

組み込みRustのライブラリ 便利クレート探偵団



新連載

第1回

マイコンなどリソースに制約があるデバイスを対象としたロギング・フレームワーク

中村 智之

表1 組み込みRustに関する情報が得られるページ

名称	URL
crates.io	https://crates.io/
Rust on Embedded Devices Working Group	https://github.com/rust-embedded
awesome-embedded-rust	https://github.com/rust-embedded/awesome-embedded-rust
Rust Embedded Community	https://github.com/rust-embedded-community

RustはC/C++の代替となりうるシステム・プログラミング言語です。組み込みシステム開発をはじめ、OSやコンパイラの開発にも使えます。

C/C++と同様に直接ハードウェアを制御できる機能を備えつつ、その安全性の高さから、近年注目を集めています。Linuxカーネルでも正式にサポートされたり、機能安全認証の取得を行うなど、ますます私たち組み込みソフトウェア・エンジニアにとって無視できない存在となっています。

そこで本連載では組み込みRustで使える便利なクレート(C言語でいうライブラリ)を紹介します。

組み込みでRustを使ってプログラムを書くに当たり、ESP32シリーズのようにstd(Rustの標準ライブラリ・フルセット)が使える環境ばかりではありません。本連載では、マイコンをターゲットにRustを書くときに使うno_stdな環境でも使えるクレートを中心に紹介します。

何らかの組み込みシステムをRustで作っていく上で、使える機会が多い汎用的なクレートから、特定のアプリケーションを作るためのクレートまで幅広く紹介していきます。

クレートの探し方

クレートをはじめとした組み込みRustに関する情報が得られるページを表1に示します。

● 基本的にcrates.ioを探す

Rustのクレートは、自由にクレートを登録できるcrates.io(図1)に集約されています。

<https://crates.io/>

各自が作成したクレートはcrates.ioに登録できます。登録されているクレートであれば、誰でも自由に自身のプログラムに組み込むことができます。

Rustでクレートを探したければcrates.ioを探すのが最も手軽です。特定クレートに独自の修正を加えたfork版など、まれにGitHubでソースコードが公開されているだけで、crates.ioに登録されていないものもありますが、crates.ioに登録されているものでどうしてもダメなときに調べてみるのがよいでしょう。

● 組み込みで使えるクレートの探し方

crates.ioで組み込みで使えるクレートを検索する際には、no_stdやembeddedをキーワードに加えると効果的です。

使えそうなクレートがあったら最新のリリース・バージョンをチェックして、Cargo.toml(プロジェ

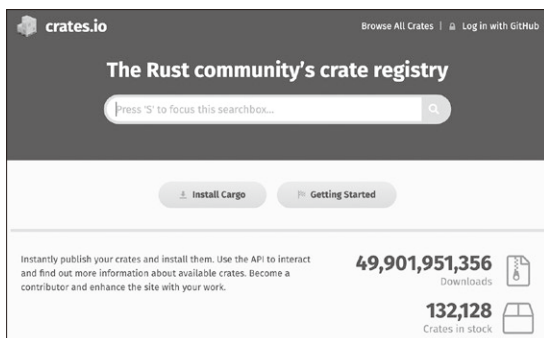


図1 crates.ioのウェブ・ページ(<https://crates.io/>)