

特集2 マイコンで動くフル機能 Rust

第1章

入出力/ネットワーク機能の面倒な初期化や手続き不要!
HTTPセンサ・ノード作りで使い勝手の良さを味わう

いよいよ試す環境が整った! フル機能 Rust × マイコン

中林 智之

フル機能 Rust をマイコンで使う メリット

特集2では、ESP32-C3 (Espressif Systems) を搭載したマイコン・モジュール M5Stamp C3U Mate と、Rust の標準ライブラリ std を使った組み込み Rust 開発を紹介します。

これまでマイコンで Rust を使うには no_std 対応の制限されたライブラリのみ使っていました。no_std はマイコンでも使えるよう資源の使用量が少なくて済む反面、使えない機能も多いという悩みがありました。

std を使った組み込み Rust 開発の利点は、Rust が標準ライブラリで用意している入出力機能やネットワーク機能を、面倒な初期化や手続きなしに組み込みシステムの開発で使うことができることです。

また、std が使える場合には、開発に利用できるライブラリ(クレート)の選択肢が飛躍的に増え、よりアプリケーションの実装に集中できます。その使い勝手の良さは std を使わない組み込み Rust とは一線を画するものです。

本稿を通じて、読者のみなさまに std Rust 環境の使い勝手の良さを味わっていただけると幸いです。

ESP32-C3 を選んだ理由

● RISC-V+ESP-IDF は std も Rust コンパイラも正式サポート

今回使用する RISC-V コアの ESP32-C3 の他、ESP32、ESP32-S2、ESP32-S3 で std Rust がサポートされています。

今回メジャーな ESP32 ではなく、比較的新しい ESP32-C3 をターゲットとして選んだ理由は、Rust コンパイラがデフォルトでサポートしている命令セット(RISC-V)を採用しているためです。そのため、Rust 公式の手順でインストールできるツールチェーンで、組み込みで std な Rust プログラミングを開始できます。

表1 ESP32-C3の概要

CPU	32ビット RISC-V (rv32-ime)注
最高動作周波数	160MHz
RAM	400K バイト
内蔵フラッシュ・メモリ	4M バイト
I/O ピン数	14

注：Atomic 拡張は実装されていません。

● Xtensa は Rust コンパイラがサポートしていない

本原稿を執筆している 2023 年 1 月現在、ESP32 や ESP32-S3 が採用している Xtensa は Rust コンパイラのサポート外の命令セットとなっています。Xtensa をサポートする Rust コンパイラも存在していますが、現在のところ LLVM と Rust コンパイラを fork しており、正式なサポートはなされていない状態です。

まとめると、ESP32 や ESP32-S3 は std はサポートされているものの、Rust コンパイラが正式サポートされていないということです。

そこで、本稿では std と Rust コンパイラの両方がサポートされている RISC-V コアの ESP32-C3 をターゲットとして選びました。ESP32-C3 の概要を表1に示します。

使用するハードウェア

M5Stamp C3U Mate (1,133 円、スイッチサイエンス) は ESP32-C3 を搭載している Wi-Fi/Bluetooth LE モジュールです。ESP32 が 2 コアなのに対して C3 は 1 コアであったり、パシフェラルや I/O が少ないなど、ESP32 と比較すると性能は落ちますが、無線通信機能を利用したちょっとした組み込みシステムを作るには十分なモジュールです。USB Type-C コネクタと比較した写真1を見ていただくとその小ささが伝わるかと思えます。

モジュールにはその他に、リセット・ボタン、RGB LED (NeoPixel) が 1 つ、汎用ボタンが 1 つ、搭載されています(写真2)。USB コネクタは USB Type-C を採