

高い安全性を誇る理由

池田 有

```

出力
出力元の表示(S): ビルド
          ワーニングが発生している
>> Compiling rust_blinky_lib v0.1.0 (C:\Users\2022-034-user\Doc
>> CARGOBUILD(0,0): warning: variable `numGpioArray` should have a
>> C:\Users\user\Documents\SOLID-IDE\Projects#blinky_rust_raw#rust
>>
>> |
>> |   let numGpioArray: Vec<usize> = vec![42, 5, 6];
>> |   |
>> |   |   ^^^^^^^^^^^^^ help: convert the identifier to snake
>> |   |
>> |   = note: `[warn(non_snake_case)]` on by default
>> CARGOBUILD(0,0): warning: 1 warning emitted
>> Finished dev [unoptimized + debuginfo] target(s) in 0.96s
  
```

図1 変数が命名規則に従っていないときに発生するワーニング

Rustは安全性を追求するために、従来言語と比較して厳格な言語仕様を採用しています。ここでは、厳格な点とその理由について解説します。

変数の命名規則

Rustではローカル変数をスネーク・ケース `num_gpio_array` のように命名します。これを例えば、`numGpioArray` などのように先頭が小文字のローワ・キャメル・ケースで付けると、図1のようにビルドの際にワーニングが出ます。

Rustでは、基本的な命名規則として、RFC 0430 という定義があります⁽¹⁾。このRFCは、Rustのコミュニティが管理しているものです。一般的に知られている、ネットワーク系の規則が書かれているRFCとは異なります。

命名規則表を表1に抜粋します。表1を補足すると次のようになります。

アップバ・キャメル・ケース：`NumGpioArray` のように、単語の先頭が大文字
 ローワ・キャメル・ケース：`numGpioArray` のように、先頭以外の単語の先頭が大文字
 スネーク・ケース：`num_gpio_array` のように、アンダスコアで区切る。普通小文字を使う

C言語では、どのシチュエーションで何を使うかは、ある程度決まってはいるものの、運用はプログラマに任されていました。

RustではRFC 0430に従っていない変数名にはワー

表1 命名規則表の抜粋

アイテム	規約
クレート	スネーク・ケース (例: <code>snake_case</code> 。ただし1単語が望ましい)
モジュール	スネーク・ケース
Type	アップバ・キャメル・ケース (例: <code>UpperCamelCase</code>)
トレイト	アップバ・キャメル・ケース
列挙型	アップバ・キャメル・ケース
関数	スネーク・ケース
メソッド	スネーク・ケース
General コンストラクタ	<code>new</code> または <code>with_more_details</code>
Conversion コンストラクタ	<code>from_some_other_type</code>
ローカル変数	スネーク・ケース
静的変数	スクリーミング・スネーク・ケース (例: <code>SCREAMING_SNAKE_CASE</code>)
定数変数	スクリーミング・スネーク・ケース
Typeパラメータ	コンサイス・アップバ・キャメル・ケース、通常は大文字1文字のT
ライフタイム	<code>short, lowercase: 'a</code>

ニングを出すことによって、プログラムする人が変わっても命名規則を遵守でき、統一性のあるコードを書けるという流れになっています。

文字コードはUTF-8

● 処理系における文字コード事情

C言語の場合、文字コードは、基本1バイトで表せるASCIIコードです。日本語を使う場合などの拡張された文字コードはコンパイラによって異なりました。少し前はWindows用 Visual StudioだとだいたいシフトJIS、Linux用だとだいたいUTF-8、というように異なっていました。

Rustの場合、何もしなければ文字コードは常にUTF-8です。

試しに、外部エディタを使ってシフトJISで書いた `lib.rs` を、SOLID-IDEでビルドしてみたところ、