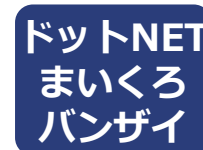


## オープンソースになったマイコン開発環境 .NET Micro Frameworkの紹介。

- 簡単に使える Microsoft の小型組み込みOS -

ドットNETまいくろバンザイ倶楽部  
株式会社デバイスドライバーズ  
日高亜友      [info@devdrv.co.jp](mailto:info@devdrv.co.jp)



# セッション内容

- はじめに
- 基本事項
- 新プラットフォームへの移植
- アプリケーション開発

# はじめに:参考情報

- インターフェース誌 2009年12月号 特集
  - **第4章 .NET Micro Frameworkによるネットワーク端末の製作**
- インターフェース誌 2011年3月号
  - **特集 組み込みで使うVisual C#と.NET Micro Framework**
- インターフェース誌 2011年4月号
  - **技術解説 Linux上への.NET Micro Framework移植の勘所(前編)**  
技術解説 SH-2Aマイコン基板に.NET Micro Frameworkを移植する
- インターフェース誌 2011年6月号
  - **技術解説 Linux上への.NET Micro Framework移植の勘所(後編)**
- インターフェース誌 2011年11月号
  - 技術解説 SH-2A向けネットワーク対応.NET Micro Frameworkの活用
- インターフェース誌 2012年8月号
  - **特集 定番Visual StudioとC#でマイコン開発にトライ!**
- インターフェース誌 2012年9月号
  - 技術解説 定番Visual StudioでルネサスRX マイコン開発

# .NET Micro Framework とは？

- Microsoft Research 開発の組み込みOS
- Visual C# / Visual Basicで開発した IL  
.NET アセンブリを中間コードインタプリタで実行
- 豊富なドライバとライブラリを標準装備
- オープンソース・ライセンスで公開(Apache V2)
- 20種以上のプラットフォームに移植されて動作
  - .NET Micro Framework (NETMF) 移植済みSOC
  - 業務アプリケーション・システム応用事例
- アプリケーション開発が**”超”**簡単な組み込みOS

# はじめに:用語

- NETMF
  - .NET Micro Framework
  - 今日のメインテーマ, SPOT とも
- PK
  - Porting Kit
  - 動作プラットフォームを作成するためのソースコード付ツールキット
- SDK
  - Visual Studioに組み込むアプリケーション開発環境

# セッション内容

- はじめに
- **基本事項**
- 新プラットフォームへの移植
- アプリケーション開発

# .NET Micro Framework の歴史

- 2001年 V 1.0  
SPOT Watch / MSN Direct Service  
(Microsoft リサーチ)
- 2006年 V2.0  
Windows SideShow Device
- 2008年 V3.0 , 3.5 for Network
- 2010年 V4.0 for Open Source
- 2010年 V4.1 for Big Endian / lwIP
- 2011年 V4.2 for Visual Basic
- 2012年 V4.3 for Visual Studio 2012

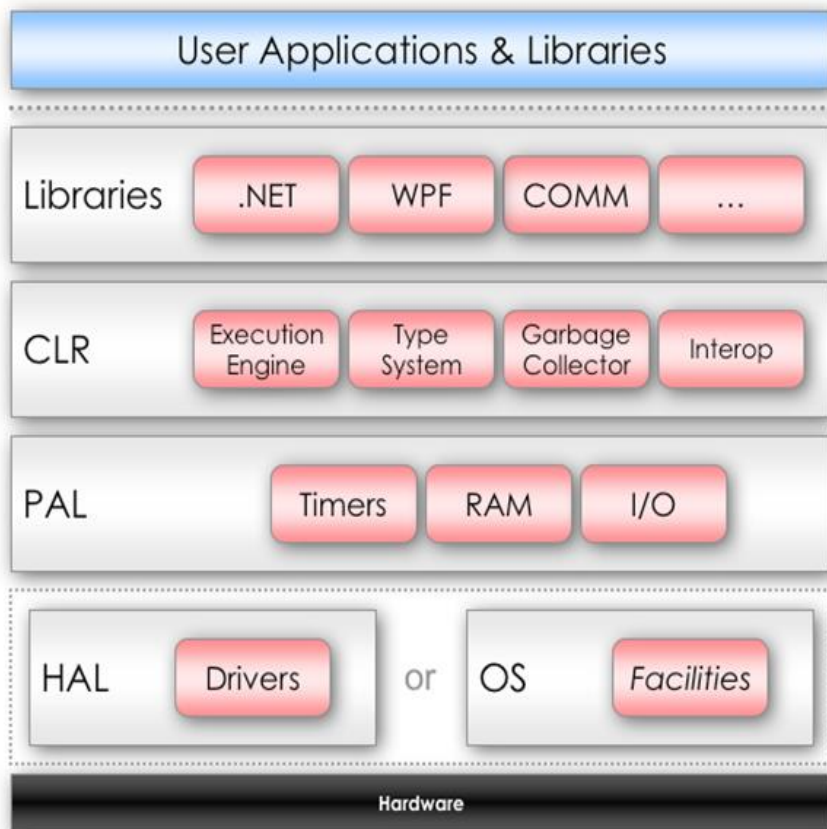


# 私とNETMFのかかわり

- 2008年にソースコード (V3.5) 購入&評価
- 2009年以降、毎年関連記事執筆
- NETMF関連事業
  - GHI Electronics社 正規代理店(2010年より)
    - 製品群 <http://www.ghielectronics.com/catalog>
  - NETMF 関連受託開発
    - ボード開発、移植作業、アプリケーション開発
- 各種普及活動
  - イベント、セミナー、勉強会、雑誌記事等
  - 日本マイクロソフト株式会社との連携

# アーキテクチャ

## ■ 全体構成



アプリケーション開発  
ライブラリ開発

プラットフォーム移植



**2種類の実装方法**

- Native Platform
- “OS on OS” solution

CLR: Common Language Runtime  
HAL: Hardware Abstraction Layer  
PAL: Platform Abstraction Layer

# OSとしての提供機能

- IL (.NETのアセンブリ) の実行環境 = **TinyCLR**
  - タイマ, 割り込みハンドラ, マルチスレッド, スケジューラ
  - ブートローダ, 各種デバイスドライバ, リソースマネージャ
- 組み込み環境対応
  - MMU 不要
  - 最小 64KB RAM / 256KB ROM (特別な環境)
  - 標準サポート ARM7/9/Cortex-M3, Blackfin, SH-2/2A
- サポート機能
  - ブートローダ, アプリケーションローダ, デバッガIF

# 移植方法の選択

## ■ Native実装

TinyBooter = NETMF専用ブートローダ

- TinyCLR直接実装 VS. TinyBooter実装
- TinyCLR / TinyBooter のメンテナンス手順が必須

## ■ 他のブートローダ利用

- U-Boot, T-Monitor (T-Engine)等
- Network ブート

## ■ OS on OS

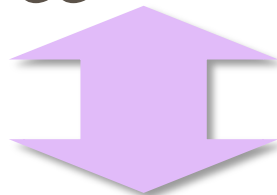
- 他のOSのアプリケーションとしてNETMFを実装

## ■ ソースコードが公開されているので 実際は何とでも作ることが可能

# SDK vs. PK (Porting Kit)

- SDK: 組み込みアプリケーション開発
  - アプリケーション開発に必須
  - Visual Studio / VC# Express に組み込んで使用
  - Simulator, Debugger, Deployment Tools を含む

バージョンを  
合わせる必要がある



- PK: 動作プラットフォームの構築(移植)
  - .NET Micro Framework の移植に必須
  - ほかにクロスコンパイラ(+デバッガ)が必須
  - ツール類を含めた全ソースコードが提供
  - ほとんどを C++ で記述 (一部アセンブラ, C, C#)

# サポート・プラットフォーム

- PK が用意しているサポート・テンプレート(標準サポート環境)
  - BF537 -- ADI Blackfin
  - EA\_LPC2478 -- NXP LPC2478 (Embedded Artists LPC2478 4MB / 512KB)
  - iMXS -- MC9328MXS (32MB RAM / 8MB Flash ROM)
  - **MCBSTM32E – STM32F103 Cortex-M3 (KEIL MCBSTM32E 8MB / 1MB)**
  - MOTE2 -- Marvell PXA271 (Crossbow imote2)
  - PCM023 -- NXP LPC22xx ARM7
  - **SAM7S\_EK -- Atmel AT91SAM7S256 (64KB RAM / 256KB Flash ROM)**
  - SAM7X\_EK -- Atmel AT91SAM7X512 (128KB RAM / 512KB Flash ROM)
  - **SAM9RL64 -- Atmel AT91SAM9RL64EK (64MB RAM / 256MB Flash ROM)**
  - **SAM9261\_EK -- Atmel AT91SAM9261EK (64MB RAM / 256MB Flash ROM)**
  - SH7216\_RSK -- SH-2A 16MB SRAM / 1MB Flash ROM
  - SH7264\_M3A\_HS64 -- SH-2A 64MB RAM / 516MB Flash ROM
  - **SH7264\_RSK -- SH-2A 64MB RAM / 64MB Flash ROM**
  - SH7619\_EVB -- SH2 8MB RAM / 4MB Flash ROM
  - STM32Stamp -- STM32F103 Cortex-M3 (ET-STM32 Stamp 512KB RAM / 64KB)
  - Windows2 -- Windows Emulator

**デモ予定: SAM9261\_EK, SH7264\_RSK**

# ドライバとサービス

## デバイスコントローラ

- Cache
- Ethernet, WiFi
- GPIO
- I2C
- INTC
- LCD / Display
- Piezo / Speaker
- PMC
- SDRAM
- SDCARD / MMC
- SPI
- Thermistor
- TIME (RTC)
- TIMER
- USART
- USB Client

## デバイスドライバ

- Backlight (GPIO,SPI)
- BatteryCharger / Measurement
- BlockStorage (Flash, SD)
- Buttons
- COM
- **DPWS**
- **Gesture**
- Font, Graphics
- HTTP
- Ink
- Interop
- FileSystem (FAT)
- Security, Crypto
- TCP/IP, Socket, **RPC, SSL**
- Touch Panel (**Multi Touch**)
- **XML**

# セッション内容

- はじめに
- 基本事項
- **新プラットフォームへの移植**
- アプリケーション開発

# 移植前のチェック

- ターゲット・プラットフォーム
  - RAM/RAM容量 (ROM不足は RAMで代用可能)
  - CPU アーキテクチャ
  - gcc利用時の最低メモリ目安
    - RAM 256KB / ROM 512KB (Thumb / Thumb2)
- コンパイラとデバugga
  - Windows上で動作するクロスコンパイラ
- Native 実装 vs. “OS on OS” 実装
  - ターゲット・プラットフォームの適正
  - ツール類の準備

# 移植に必要な知識

- 組み込みシステム全般
  - 割り込み処理, 割り込みVector, Cache, MMU, PLL
  - ブートローダ
- 移植対象のCPU, ターゲット・ボード
  - データシート, 回路図, メモリマップ, クロック
- ツール
  - Toolchain (コンパイラ, アセンブラ, リンカ, ライブラリアン)  
C, C++, CPUのアセンブラ命令, コマンドプロンプト, VB Script
  - デバッガ, ICE, Eclipse, OpenOCD+GDB
- OSアーキテクチャとリソース管理
  - メモリ, タイマ, 割り込み, イベント
  - .NET Framework と CLR

# 開発環境の準備

## ■ ソフトウェア環境

- Visual Studio 2012 (VC++ 2012 exp, VC# 2012 exp も可)
  - Windows SDK 8.0 (MSBuild)
- .NET Framework 4.3 SDK
- .NET Micro Framework 4.3 Porting Kit
- クロスコンパイラ, デバッガ (gcc, gdb, ...) 他のgcc, gdbは各自で対応させる必要有
  - PKでの検証済 GDB (4.2.1)  
CodeSourcery arm-2007q3-53-arm-none-eabi.exe
- その他のツール

## ■ ハードウェア環境

- ターゲット・ボード(プラットフォーム)と **JTAG-ICE**

# 移植で使用するその他ツール

## ■ デバッグ

- OpenOCD
- gdb (Compiler 付属)
- Eclipse (IDE, ソースコードデバッグ)
  - JAVA (Eclipse 動作に必須)

## ■ ソースコード管理

- 7zip, WinMerge, VSS, svn, ...

# 移植の主要作業

- 開発環境構築と基本設計
  - メモリ割当て設計 & Scatter ファイル設定
  - Solution の作成
- ビルド & デバッグ
  - NativeSample を動作させる
    - ブートローダ, 初期化処理 (Clock, MMU, PLL)
    - ドライバ, PAL IF
  - TinyCLR の動作
    - TinyBooter, アプリケーション・ロードとデバッグIF
  - 必要に応じて: TinyBooter の動作

# 開発環境構築の実際

- クロスコンパイラのインストール
  - クロスコンパイラはScriptを修正・作成することでWindows上で動作するものであれば全て利用可能
- PKのインストール(要 Custom Install)
- 開発用コマンドプロンプトの作成
  - 環境変数の設定
  - ビルドスクリプトの作成・修正
    - setenv\_????
    - tools¥Targets¥Microsoft.Spot.system.???.targets
- デバッグ環境のインストール(オプション)

# ビルド&デバッグ

## ■ NativeSample

- Native Code での”Hello world” 表示出力
- ビルド環境の検証(自由にカスタマイズ可能)
- ドライバ類の初期化&動作の検証

## ■ TinyCLR

- .NET Micro Frameworkの移植対象・本体
- 実行にはIL (コンパイル済アセンブリ)が必須
  - Product¥Sampleディレクトリ (HelloWorld, Network など)

## ■ TinyBooter

- NETMF 純正ブートローダ (MFDepoy IF)

# Flash ROM インターフェース

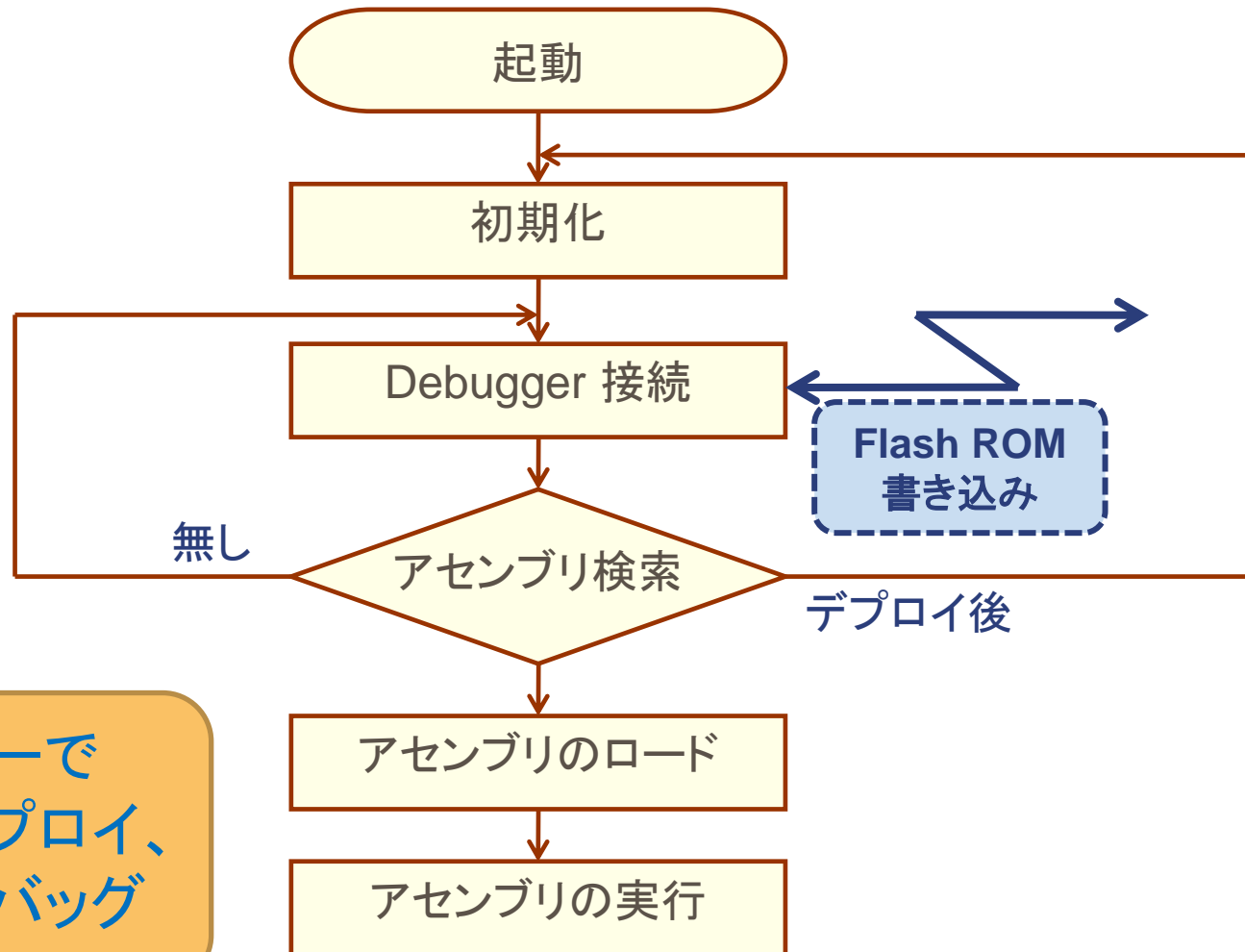
## ■ 必用機能

- ROMの読出し、書き込み、消去
- ブロック単位のアクセス
- ROM/RAM展開
- 圧縮・伸長

## ■ 導入先

- TinyBooter, PortBooter, TinyCLR

# TinyCLRとアセンブリの実行動作



‘F5’ キーで  
ビルド、デプロイ、  
実行、デバッグ

# 考察: Native 環境への移植

## ■ 利点

- 全ソースコードが用意されている
  - ブートローダ, CPU初期化, リソース管理
- 動作している実機サンプルが入手し易い
- 割り込み / イベント処理が実装し易い

## ■ 欠点

- 未対応アーキテクチャには移植し難い
- クロスコンパイル, デバッグ環境を手当てする必要

# 考察：OS on OS 環境への移植

## ■ 利点

- OSが用意している開発環境を流用可能
- JTAG-ICEの代わりにデバッガで代用可能
- 1回移植すれば、同一OSへの移植が容易

## ■ 欠点

- 参考にするソースコードが用意されていない
- 動作している実機サンプルが入手し難い
- OS依存APIを変換する作業(PAL層開発)が必須  
(Timer, Event, Select, Socket, ...)
- ドライバの移植数、作業工数はNativeよりも多い

# セッション内容

- はじめに
- 基本事項
- 新プラットフォームへの移植
- **アプリケーション開発**

# アプリケーション開発の基本

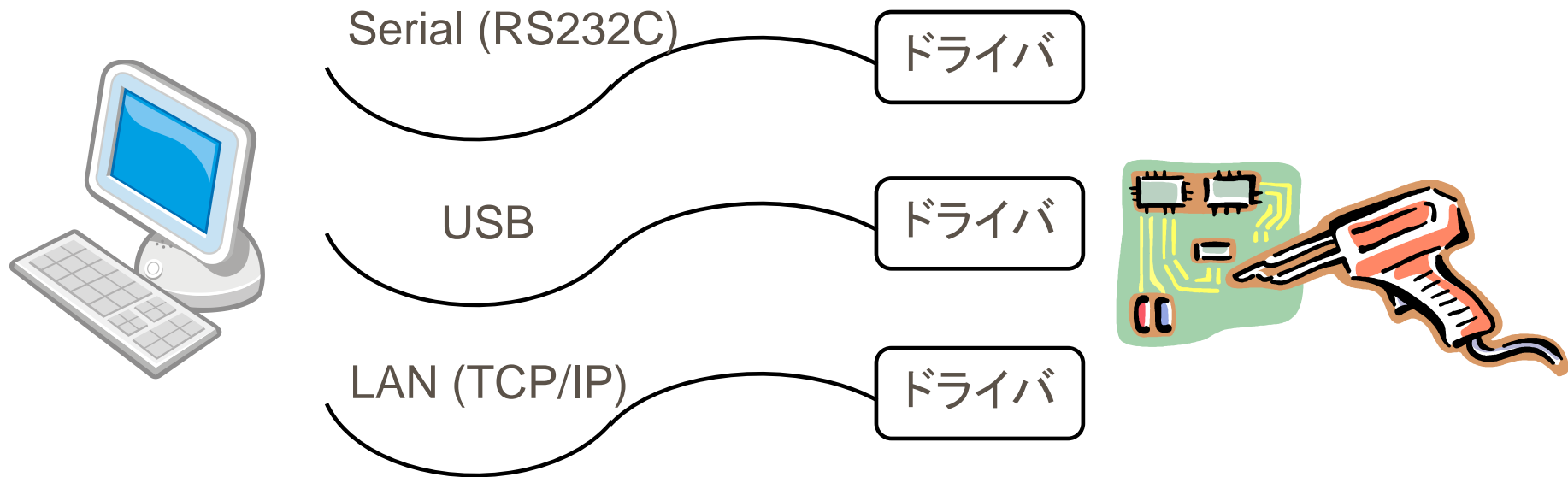
- C# での開発が基本
  - Visual Basicも利用可能
  - C, C++, Assemblerは使えない
- Flash ROM 書き込みデバッグが前提
- エミュレータ
- 強力なソースコード・デバッガ
  - .NET アセンブリの送りこみ(デプロイ)
  - オンライン・デバッグ
- ICE不要(使えない)

# アプリケーション開発手順

- 事前調査・基本設計
  - Interface 仕様の確認 (H/W, S/W, Network, ...)
- コーディング
  - Full .NET 環境用コードの流用
- デバッグ
  - エミュレータ, オンライン・デバッガ
  - Full .NET 環境でのデバッグ
- デプロイ
  - ROM焼き, 配布用署名付きコード生成

# デバッガ インターフェース

## ■ デバッガとの接続手段：3種類から選択

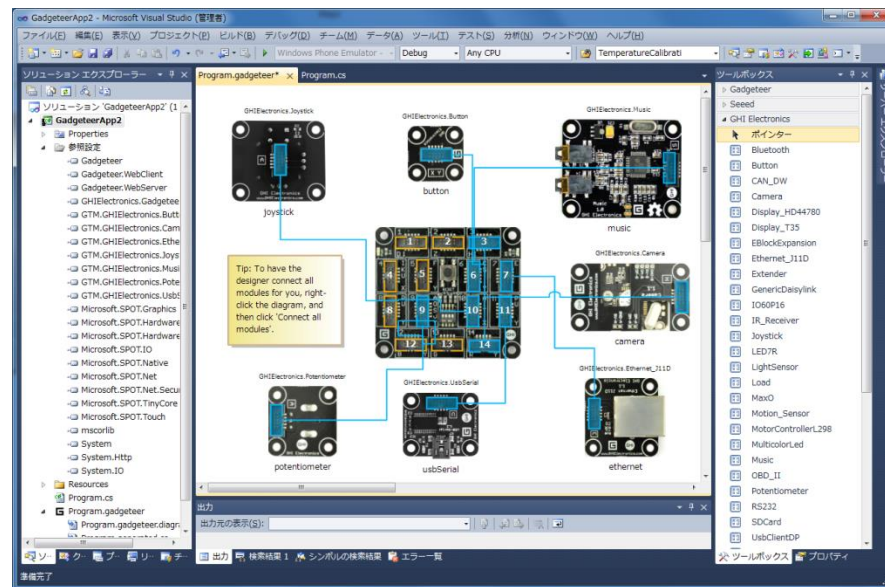
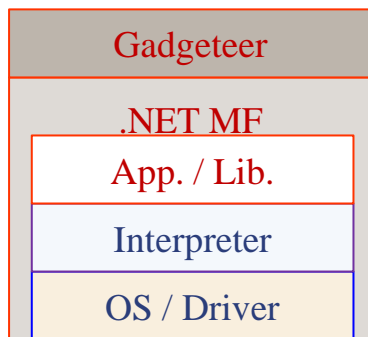


## ■ 独自ハードウェア開発の注意点

- OS更新用のデバッガ(ブート)・スイッチの必須
  - 通常はGPIOタクトSW(ボタン), DIP-SW等を使用

# .NET Gadgeteer とは？

- Microsoftリサーチが開発している  
.NET Micro Framework ベース組み込みシステム統合開発環境とそれを構成するモジュール製品群
- MMU無しマイコン上でVisual Studioで開発した  
.NETアセンブリ(中間コード)をインタプリタ実行
- GUI操作で自動的に  
初期化コードとクラスライブラリを追加

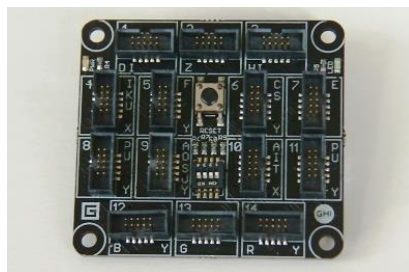


# .NET Gadgeteer とは？

## ■ 具体的には次のもの

### ■ ハードウェア

- 専用メインボード、専用モジュール  
開発環境としての Windows PC



### ■ ソフトウェア

- ROM化されたOS、組み込み用各種ドライバとライブラリ、  
Visual Studio用Plug In(ライブラリとGUIツール、シミュレータ)

## ■ 新規モジュール開発やカスタマイズ可能

### ■ オープンソース












# .NET Gadgeteer とは？

- C#/VB言語レベルでライブラリと豊富なミドルウェアを提供
  - プロトコル、グラフィック
  - ファイルシステム
  - サービス、マルチスレッ

- 6種のCPUボードと60種以上のアクチュエータ、センサー、ネットワークアダプタ、コネクティビティ、ディスプレイ等のモジュールを提供

## Sensor

 <p><b>Button Module</b> Button Module for Gadgeteer-compatible devices with on-board LED.</p> <p>1 <a href="#">Add to Cart</a></p>	\$4.95	 <p><b>Joystick Module</b> A hat-type analog joystick with push-button.</p> <p>1 <a href="#">Add to Cart</a></p>	\$6.95	 <p><b>Potentiometer Module</b> Good for measuring position or controlling levels.</p> <p>1 <a href="#">Add to Cart</a></p>	\$8.95
 <p><b>Accelerometer Module</b> Measures Acceleration.</p> <p>1 <a href="#">Add to Cart</a></p>	\$12.90	 <p><b>Barometer Module</b> Barometer Module</p> <p>1 <a href="#">Add to Cart</a></p>	\$33.00	 <p><b>Compass Module</b> Digital Compass.</p> <p>1 <a href="#">Add to Cart</a></p>	\$15.90
 <p><b>Current ACS712 Module</b> Used to measure higher voltage currents.</p>	\$19.95	 <p><b>Current Module</b> Non-invasive Current measurement.</p>	\$21.00	 <p><b>GasSense Module</b> Detects gasses in the air</p>	\$14.95

# 事例：つぶやきステーション

## ■ インターフェース 2012年8月号 特集

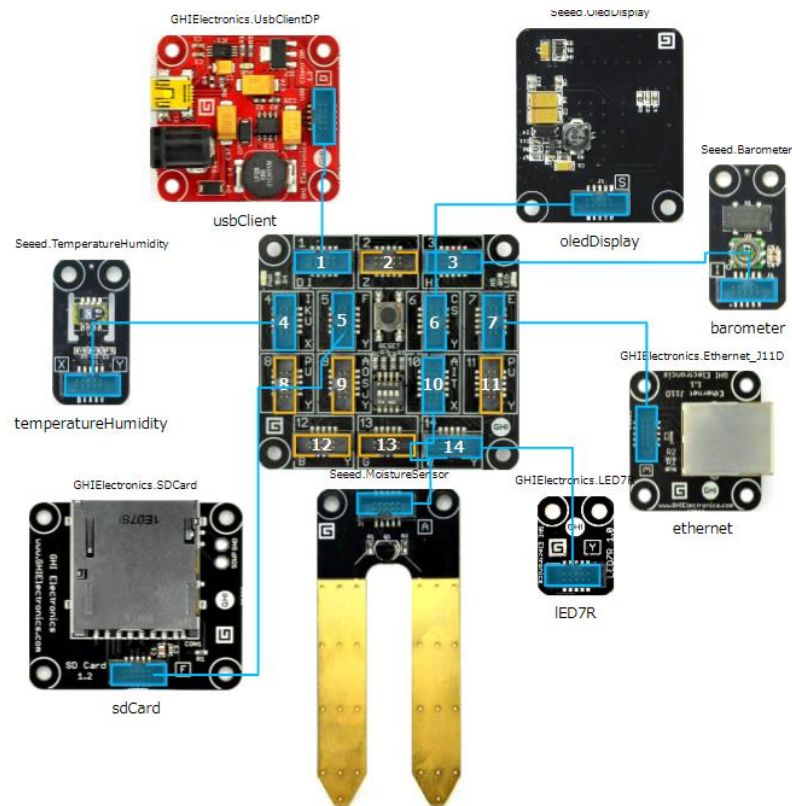
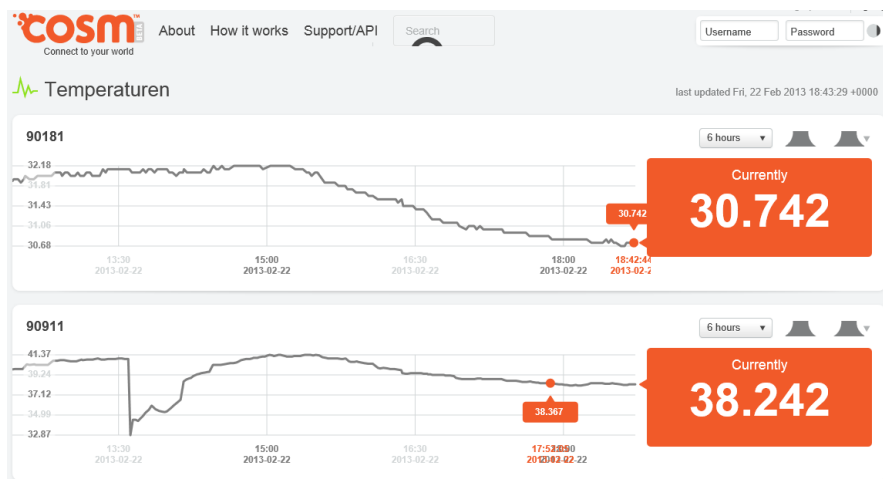
少ないメモリでリッチなライブラリのオープン・ソースOSを試す  
定番Visual StudioとC#でマイコン開発にトライ! ~Windowsプログラマでもできる~

## ■ 第3章 気圧/温度/湿度をTwitterに全自動つぶやき 出来あい品ですぐに試すWebウェザーつぶやきステーションの製作



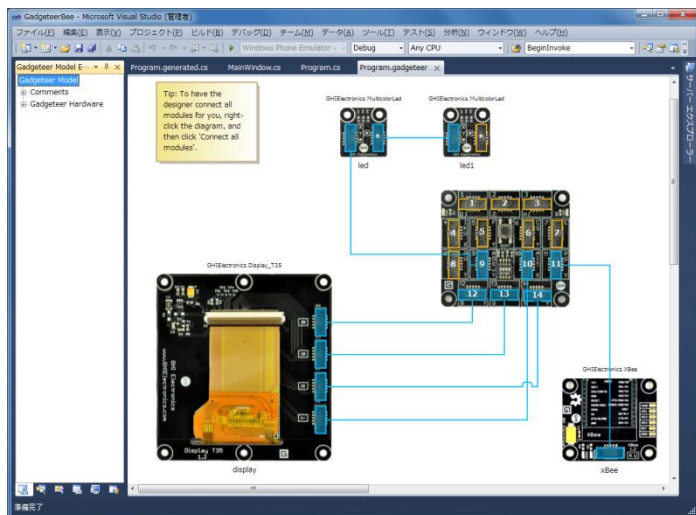
# 事例 : Connecting your station to the cloud

- Microsoft Research 提供の教材
- COSM.COM Pachube (パッチベイ) への接続



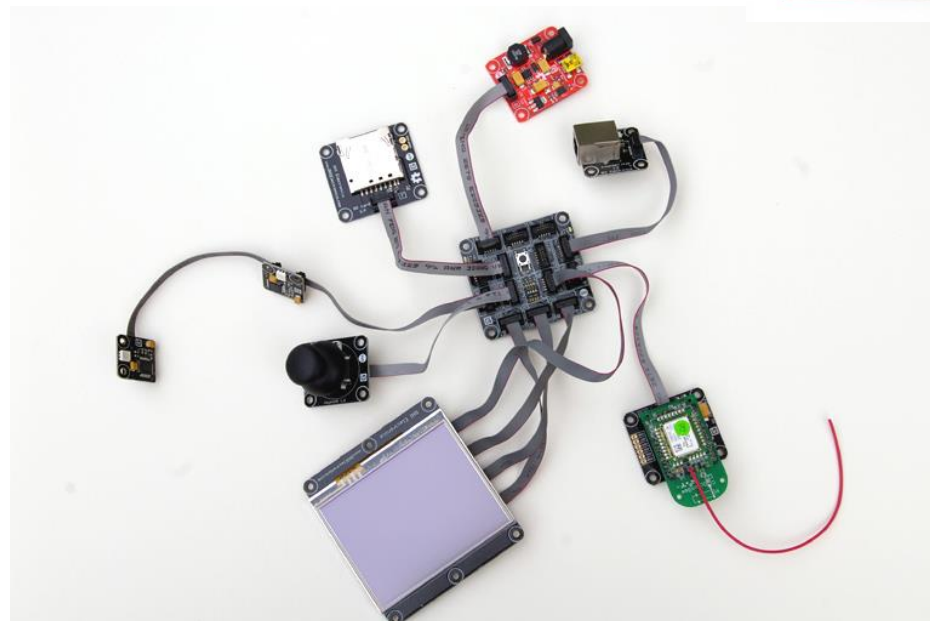
# 事例：OceanBeeとGadgeteer

## ■ 開発期間の短縮



- EnOcean無線データ受信
- 新無線通信方式検証やプロトタイピングに最適

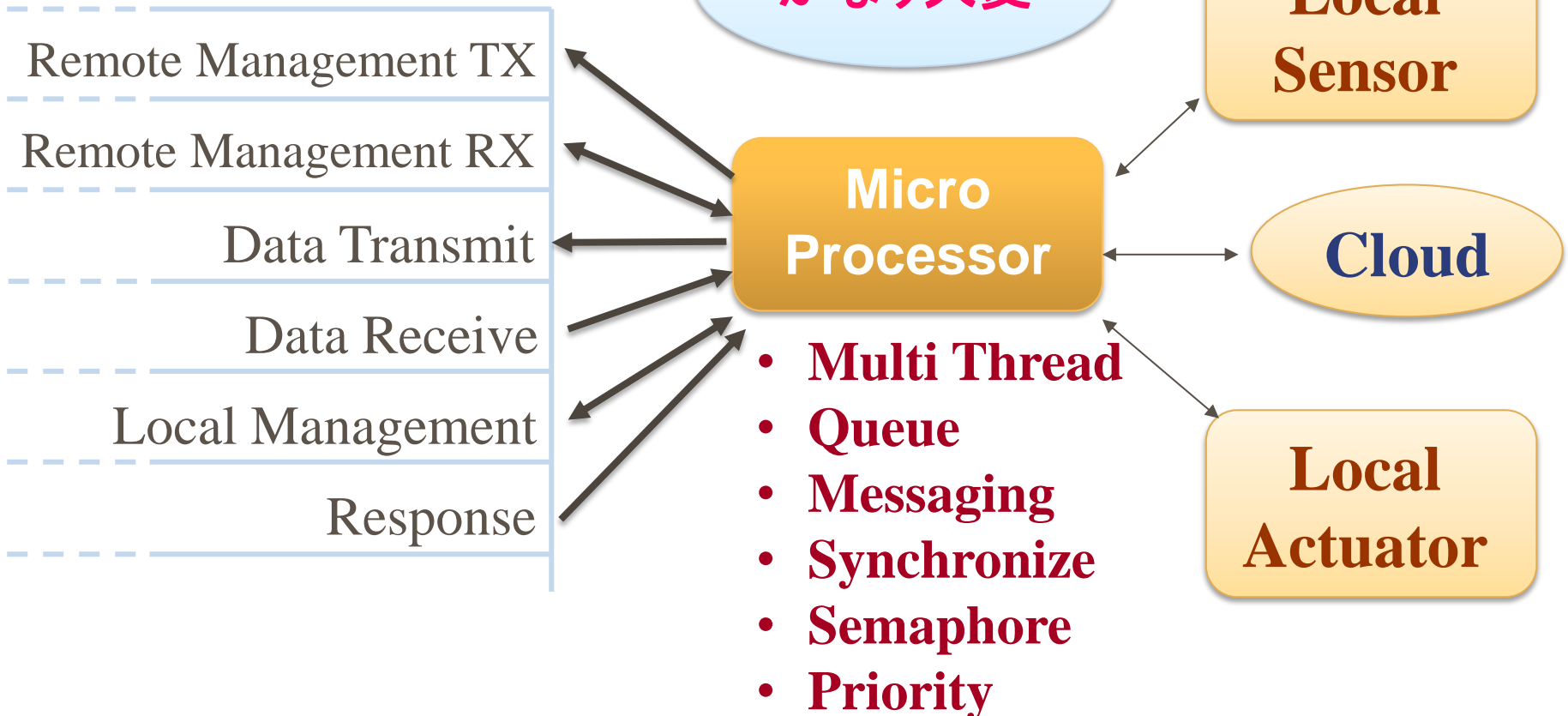
OceanBeeとは？  
XBeeピン互換  
EnOcean Transceiver



# 事例: EnOceanセンサーシステム

## ■ アプリケーション事例

**TCM3x0C / ESP3**



- はじめに: 参考情報
- 基本事項
  - .NET Micro Framework 概要
- 新プラットフォームへの移植
  - 一般的な移植手順
  - 移植に関する考察
- アプリケーション開発
  - 開発手順、事例

資料は  
[www.devdrv.co.jp](http://www.devdrv.co.jp)  
で後日公開

- Porting Kit 付属の説明書
  - RCLPort.chm
- GHI Electronics社
  - **Beginners' Guide to Porting NETMF**  
<http://www.ghielectronics.com/downloads/FEZ/Beginners%20Guide%20to%20Porting%20NETMF.pdf>
- インターフェース誌
- Eclipse デバッグ関連
  - How to use Eclipse and GDB to debug the .NET Micro Framework  
[http://discovertheexperience.blogspot.jp/2008/11/how-to-use-eclipse-and-gdb-to-debug-net\\_09.html](http://discovertheexperience.blogspot.jp/2008/11/how-to-use-eclipse-and-gdb-to-debug-net_09.html)
  - Configuring Eclipse + OpenOCD + GCC to Debug NativeSample  
<http://msmicroframework.blogspot.jp/2009/02/configuring-eclipse-openocd-gcc-to.html>
  - PS3とLinux、電子工作も  
<http://todotani.cocolog-nifty.com/blog/>

- 開発コミュニティと配布
  - NETMF Codeplex Projects  
<http://netmf.codeplex.com/>  
<http://netmf.codeplex.com/documentation>
  - Microsoft .NET Micro Framework  
<http://www.netmf.com/>
- おすすめサイト
  - Team Blog  
<http://blogs.msdn.com/b/netmfteam/>
  - Microsoft .NET Micro Framework Tools & Resources  
<http://informatix.miloush.net/microframework/>
  - GHI Electronics  
<http://www.ghielectronics.com/>
  - いまさら聞けない .NET Micro Framework入門 (@IT)  
[http://monoist.atmarkit.co.jp/fembedded/articles/netmicro/netmicro\\_01a.html](http://monoist.atmarkit.co.jp/fembedded/articles/netmicro/netmicro_01a.html)