

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8回 ビット・フィールド構造体⇔バイト列の
シリアライズ/デシリアライズを簡単にする deku

Tag	Length	Value
-----	--------	-------

図1 通信プロトコルなどで用いられるデータ形式 TLV
タグ, 長さ, 値をもつデータ形式

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)を本連載では紹介します。

前回から引き続き Rust でビット・レベルの操作を行うのに便利なクレートを紹介します。今回はビット・フィールドのシリアライズ/デシリアライズを簡単にする deku⁽¹⁾ を紹介します。

特徴

● 通信パケットや TLV データを扱うのが得意

シリアライズはデータ構造やオブジェクトを文字列やバイト列に変換する処理を意味します。デシリアライズは文字列やバイト列をデータ構造やオブジェクトに復元する処理のことです。

Rust でのシリアライズ/デシリアライズについては、連載第6回(2024年10月号)で serde⁽²⁾ を紹介しました。serde は JSON や YAML など、さまざまなフォーマットのシリアライズ/デシリアライズが可能なクレートです。

今回紹介する deku では serde でのシリアライズ/デシリアライズのように、ビット・フィールド構造体からバイト列へのシリアライズ、バイト列からビット・フィールド構造体へのデシリアライズを実装できます。deku はバイトおよびビット・レベルのシリアライズ/デシリアライズについて強力な機能が豊富にそろっており、通信パケットや TLV (Tag Length Value) データを扱うのが得意です。

deku は、serde のようにさまざまなフォーマットには対応していない代わりに、ビット・フィールドを含むようなバイナリ・フォーマットのシリアライズ/デシリアライズに特化しています。serde と比較すると、

リスト1 Cargo.toml に deku への依存を追加する

```
[dependencies]
"deku" = "0.18.1"
```

アトリビュートを用いた細かな設定が可能です。

deku の語源が何かは定かではないのですが、公式ページの説明に Deku: Declarative binary reading and writing とあり、Declarative の音から取ってきているのかもしれませんが。

● TLV とは

通信プロトコルなどで用いられるデータ形式です。データの種類(Tag)とデータの長さ(Length)と値(Value)というセットで情報を表現します(図1)。

TLV はさまざまなプロトコルで使われています。スマート・ホームの新しい通信プロトコルの Matter でもさまざまな情報を表現するのに TLV が使われています⁽³⁾。

● no_std でも使えるが alloc が必要

deku はマイコンなどをターゲットにした no_std でも使えます。ただし、ヒープ・アロケータの alloc が必要です。deku を使う場合はヒープ・アロケータをセットで使うようにしてください。

基本的な使い方

プログラム一式は本誌ウェブ・ページからダウンロードできます。

<https://www.cqpub.co.jp/interface/download/contents2025.htm>

まず、Cargo.toml に deku への依存を追加します(リスト1)。

use deku::prelude::*; により deku でよく使うマクロや構造体を一括インポートできます。

#[derive(DekuRead, DekuWrite)] で

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(2024年3月号)

第2回 便利機能満載の汎用デバッグ・ツールキット probe-rs(2024年4月号)