

Pico SDKと連携するハイブリッド・システムの構築

豊山 祐一

第6部では、複数のCPUコアでそれぞれ異なったシステムを動作させるハイブリッド・システムを作成します。

ラズベリー・パイ Pico C/C++ SDK (以降Pico SDK)とTry Kernelをマルチコアで同時に実行することにより、Pico SDKの豊富なハードウェアの制御機能と、Try KernelのリアルタイムOSの機能のどちらもが使用できるようになります。

さらに、第2章以降では、Pico WのWi-Fi通信機能を使用したアプリケーション・プログラムをハイブリッド・システムで作ります。



● システムの構成とメリット

本章で作成するハイブリッド・システムは、図1に示すようにCPUコア0でPico SDKを動作させ、CPUコア1でTry Kernelを動作させます。それぞれのCPUコアで独立にユーザ・プログラムを実行し、CPUコア間の連携を行います。

Pico SDKは、Picoの公式のソフトウェア開発キット (Software Development Kit) で、C、C++、アセンブリ言語に対応します。公式のSDKだけあって、Picoのハードウェアを制御する各種ライブラリや、USB、Wi-Fi、Bluetoothといった通信の機能に対応するミドルウェアが含まれています。特に、Pico Wを使用する場合には、無線通信機能の提供はありがたいです。

一方、Pico SDKにはリアルタイムOSに相当する機能は含まれていません。補助的にFreeRTOSを実行できますが、Pico SDKそのものはマルチタスクに対応していません。Pico SDKで実行されるアプリケーション・プログラムは、Pico SDKの提供するライブラリなどを使用したベアメタルのプログラムとなります。

そこで、各CPUコアでPico SDKとTry Kernelを同時に動作させることにより、それぞれの機能を活用

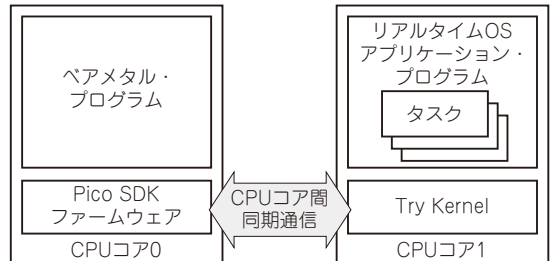


図1 Pico SDKとTry Kernelのハイブリッド・システム概要

できます。アプリケーション・プログラムは、Try Kernelのマルチタスク機能を活用することにより、リアルタイム性の実現が容易になり、タスクに分割した簡潔な記述も可能となります。また、ハードウェアの制御や無線通信などは、Pico SDKが提供する機能を利用することで簡単に実現できます。

● Pico SDKのバージョンについて

本記事で使用するPico SDKのバージョンは1.5.1 (2023年6月リリース)です。また開発ツールもこれに合わせたバージョンを使用します。

なおPico SDKは、2024年8月のPico 2発表と同時に2.0にバージョンアップしました。開発環境はVisual Studio Code (以降、VSCoDe) が主となり、Pico SDKやツールチェーンのインストールはVSCoDeの機能拡張からできるようになりました。

今回はPico SDK 1.5.1を使用しますが、Pico SDK 2.0はPico 2への対応が主ですので、PicoやPico Wを使う限りは特に問題はないと思います。

もしPico SDK 1.5.1などの旧バージョンのソフトウェアや開発ツールを使いたい場合は、GitHubから入手が可能です^{注1}。ドキュメントは更新されて以前の情報を参照できなくなっているところがありますので、記事の中で説明していきます。

注1: <https://github.com/raspberrypi/pico-sdk/releases/tag/1.5.1>