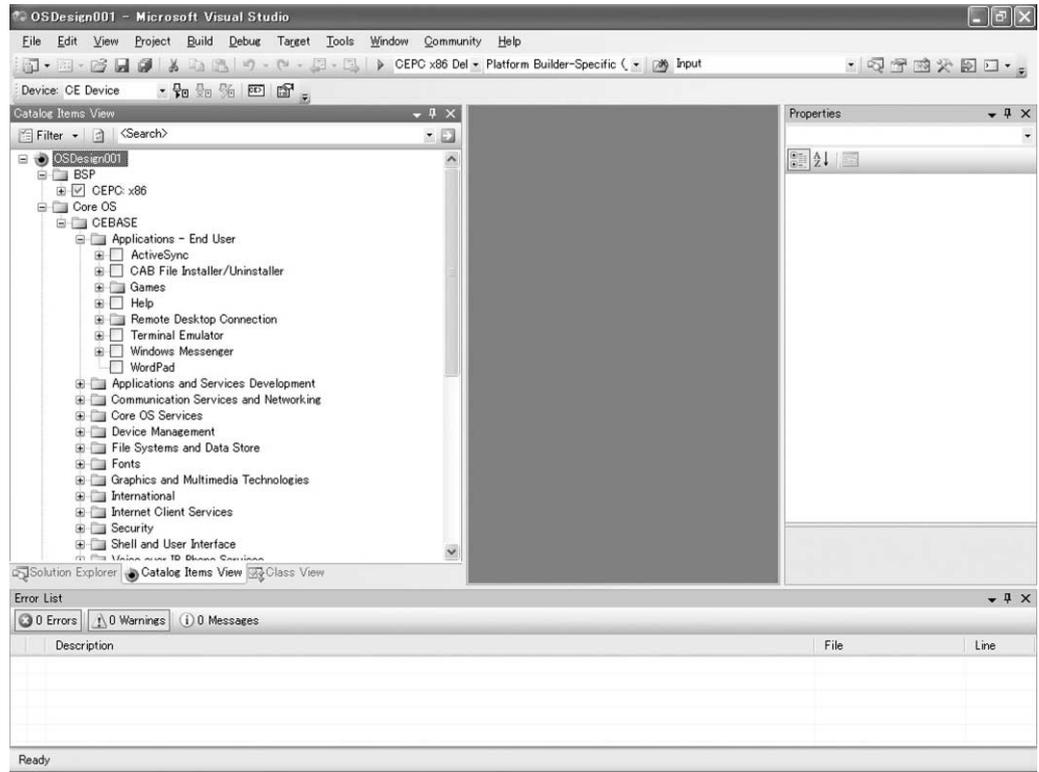


図2
Visual Studio 2005 の画面

Visual Studio .NET 2003 においては、Visual C# と Visual Basic が利用できました。ネイティブ・コードの開発には、Visual Studio .NET 2003 とは別に、eMbedded Visual C++ を利用する必要がありました。

Visual Studio .NET 2003 では、Visual C# や Visual Basic .NET を利用してマネージド・コード・アプリケーションを開発し、eMbedded Visual C++ でネイティブ・コード・アプリケーションを開発する構成となっていました。一方、Visual Studio 2005 では、両方のコードの開発が行えるようになっていきます。

組み込みシステム開発

マネージド・コード・アプリケーションは、.NET Compact Framework で提供されるクラス群や CLR のガベージ・コレクションによるメモリ・リソース管理が利用できるという利点もあります。 .NET Compact Framework が提供している機能のみ利用することも可能です。そのため、たとえばデバイス固有の処理などにおいて、.NET Compact Framework では提供されていない機能はネイティブ・コードを利用することになります。

マネージド・コードでのアプリケーション開発は、先に説明したとおり、さまざまな利点があります。しかしながら、数 ms

コラム 1

マネージド・コードとネイティブ・コード

Windows CE のアプリケーションには、マネージド・コード・アプリケーションとネイティブ・コード・アプリケーションの 2 種類が存在します。

CPU に依存しないマネージド・コード・アプリケーション

マネージド・コード・アプリケーションは .NET Framework のサブセットである組み込みシステムおよびモバイル端末向けの .NET Compact Framework 上で動作するアプリケーションです。

マネージド・コード・アプリケーションの実行モジュールに

は、IL と呼ばれる中間コードが格納されています。この中間コードは .NET Compact Framework の CLR (Common Language Runtime) によって各ハードウェア固有のネイティブ・コードに変換され、実行されます。 Windows CE では、4 種類の CPU をサポートしていますが、この CLR によって CPU 命令の違いを吸収しているため、CPU ごとに実行モジュールを用意する必要がありません。

CPU に依存するネイティブ・コード・アプリケーション
それに対して、ネイティブ・コード・アプリケーションでは、プロセッサごとに固有のネイティブ・コードが格納されるため、サポートするプロセッサごとに実行モジュールを用意する必要があります。