

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4回 IoTフレームワーク Nervesでネットワーク接続

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のでしょうか。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。

前回までに、Elixir/Erlangのインストールと開発環境の構築方法、そしてmixコマンドによるNervesアプリケーションの開発方法を紹介しました。今回はNervesでネットワーク接続を設定する方法と、内部ネットワーク越しにコンピュータ・ボードのファームウェアを更新する方法を紹介します。

● ネットワーク設定のためのライブラリ

VintageNet⁽¹⁾は、ネットワーク設定のためのライブラリです。Nerves対応のボード用のネットワーク設定に特化して開発が進められています(図1)。現時点では、イーサネットやWi-FiとUSB接続によるダイレクト通信のライブラリが用意されています。USB接続は、前回の記事で使用したように、ラズベリー・パイZeroと3A+, または、BeagleBoneファミリのボードで使用できます。ホストPCとボードをUSBで接続して、Micro/Mini-USBポートをアダプタ名がusb0である仮想イーサネットとして使用できる機能です。

VintageNetでは、Nerves対応のボードのネットワークを簡単に設定できます。一度に複数のネットワークに接続し、使用するWi-Fiに優先順位を付けることもできます。拡張性の高い設計となっており、イーサネットやWi-Fiの他にも、ブリッジ接続およびセルラ通信機能のサポートが開発進行中です。将来的にはNerves対応のボード同士を相互接続してメッシュ・ネットワークを構成できるようにする計画もあるそうです。

VintageNetは、Nervesの初期設定ライブラリであるNervesPack⁽²⁾に同梱されています。このライブラリは言わばNervesの標準ライブラリであり、Nerves向けのmixプロジェクトの作成時に、mix.exsのdepsにてデフォルトで記述されています。つまり、VintageNetを使用するための設定は特に必要ありません。

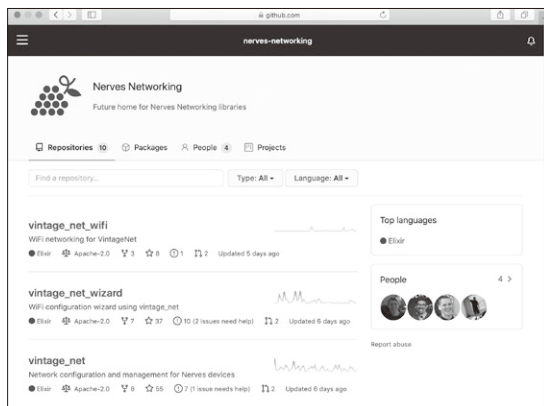


図1 (1) VintageNetの開発が進められているNerves NetworkingのGitHubページ

● ハードウェアの準備

今回は、イーサネットとWi-Fiの両方が備わっているラズベリー・パイ3B+を使います。他のコンピュータ・ボード(本稿では以下、IoTボード)でも、それぞれ同様に設定可能です。ラズベリー・パイZeroやBeagleBoneファミリーでのUSB接続によるダイレクト通信は、前回で解説したように、特に細かい設定をすることなくホストとssh接続できます。

また、前回解説した、mixコマンドによってNervesプロジェクトを作成してビルドし、ファームウェアを書き込んだmicroSDカードを用意します(リスト1)。

それでは、microSDカードをラズベリー・パイ3B+に差し込み、HDMIディスプレイとUSBキーボードを接続して立ち上げます。まだLANケーブルは接続しないでください。

いexターミナルでifconfig/0を実行します(リスト2)。これはLinuxコマンドと同等の情報を返します。ネットワークの状態を確認してみると、まだ何も設定されていないことが分かります。

次に、VintageNetにおいてネットワーク情報を取得するVintageNet.info/0を実行してみます(リスト3)。より詳細な情報が取得できます(写真1)。

VintageNetでは、アプリケーションの実行中また