

文字列処理 / 画像描画 / プロセス間通信…全く同じようには使えません

# パソコン用 .NET Framework と マイコン用 .NET Micro Framework の違い

目高 亜友, 太田 寛

Windows プログラマ必見!

.NET Micro Framework 上で動作する C# は、言語仕様としては Windows 上で動作する .NET Framework と同じです。しかし、文字列処理や画面描画、プロセス間通信などのクラス・ライブラリは違います。そのような使用上の注意点などについて解説します。(編集部)

筆者らが .NET Micro Framework 3.0 の評価を始めたのが約 4 年前、本誌に初めて記事を書いたのが約 3 年前の 2009 年 12 月号<sup>(1)</sup>です。そのころは用途というとホビーや研究・実験用が多く、あまり普及の兆しが見えませんでした。その後バージョン 4.0 でオープン・ソース化され、去年ぐらいいからは日本でも少しずつ普及し、業務用システムにも利用されるようになりました。

ハードウェアの基本的な条件さえ満足すれば、短時間で容易にアプリケーション開発ができるため、現在では .NET Micro Framework (以降、.NETMF) は、多くの業務用組み込みシステムのアプリケーションに利用されるようになってきています。

本章は、このような開発経験の中で得た Full 実装の Windows 版 .NET Framework (以下、Full.NET) および C# との違いや、注意点などのノウハウを紹介します。

## ● コンパイラは同じだが…

もともと .NETMF は、Full.NET と同じコンパイラ (Visual C# 2010 SP1) を使用するため、言語仕様は同じです。ハードウェアがパソコンとは異なるのは当然なのですが、そのほかに異なる点は主に、クラス・ライブラリでサポートされている機能ということになります。

Full.NET 上で C# プログラミングを行う開発者が .NETMF を使用して組み込みシステムの業務アプリケーションを開発する場合、痛切に感じる違いは特に次のあたりではないでしょうか。

- 文字列処理・編集機能
- WPF 画面描画
- IPC (Inter-Process Communication ; プロセス間通信)
- ネットワーク通信

## ● その 1 : 文字列処理・編集機能の不足

Full.NET と比較した場合、文字列処理・編集では主に次の機能が不足しています。

- 文字コードの変換 (エンコード / デコード) 機能
- バイト配列を String 型に変換する機能
- 変数値を文字列に編集表示する機能

これらの文字列処理機能は、慣れてしまえば問題となることはそれほどないのですが、C# のプログラマがはじめて .NETMF を使う際には注意が必要です。特に編集の文字列表示処理にはかなり不自由な点があるので、文字列処理を多用するシステムを開発する場合には事前に確認をしておくといでしょう。

必要に応じて筆者らは、.NETMF を公開している

表 1 文字列処理関連クラス・ライブラリ

クラス名	内容
GetString	バイト配列の文字列変換
XnString, X2String	バイト配列の 16 進文字列変換
EqualsIgnoreCase	大文字小文字を無視した文字列比較

(a) String Extensions

クラス名	内容
StringBuilder	バイト配列の文字列変換
Replace	部分文字列の置き換え
StringUtility.Format	Full 版と同様な   中括弧   編集表示
StringUtility.IsNullOrEmpty	空文字列検査
Parse.TryParseInt	文字列操作と変換

(b) Common Extensions