



# 組み込みRustのライブラリ 便利クレート探偵団

中林 智之

第3回 マイコンでも動的メモリ確保ができる  
embedded-alloc

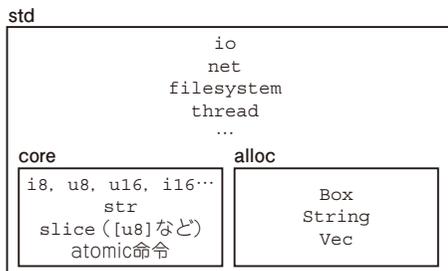


図1 Rust標準ライブラリの構成  
no\_stdである組み込みRustではcoreのみが利用可能

組み込みで使えるプログラミング言語として注目されているRust。そんなRustの組み込み開発で役立つライブラリ(クレート)を紹介します。

今回は組み込みRustでもスマート・ポインタのBoxや可変長配列のVecが使えるようになるメモリ・アロケータembedded-allocについて紹介します。

## Rustのメモリ・アロケータ事情

### ● 組み込みRustではmallocがない

本誌2023年5月号で紹介したM5Stamp C3U Mate (M5Stack)でのstdを使った組み込みRustでは、デフォルトでメモリ・アロケータ(C言語で言うmallocなどの動的メモリ確保)が使えました。そのため、内部でメモリ・アロケータを呼び出しているスマート・ポインタのBoxや可変長配列のVecを特に意識することなく使うことができました。

しかし、マイコンをターゲットとした組み込みRust(no\_stdと呼ぶ。PC向けはstd)では、メモリ・アロケータはデフォルトでは使えません。

図1のようにRustの標準ライブラリはcore, alloc, stdからなる構成になっており、no\_stdな組み込みRustではcoreのみが利用可能です。

動的メモリ確保の観点からすると、これはC言語でmallocのような動的メモリ確保APIが使えない状態と考えてください。

そのため、no\_stdな組み込みRustでは、メモリ・アロケータを使わない範囲でプログラムを作るか<sup>注1</sup>、なんとかメモリ・アロケータが使える状態にしてallocが使えるようにするかを選ぶことになります。

でも安心してください、メモリ・アロケータが使えるようにするのは難しいことはありません。Rustでは、外部クレートを追加することで、C言語のmallocのような動的メモリ確保APIを実装できるようになっています。

今回は、マイコン用のメモリ・アロケータ実装の1つであるembedded-allocをアプリケーションに組み込み、コレクションを使う方法を解説します。

## embedded-allocを使う

embedded-alloc<sup>(1)</sup>はマイコン用のメモリ・アロケータです。embedded-allocを所定のお作法に則って組み込みRustのプログラムに取り入れるだけで、allocにある表1のコレクションが使えるようになります。ある程度、RAMに余裕があるマイコンを使う場合は、確実にプログラム作成が楽になるのでやり方を覚えておきましょう。

allocで他に何が使えるかは参考文献(3)を参照ください。

### ● 環境構築

Rust 1.68以降が必要です。次のコマンドでRustコンパイラのバージョンをチェックして、1.68より古いバージョンであれば、rustup updateでツールチェーンを更新してください。

```
$ rustc --version
rustc 1.76.0 (07dca489a 2024-02-04)
$ rustup update
```

注1: メモリ・アロケータを使わない場合でもheapless<sup>(2)</sup>を使うことで可変長配列Vecのような便利なコレクションを使うことができます。heaplessについては次回の連載で紹介しますので楽しみにお待ちください。