

アマゾンのバックアップで機能が充実!
複数の処理もマルチタスクでシンプルに書ける

第1章

FreeRTOS を載せる方法

石岡 之也

ワンチップ・マイコンでリアルタイムOSを用いる大きな利点としては、マルチタスクによって複数の処理を、あたかも同時に動かすようなプログラミングが可能になることだと思います。

Arduinoのように1つのループで複雑な処理を行うことも可能ですが、何かを処理中に別の処理を行うには、プログラミングや設計方法などの技術が必要になってきます。こういったときにRTOSを使うと、全ての問題を解決してくれるわけではありませんが、技術不足を補ってくれると思います。

リアルタイムOSの導入は楽ではありませんが、FreeRTOSのように利用者が多いものは、ウェブ上で情報が見つけやすかったり、掲示板へ質問したりすることで、解決も可能です。

RAMサイズが小さいマイコンでは、リアルタイムOSが占有するサイズがネックになることがあります。ラズベリー・パイPicoは、ワンチップ・マイコンとしては大きな264KバイトものRAMを搭載していることから、リアルタイムOSを有効に利用できると思います。

FreeRTOSは2000年代前半にリリースされたソースが公開されているリアルタイムOS (RTOS) です。マイコン・チップ・ベンダ各社が提供するサンプル・プログラムに同梱されていることがあります。また、安価なマイコン・モジュールに使われるなど、有名なRTOSの1つです。

2017年にアマゾンが買収し、付加機能が充実したこと、Wi-Fi搭載マイコン・モジュールESP32で使われていることから、日本でも名前をよく目にするようになりました。

● Pico向けFreeRTOS

GitHub上に、Pico向けのFreeRTOSが公開されています。

<https://github.com/PicoCPP/RPi-pico-FreeRTOS>

このソースコードを使って、

- ビルドに必要な環境構築

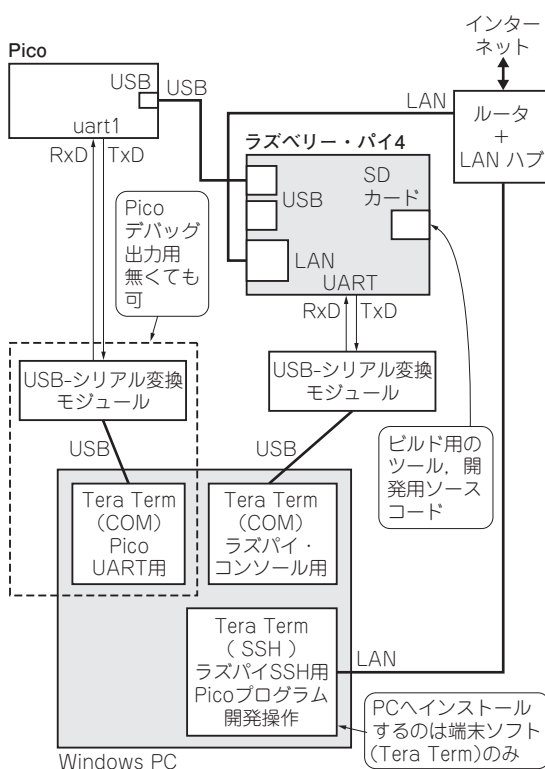


図1 Picoの開発環境はラズベリー・パイに構築した

- 必要なソースコードのダウンロード方法
- ビルド方法
- 製作事例

を紹介します。なお、執筆時点ではシングル・コアでの動作のようです。

開発環境

● 長期運用を考慮してラズベリー・パイにUbuntuを使って構築した

Pico向けのFreeRTOSの開発環境を、今回はラズベリー・パイ4とUbuntuを使って構築しました(図1)。ラズベリー・パイのLinuxは、最初にRaspberry Pi OS