

# ウェアラブルなLinuxサーバを用いた スマートフォン連携型デバイスの開発

後濱 龍太\*<sup>1)</sup>

## Development of smartphone-linked devices using wearable Linux server

Ryuta Atohama\*<sup>1)</sup>

キーワード: モノのインターネット, ウェアラブル機器, HTML5 Webソケット通信, Intel Edisonボード, Linux OS

Keywords: IoT (Internet of Things), Wearable devices, HTML5 Web sockets, Intel Edison, Linux OS

### 1. はじめに

2020年には250億台の機器がインターネットに接続されサービスを提供すると予測されており, IoT (Internet of Things) 市場の成長が注目されている<sup>(1)</sup>。すでにブレスレットのように装着でき, 心拍数や活動量の計測機能, スマートフォンとの通信機能を備えた製品が販売されている。

こうした製品を支える要素技術にはWi-FiやBluetoothといった無線通信, 加速度センサや心拍センサなどセンサモジュールの制御, 計測したセンサ信号の処理がある。

これらを扱う上で汎用OSのLinuxは都合が良い。2013年頃からのトレンドは, (i)寸法が比較的小型で, Linux OSが動作し, 最初から無線モジュールを内蔵し, かつ1万円前後という調達しやすい価格のCPUボードが安定的に流通していること, (ii)製品の一部あるいは全部が自由に利用できるオープンソースとして公開されていること, (iii)製品ベンダがフォーラムサイトを設けてコミュニティベースの開発を促進する仕組みを整備する点が特徴である。例えばオープンソースなマイコンボードのArduinoシリーズからは, Wi-FiモジュールやGPIO (汎用入出力端子; General Purpose Input/Output), Linux OSを搭載したArduino Yúnが登場し<sup>(2)</sup>, Intelも無線モジュールやGPIOを実装しつつSDカードとほぼ同じサイズのLinuxボードEdisonを販売している<sup>(3)</sup>。

しかし, Linuxには利用可能なツールが膨大に存在するため, どのツールを組み合わせれば所望の機能を実現可能か検討する作業が必要となる。すなわちLinuxを用いた開発では, 要求機能の定義と実装方法の擦り合わせが肝要である。

本稿ではウェアラブル製品の基盤として, 無線接続したスマートフォンからEdisonボードのハードウェアを操作するシステムの構成を検討したので報告する。

### 2. 開発環境

2.1 Edisonボード ボードはウェアラブル用途に適しているEdisonを選択した<sup>(4)</sup>。主な特長は次の6点である。(i)基板寸法がSDカードとほぼ同じで組込みに適している, (ii)電池駆動に適した電源電圧(3.3~4.5V), (iii)Wi-FiおよびBluetoothモジュールを搭載し, かつ技適取得済みなので, 日本国内で利用できる, (iv)オプションのArduino Breakout BoardによりArduinoシリーズ向けの機能拡張ボードも利用できるなど, 試作開発を加速する工夫を備える。(v)Linux資産(シェルスクリプト, システム管理デーモン他)を利用可能, (vi)GPIO制御ライブラリLibmraa<sup>(5)</sup>により, 後述するNode.js<sup>(6)</sup>アプリケーションからハードウェアを制御可能。

本稿ではターゲット基板としてEdisonにArduino Breakout Boardを装着したものをを用いる(図1)。

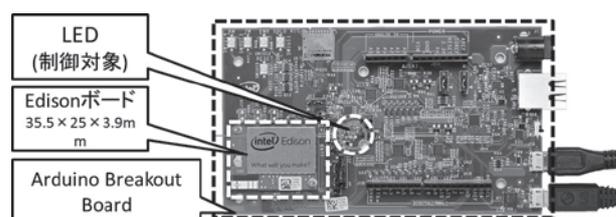


図1. Edisonボード (Arduino Breakout Boardを装着) LED (中央丸棒) の点灯/消灯を制御する

2.2 Node.js Node.jsはJavaScript言語を用いてサーバアプリケーションを構築するためのプラットフォームの1つである<sup>(7)</sup>。JavaScriptの特長であるイベント駆動型でサーバアプリケーションを設計できるので, コードがシンプルになる。また構築するサービスの要件に応じて必要なモジュールを読み込むことができる柔軟性と, モジュールの依存関係を管理するツールを備えている。以上のことから, ウェアラブル製品向けに開発に適している。

2.2 Socket.IO Socket.IO<sup>(8)</sup>はNode.jsでWebソケット

事業名 平成26年度 共同研究

\*<sup>1)</sup>生活技術開発セクター

通信（サーバ・クライアント間の双方向通信<sup>(8)</sup>）を行うモジュールである。Webソケット通信は、低レイテンシや軽量のデータ通信を特長としており、これを使うとメッセージをサーバ・クライアント間で双方向・リアルタイムに通信できる。この実行結果はWebブラウザから確認でき、クライアントのOSに依らず同じ結果を表示できる。

### 3. システムの実装

3.1 構成 2.で述べたツールを組み合わせると、JavaScript言語のみで、クライアントのOSに依らずに同じユーザインタフェース（GUI）を提供しつつハードウェアの制御を実現できる。本稿では、(1) ユーザのスマートフォンに表示されたボタンGUIへの操作を検出して、(2) 無線接続されたEdison基板上のLEDを点灯/消灯するシステムを実装した。機能構成を図2に示す。

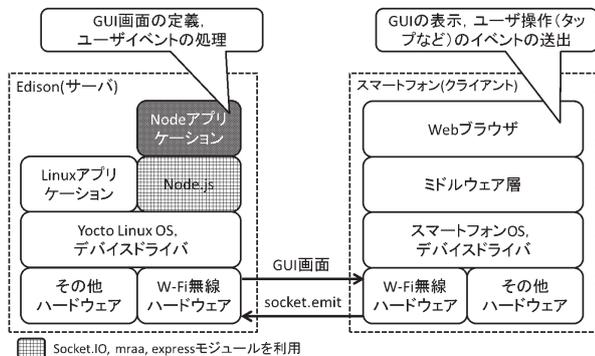


図2. システム構成図

3.2 GUIの実装 ユーザに提示するGUIをHTMLで実装した（図3）。以下では実装したシステムの主要なコードを示す。JavaScript言語の特長であるイベント駆動を用いたので「このボタンが押されたとき実行する関数の定義」といったシンプルな記述ができた。

```

1 <form>
2   <button id="led-on">LEDをONする</button>
3   <button id="led-off">LEDをOFFする</button>
4 </form>
5 <script type="text/javascript" src="js/socket.io-1.2.0.js">
6 </script>
7 <script type="text/javascript" src="js/jquery-1.11.1.min.js">
8 </script>
9 <script>
10  var socket = io();
11  $('#led-on').click(function(){
12    socket.emit('led-is-on?', 1);
13    return false;});
14  $('#led-off').click(function(){
15    socket.emit('led-is-on?', 0);
16    return false;});
17 </script>

```

図3. GUIとして提示するHTML（抜粋）

(2, 3行目) ボタンの定義。(11~16行目) ユーザのボタン押しイベントled-is-on?を送信する関数の定義。「LEDをONする」ボタンのときイベントに値1をセットし、「LEDをOFFする」のときイベントに値0をセットする。

3.3 サーバでの処理 Socket.IOにより取得したボタンイベントの内容に応じて、LEDを点灯/消灯するNode.jsアプリケーションを実装した（図4）。GPIOを制御するライブラリLibmraaはJavaScript言語で利用できるため、Node.jsアプリケーションにそのまま統合できた。

```

1  var express = require('express');
2  var app = express();
3  var http = require('http').Server(app);
4  var io = require('socket.io')(http);
5  var mraa = require('mraa');
6  ledpin= new mraa.Gpio(13);
7  io.on('connection', function(socket){
8    socket.on('led-is-on?', function(state){
9      led-pin.write(state);});
10 })

```

図4. サーバのNode.jsアプリケーション（抜粋）

(1~6行目) 変数の初期化処理を定義する。(8行目) クライアントが送信したイベントled-is-on?を検出したときは、(9行目) Libmraaを用い、LEDが接続されている端子の電圧を制御。変数stateの値が1であればLEDを点灯し、0であれば消灯する。

### 4. まとめ

本稿では小型なLinuxボードEdisonを取り上げ、スマートフォンへのGUIの表示やユーザの操作の検出、ハードウェア制御まで、一つの言語で実装できることを示した。

Linux資産を活用すると、ソフトウェア動作のスケジューリングや動作ログの取得など応用範囲が広がる。今後は小型なLinuxサーバを活用した、ウェアラブルデバイスの高度化と試作開発の迅速化手法について検討する。

（平成27年7月16日受付，平成27年8月4日再受付）

### 文 献

- (1) Gartner, Press Release, <http://www.gartner.com/newsroom/id/2905717>.
- (2) Arduino Yún, <https://www.arduino.cc/en/Main/ArduinoBoardYun>.
- (3) Intel Enables Makers and Wearable Creators, <http://newsroom.intel.com/docs/DOC-5413>.
- (4) Product Brief – Intel® Edison, <http://www.intel.com/support/edison/sb/CS-035277.htm>.
- (5) mraa 0.7.2, <http://iotdk.intel.com/docs/master/mraa/>.
- (6) About Node.js®, <https://nodejs.org/about/>.
- (7) Node.js, <https://nodejs.org/>.
- (8) Web Sockets, <https://www.websocket.org/>.
- (9) Socket.IO, <http://socket.io/>.