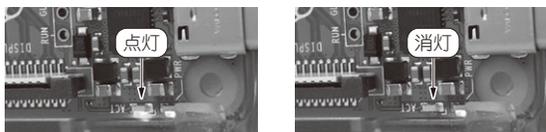


ご購入はこちら

Rustの安全性を Lチカで体験する

池田 有



(a) 点灯時

(b) 消灯時

写真1 ラズベリー・パイでLチカしている

● Lチカを例に Rustの安全性を体験する

RustがCに変わる言語として注目を集めています。C言語と比較して安全性を大幅に向上しつつも、エラー・チェックをコンパイル時にすることで実行時の速度低下を防ぎ、なおかつ組み込み系と言われるシステム・プログラミングも可能な言語として、C言語からの移行先として有力な言語です。

例えば、C言語で組み込みプログラムを作るとき、範囲外のメモリにアクセスしたり、システム・コールの引数が間違っていたりすると、いとも簡単に例外が発生してしまいます。しかしRustなら「そもそもそのような値は設定できない」という素晴らしい利点があります注1。

この利点を理解するために、ここではRustを使ってラズベリー・パイでLチカをしてみましょう(写真1)。ただ単にベアメタル・プログラミングでLチカをするだけでなく、μITRON上で動くアプリケーションとします。μITRONについては高度な機能を使うわけではないので、それほど知識がなくても大丈夫です。

Rustを知らなくても理解できるように、C言語の知識があれば分かるように説明します。必要なのはWindowsマシンとラズベリー・パイと数本のケーブル

注1: 全くできないことはありません。例えば配列を定義し、その範囲を超えたインデックスをアクセスできます。ですが、この場合には範囲を超えたときにすぐにエラー通知処理が動作するため、気が付かずオーバーフローして、最悪リセットに陥るようなことはありません。

注2: Rustの作法に従っていない、と書きましたが、Rustコード上で直接GPIOレジスタを操作する、という意味です。Rustでは基本的にメモリ空間の直接操作は非推奨です。

ルだけです。

● メモリ・アクセス・エラーの起きないプログラムを作っていく

まずはC言語でLチカするプログラムをRustの文法に置き換えただけのプログラムを作成します。Rustの作法に従っていないため注2、C言語同様、メモリ・アクセス・エラーが起これば、

次に同じ動作のプログラムをRustの作法に従って作成します。メモリ・アクセス・エラーが起これないということを実感してもらえると幸いです。

ここ以降、ラズベリー・パイで動作するプログラムを作成するために、無償の組み込み向け統合開発環境SOLID-IDE(京都マイクロコンピュータ)を使用します。一般的に、組み込みでOSを使った開発をするためには、コンパイラ、デバッガ、OS関連のライブラリ、統合開発環境などをインストールし、それらのバージョンを気にする必要がありますが、SOLID-IDEを使うことで開発環境がすぐに整います。これはラズベリー・パイ4Bが持つ4つのCPUコアのうち、2つをLinux、2つをTOPPERSベースのμITRONであるSOLID-OSに割り当て、SOLID-OS用アプリケーションの開発を行うためのIDEです。

この開発環境の構築はAppendixをご覧ください。ここからはSOLID-IDEがインストール済みであると進めます。

ワークスペースの作成

Rustライブラリを作成するためのワークスペースを作成し、本記事内のサンプル・コードをそのまま使い、ビルドして実行してみます。

● 手順1…プロジェクトの作成

C++のプロジェクトを作成します。メニューから[ファイル]-[新規作成]-[プロジェクト]で新規ワークスペースを作成します(図1)。

プロジェクトとして「PTProject」、種別として「ア

