



Visit our website.

ラズベリー・パイ Pico をモダンな言語で開発できる

Search the API reference

C#での開発を可能にする .NET nanoFrameworkの移植 前編

関本 健太郎

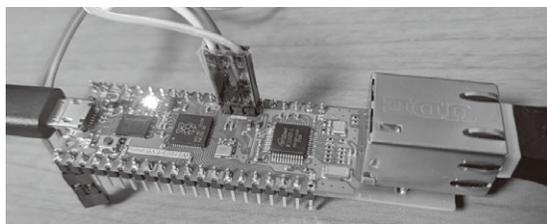
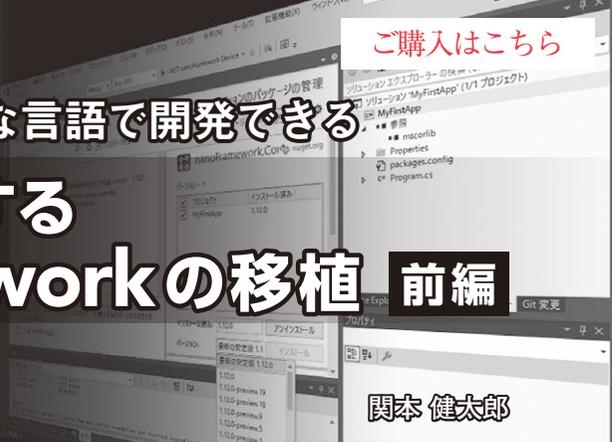


写真1 RP2040チップとTCP/IPコントローラW5100Sを搭載したW5100S-EVB-Pico

.NET nanoFrameworkを利用すると、C#でラズベリー・パイ Pico のアプリケーションを作成できます。 .NET nanoFrameworkは、ラズパイ向けなどにサポートされている.NET IoTライブラリ^{注1}のサブセット相当の機能を持ちます。

前編では.NET nanoFrameworkの概要を紹介し、ラズベリー・パイ Pico向けに移植した.NET nanoFrameworkの利用方法(C#プロジェクトの新規作成方法、HTTPクライアント・アプリケーション、Azure IoT Hubからのメッセージ取得方法)を解説します。後編では、その移植の手順を紹介します。

.NET nanoFrameworkは、RTOS環境を前提にしているため、RTOSとしてFreeRTOSを利用します。USB通信は本誌2022年10月号で解説したTinyUSBのCDCフレームワークを利用します。

移植ターゲットのラズベリー・パイ Pico の製品としては、非ネットワーク環境向けにラズベリー・パイ Pico、ネットワーク環境向けにW5100S-EVB-Pico (WIZnet, 写真1)を取り上げます。

.NET nanoFrameworkのあらまし

● 特徴

.NET nanoFrameworkは、RAMなどのリソースが制限された組み込みデバイス向けの.NET CLR (Common Language Runtime)で、.NET Base Class Libraryのサブセット版です。以下の特徴があります。

- Native マルチスレッド

- Interop コードのサポート
- .NET Core IoT 環境に移行されたUWP API
- RTOS上に複数のスレッドから構成されている
- JITは未サポート。標準の.NET assembliesは使用できない

2016年に.NET Micro Frameworkからスピニングしたメンバによって開発が続けられています。

● アプリケーション

.NET nanoFrameworkのアプリケーションは、.NET フレームワークの中間言語に変換されたもので、デバイスの(主に)フラッシュ・メモリ上のnanoCLRと呼ばれるライブラリ上で動作します。

開発時には、ホストPC上にnanoCLRと同等の機能を持つ(パッケージ・マネージャnugetで提供される).NETライブラリを取り込み、Visual StudioのnanoFrameworkのデバッガ・プラグインを介して、Visual Studioでアプリケーションを開発できます。

デバッグ時には、USB CDC通信でWireProtocolと呼ばれる.NET nanoFramework独自のデバッグ通信方式で、.NETアプリの中間言語をデバイス上のフラッシュ・メモリに書き込み、書き込んだ中間言語をステップ実行できます(図1)。

● アーキテクチャ

.NET nanoFrameworkのアーキテクチャは、下からハードウェア層、nanoCLR層、BCL (Base Class Library)、ユーザ・コード(C#)から構成されています(図2)。ハードウェア層は、主にマイコンやその周辺機能に依存する機能を実装しているライブラリで、C/C++、アセンブラで記載されています。

nanoCLR層のHAL層は、Hardware Abstraction Layerで、周辺機能を抽象化して実装しています。

PAL層はPlatform Abstraction Layerで、メモリ管理、ネットワーク、イベント、タイマ、I/Oを抽象化して

注1: <https://learn.microsoft.com/ja-jp/dotnet/iot/intro>

(1) nanoFramework getting started guide

<https://www.youtube.com/watch?v=iZdN2GmefXI>