

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 エリクサー Elixirの研究



高瀬 英希

第3回 IoTフレームワーク Nerves&ラズパイでHello World

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な時代です。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^{ナーブス}を紹介します。

IoTフレームワークNervesによる開発

● 手順

いよいよNervesアプリケーションを開発していきましょう。今回はデフォルトで用意されるテンプレートのプロジェクトを使ってラズベリー・パイで、“Hello, World!”を行います(写真1)。

Elixirアプリケーションの開発・実行方法は幾つかありますが(詳細は連載第1回参照)、今回は管理ツールでファームウェアをビルドして実行します。いったんファームウェアが立ち上がったら、コンピュータ・

ボード(本稿では以下IoTボード)上でElixir対話型シェルも実行できます。

● プロジェクトの作成

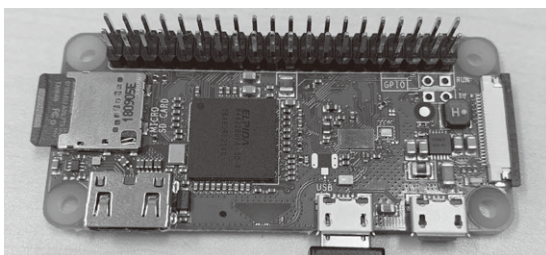
Nerves用のmixプロジェクトを作成します。コマンドはmix nerves.newで、テンプレートのプロジェクトが作成されます。ここでの例として、プロジェクト名はhello_nervesとします(リスト1)。

hello_nerves/以下にさまざまなディレクトリ、ファイルが用意されます。ディレクトリ構造とそれぞれの説明を図1に示します。

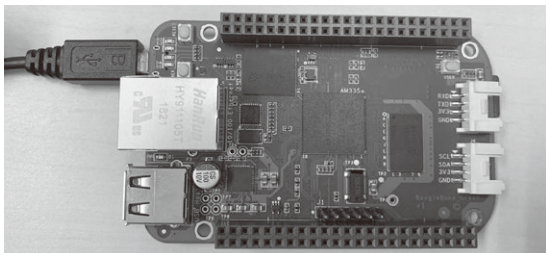
図1の中で特に重要なファイルを解説します。まずmix.exsでは、プロジェクトの情報(アプリケーションの名前やバージョン番号、使用するElixirやNervesのバージョンなど)、起動時に実行するモジュールと関数、依存するパッケージ・ライブラリの指定を記述



(a) ラズベリー・パイ3B+でのNerves起動の様子(HDMIでディスプレイ出力)



(b) ラズベリー・パイ・ゼロWH(ボード中心側のGadget Portに接続することに注意する)



(c) BeagleBone Green

写真1 “Hello, World!” を実行するコンピュータ・ボード