IoTフレームワークで組み込み開発に挑戦

IoT向きモダン言語 Elixir（エリクサー）の研究

第5回 IoTフレームワークNervesでラズパイ周辺デバイスにサクッとつながる

髙瀬 英希

リスト1　hello_nervesというmixプロジェクトを準備

```
# Nervesプロジェクトの作成
$ mix nerves.new hello_nerves
# Nervesプロジェクトのディレクトリに移動
$ cd hello_nerves/
# ターゲットの指定（今回はラズベリー・パイ・ゼロWH）
$ export MIX_TARGET=rpi0
# パッケージのダウンロードと依存性の解消
$ mix deps.get
# ファームウェアのビルド
$ mix firmware
# ホストPCにmicroSDカードを接続し，ファームウェアを書込み
$ mix burn
# ファームウェアのアップデート用のスクリプトを作成
$ mix firmware.gen.script
```

もはやC/C++だけでIoTや組み込みのデバイスを開発するのは大変だと考えている方は多いのではないでしょうか．本連載では，次世代のIoT開発手段として関数型言語ElixirとElixirを使ったIoTフレームワークNervesを紹介します．

組み込み開発の醍醐味

組み込みアプリケーション開発の醍醐味は，さまざまなセンサやアクチュエータなどのデバイスを利用して，実際のデータをやり取りしていくところにあります．

Nervesでは，センサやアクチュエータなどのデバイス・ドライバの開発のために，Elixir Circuits[(1)]というライブラリが提供されています．これには，組み込みの代表的な通信方式であるGPIOやI²C，SPIおよびUARTを制御するためのAPIが一通り用意されています．また，これらの通信方式のドライバだけではなく，LCDディスプレイなどのモジュール単位でサポートされたデバイス・ドライバも提供されています．なお，Elixir Circuitsのその名前の示す通り，汎用のOSが動作するシステム環境，例えばRaspberry Pi OS（旧Raspbian）の動作するラズベリー・パイなどでも利用することができます．

そこで今回は，Elixir Circuitsを使ってNervesからセンサやアクチュエータを制御してみます．筆者はUSB接続によるダイレクト通信が便利なラズベリー・パイ・ゼロWH（ピン・ヘッダ付きのもの）を使用しますが，Nerves対応のボードでしたらどれでも構いません．

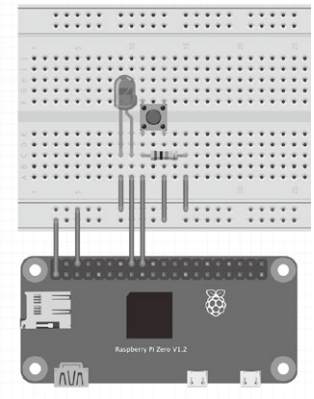

図1　ラズベリー・パイ・ゼロWHとLED，タクト・スイッチを使った回路

前回までに解説した，hello_nervesというmixプロジェクト（**リスト1**）を作成して，そこでビルドしたファームウェアを書き込んだmicroSDカードが準備済みであることとします．また，VintageNetによるネットワーク設定（連載第4回で解説）も準備済みであることとします．前回（2020年9月号）を参照してください．

GPIOを使った制御

● まずはLチカ

やはり組み込みの“Hello，World！”といえば「Lチカ」です．そこでまず，Elixir CircuitsのGPIO駆動用ドライバ・ライブラリであるcircuits_gpioを使用して，LEDの点滅を制御してみます．また，入力デバイスとしてタクト・スイッチも使ってみます．

● 必要なもの

LEDとタクト・スイッチに適切な値の抵抗，それ

第1回　IoT向きプログラミング言語Elixirの世界（2020年6月号）
第2回　ElixirのIoTフレームワークNervesとは（2020年7月号）
第3回　IoTフレームワークNerves&ラズパイでHello World（2020年8月号）