

HTML5 Conference 2016

Webエンジニアのためのスマートホームハック Node.js で家電とArduinoを操作

2016年9月3日(土)

有限会社 futomi 代表取締役

株式会社ニューフォリア 取締役 最高技術責任者

羽田野 太巳 (はたの ふとみ)



@futomi

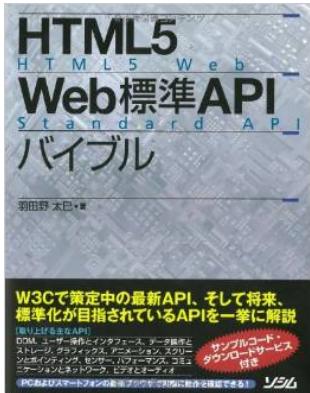


futomi.hatano

自己紹介

は た の ふ と み
羽 田 野 太 巳

有限会社 futomi 代表取締役
株式会社ニューフォリア 取締役 最高技術責任者



API解説本



Windows 8



マークアップ解説本



API解説本



APIチュートリアル本

日経SYSTEMS連載

2015年4月号～2016年3月号：こんなに変わったWeb開発/HTML5の基本テクニック



アーキテクチャからHTML5まで

Web システム 入門

羽田野 太巳 著
日経BP社

“非Web”エンジニア、必読！
誰も教えてくれない
基本から最新動向が
この1冊でわかる
JavaScript REST
Single Page Application

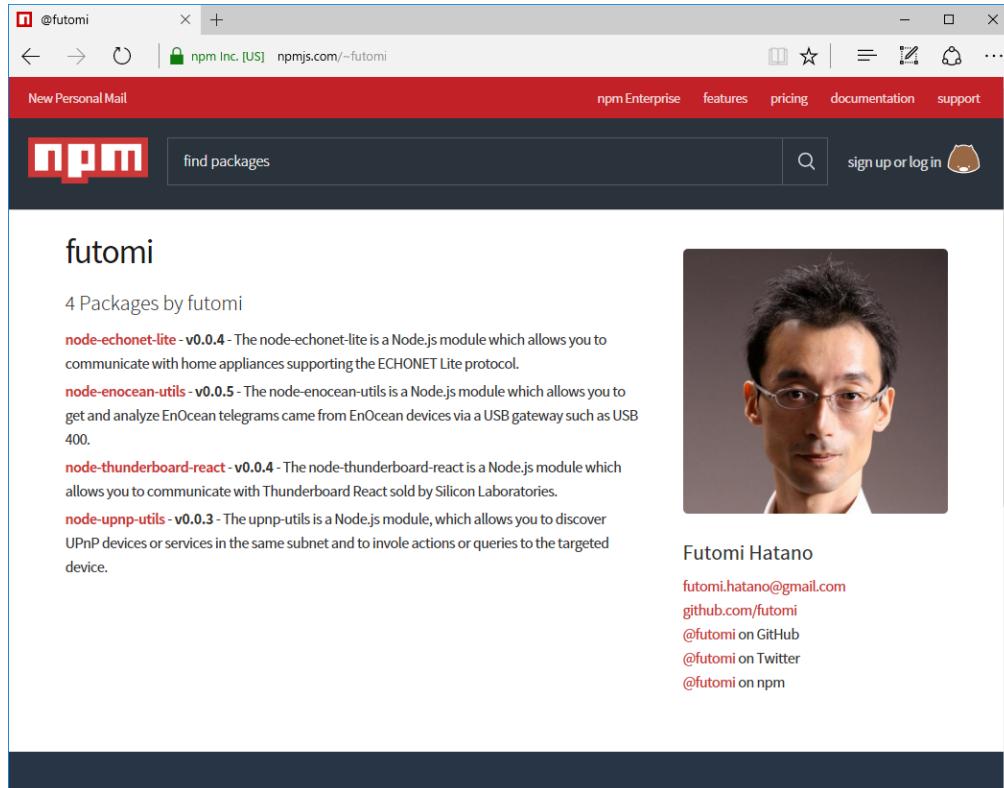
日経SYSTEMSコラム

IoTで火が付くITエンジニア魂：2016年4月号～2017年3月号(予定)



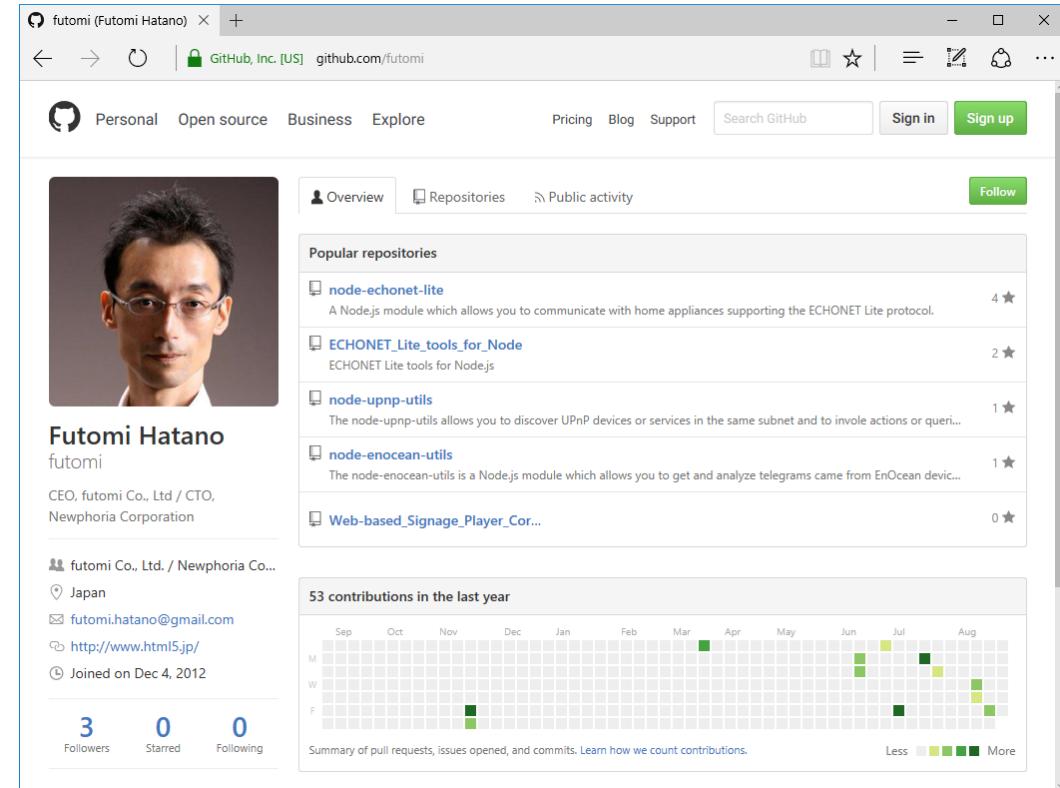
- そろそろIoTに触れてみませんか
- マイコンボードを使って大気圧と気温を取得
- Webブラウザから制御 MIDIでサーボを動かす
- Node.jsで家電を操作 赤外線を利用しよう
- ECHONET Liteで家電をコントロール
- スマホから機器を制御 「デバイスコネクトWebAPI」
- 電池交換が不要に 無線技術 「EnOcean」に注目
- ...

GitHubとnpm



The screenshot shows the npmjs.com user profile for the user 'futomi'. The profile page has a red header with the npm logo and a search bar. Below the header, the user's name 'futomi' is displayed in large bold letters. It says '4 Packages by futomi' and lists four packages: 'node-echonet-lite', 'node-enocean-utils', 'node-thunderboard-react', and 'node-upnp-utils'. Each package entry includes a brief description and a link. To the right of the package list is a portrait photo of a man with glasses. Below the photo is the user's name 'Futomi Hatano' and their contact information: email (futomi.hatano@gmail.com), GitHub URL (github.com/futomi), Twitter handle (@futomi), and npm handle (@futomi). The bottom of the page has a dark footer.

<https://www.npmjs.com/~futomi>



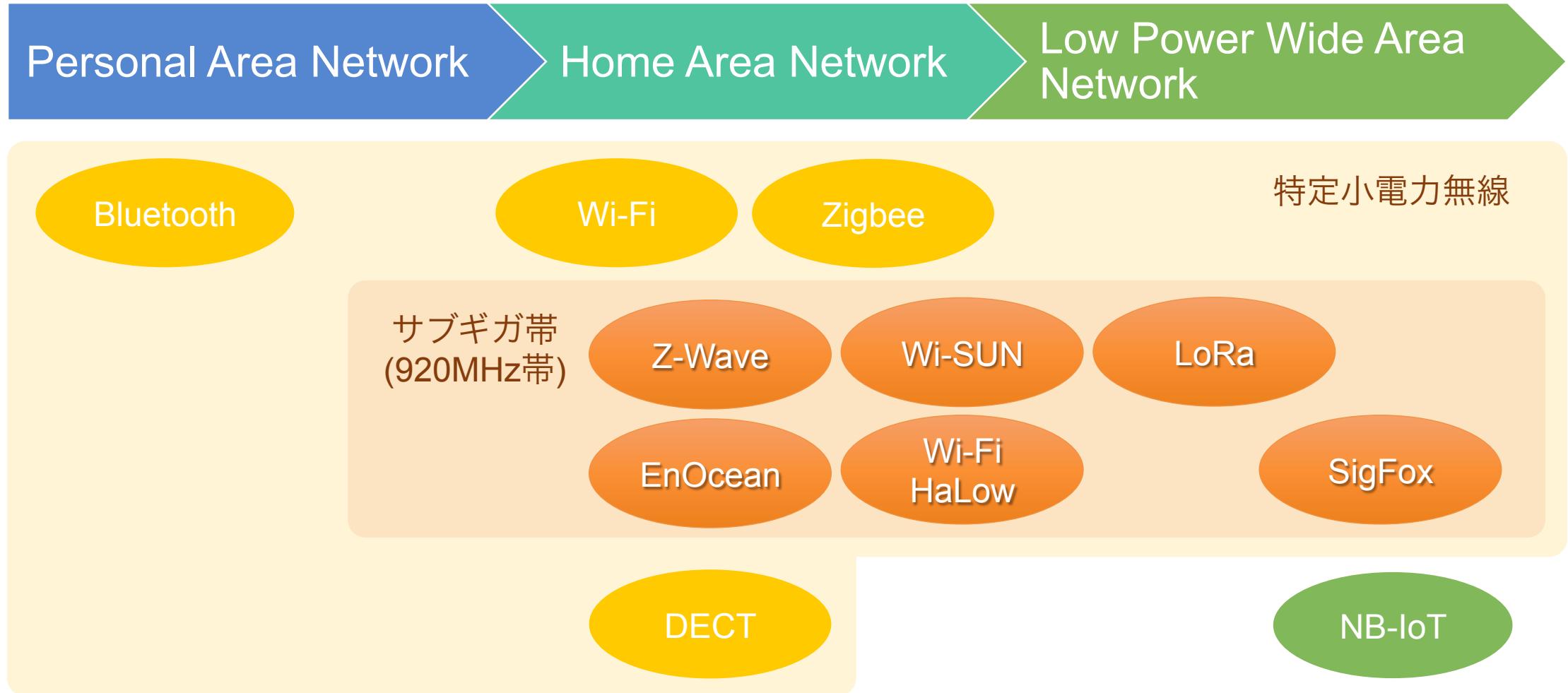
The screenshot shows the GitHub user profile for the user 'futomi'. The profile page has a white header with the GitHub logo and a search bar. Below the header, the user's name 'futomi' is displayed in large bold letters. It says 'Personal' and 'Open source' under the Personal tab. It also shows 'Business' and 'Explore' options. To the right of the tabs is a search bar and a 'Sign in' button. Below the tabs is a portrait photo of a man with glasses. The user's name 'Futomi Hatano' and 'futomi' are displayed below the photo. It also shows 'CEO, futomi Co., Ltd / CTO, Newphoria Corporation'. Below this is a section for 'Popular repositories' which lists five repositories: 'node-echonet-lite', 'ECHONET_Lite_tools_for_Node', 'node-upnp-utils', 'node-enocean-utils', and 'Web-based_Signage_Player_Cor...'. At the bottom is a chart titled '53 contributions in the last year' showing activity from September of the previous year to August. It includes a legend for 'Less' (light green) and 'More' (dark green).

<https://github.com/futomi>

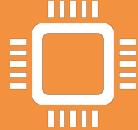
IoTで話題の通信方式

PANからLPWANまで

話題の無線方式



機器操作プロトコル例

センサー	家電/センサー	家電/センサー	AV/NW機器	CCTVカメラ
EnOcean 	Z-Wave 	ECHONET Lite 	UPnP 	ONVIF 
		UDP IP Wi-Fi Ethernet	SSDP UDP IP Wi-Fi Ethernet	WS-Discovery SOAP SOAP HTTP TCP IP Wi-Fi Ethernet
		Wi-SUN	HTTP TCP	SOAP HTTP TCP
				IP Wi-Fi Ethernet

Web屋でも試せる条件

- USBドングルとデバイスがネットで1個から手に入る
 - まとめ買い前提はつらい...
 - 野良ディベロッパーはお呼びでないメーカーが多い...
- 技適が通っている
 - 電波を使う以上、避けては通れない...
 - 海外で普及していても日本では使えないものが多い...
- 通信仕様が無料で一般公開されている
 - 標準化団体によっては会員でないと仕様が公開されていない
 - 非会員には有料で売っている団体もある
- 機器種別ごとのプロファイルが規定されている
 - 低レイヤだけ規定されても上位レイヤが規定されていないと事実上メーカー独自規格と同じで野良ディベロッパーは扱えない



技適マーク

もくじ

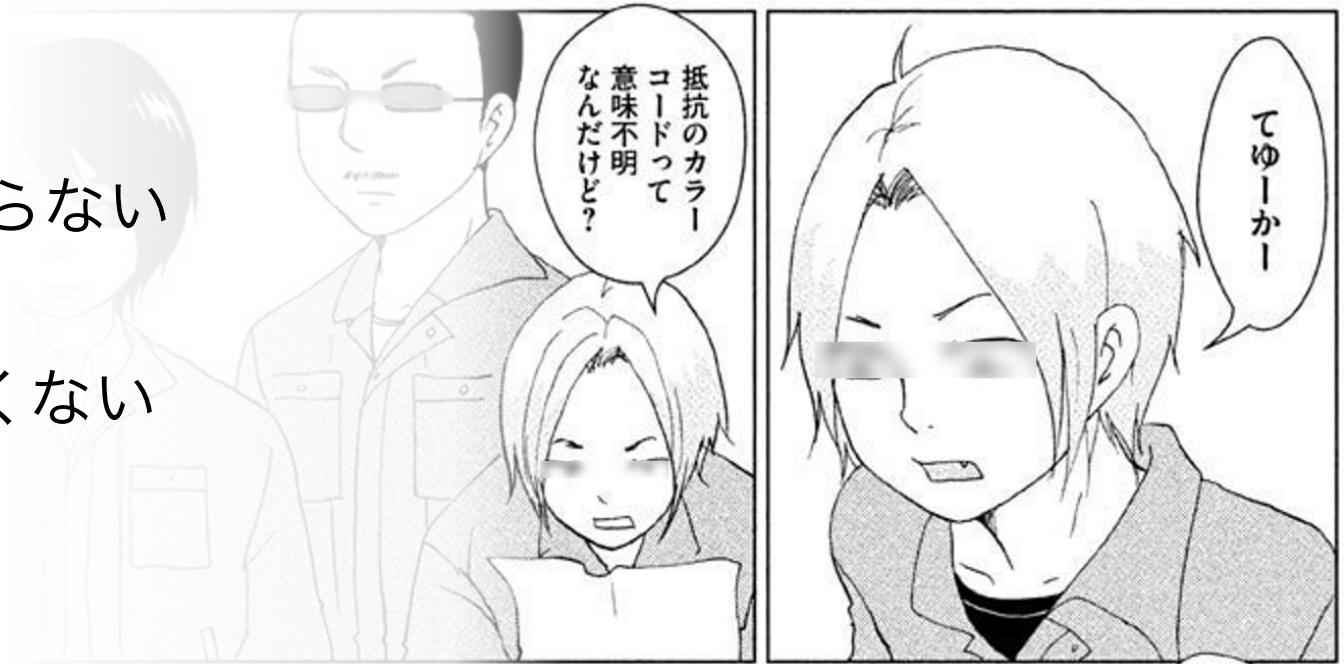
- IoTで話題の通信方式
- Arduino編
- EnOcean編
- ECHONET Lite編
- UPnP編
- まとめ

Arduino編

JSでArduinoにアクセスする方法

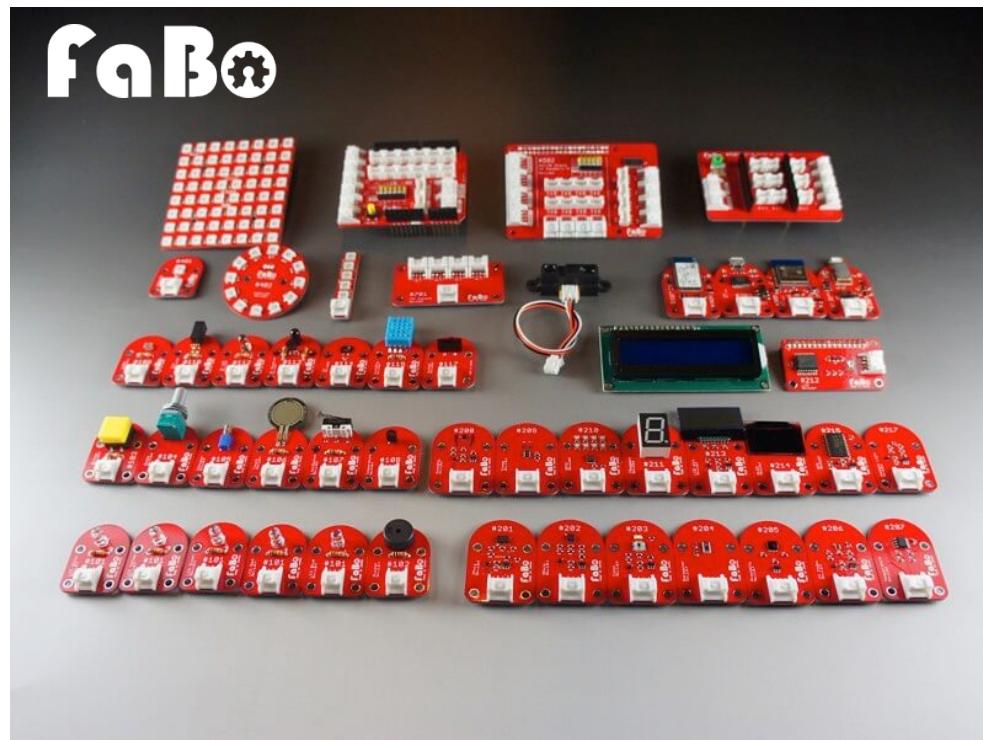
Web屋のArduino

- ・はんだ付けなんてやりたくない
- ・電気回路に興味はない
- ・抵抗のカラーコードなんて分からぬ
- ・ブレッドボードも使いたくない
- ・C言語なんてできる限り使いたくない
- ・やっぱりJavaScriptで



そんなワガママな貴方 (私も) のために...

Arduino Shield Kit



FaBo

<http://fabo.io/>



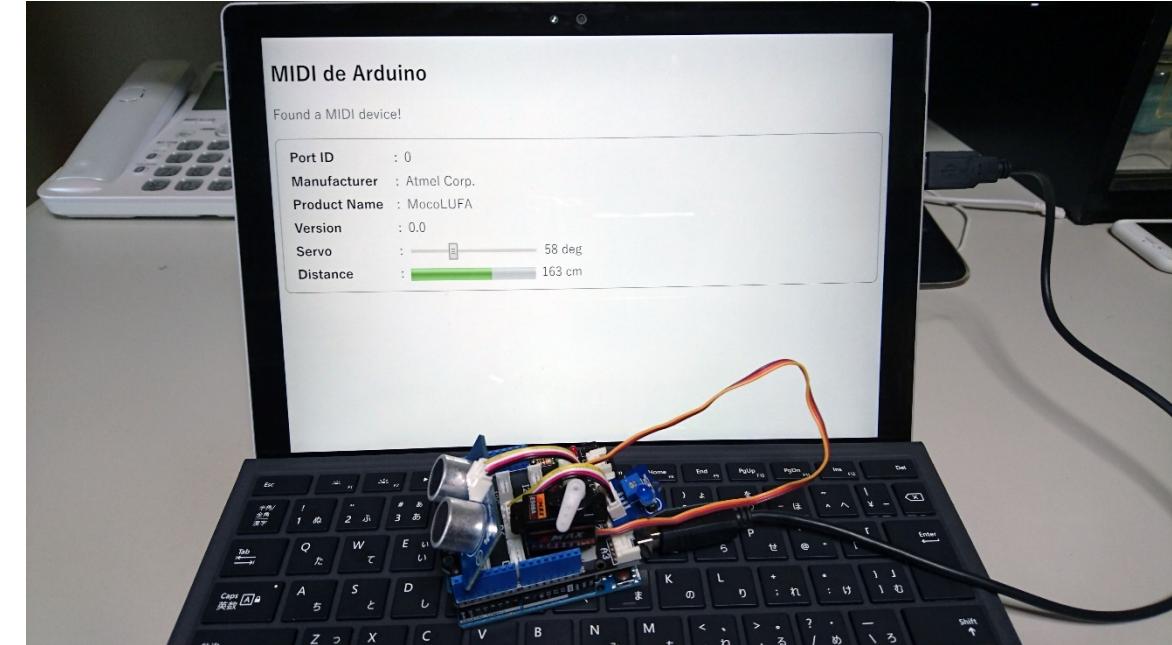
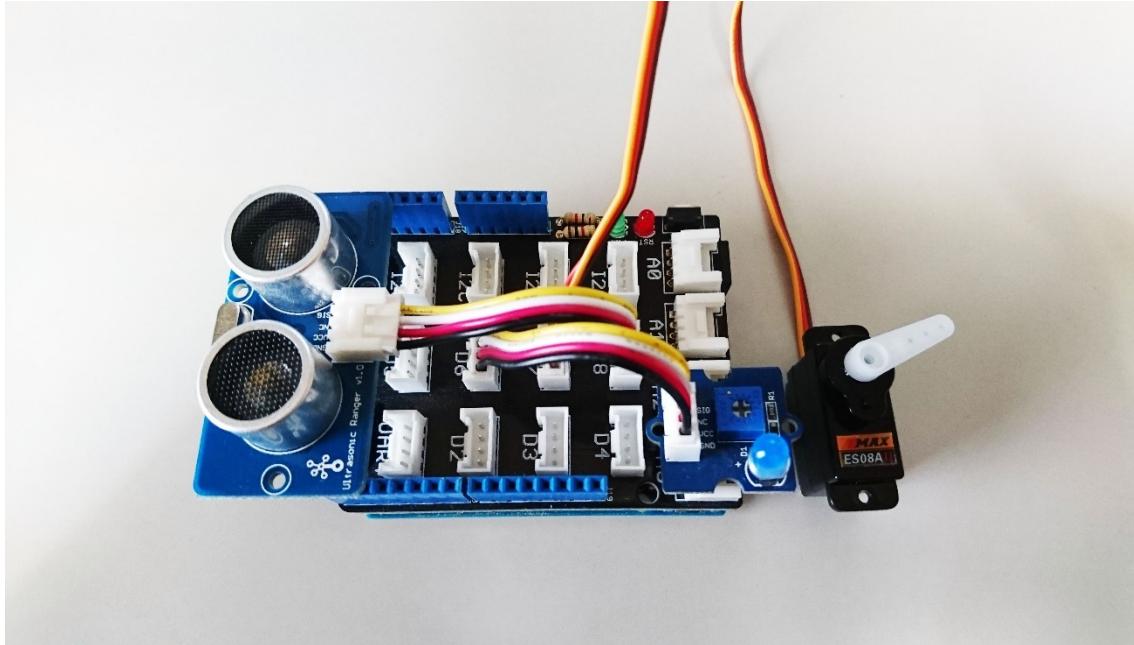
seeed studio - Grove

<https://www.seeedstudio.com/>

Chrome + Web MIDI API + Moco

- Midi Firmware for Arduino Uno (Moco)でArduinoのUSBファームを書き換えて、ArduinoをMIDIデバイスとして認識されるようにする
<http://morecatlab.akiba.coocan.jp/lab/index.php/arduino/midi-firmware-for-arduino-uno-moco/>
- 残念ながらスケッチはC言語で書くが、Arduino MIDI Libraryが便利
<http://playground.arduino.cc/Main/MIDILibrary>
- データはMIDIフォーマットに載せてオレオレフォーマットで送る

実験



- ・ Seeed社のGroveを使って超音波距離センサー, LED, サーボをArduinoに接続
- ・ 距離センサーの結果をリアルタイムに取得して表示
- ・ ブラウザ上でのスライダーでサーボを操作
- ・ サーボ操作のコマンドを受け付けたらLED点灯

ビデオ

<https://youtu.be/d7S5Mqelf08>

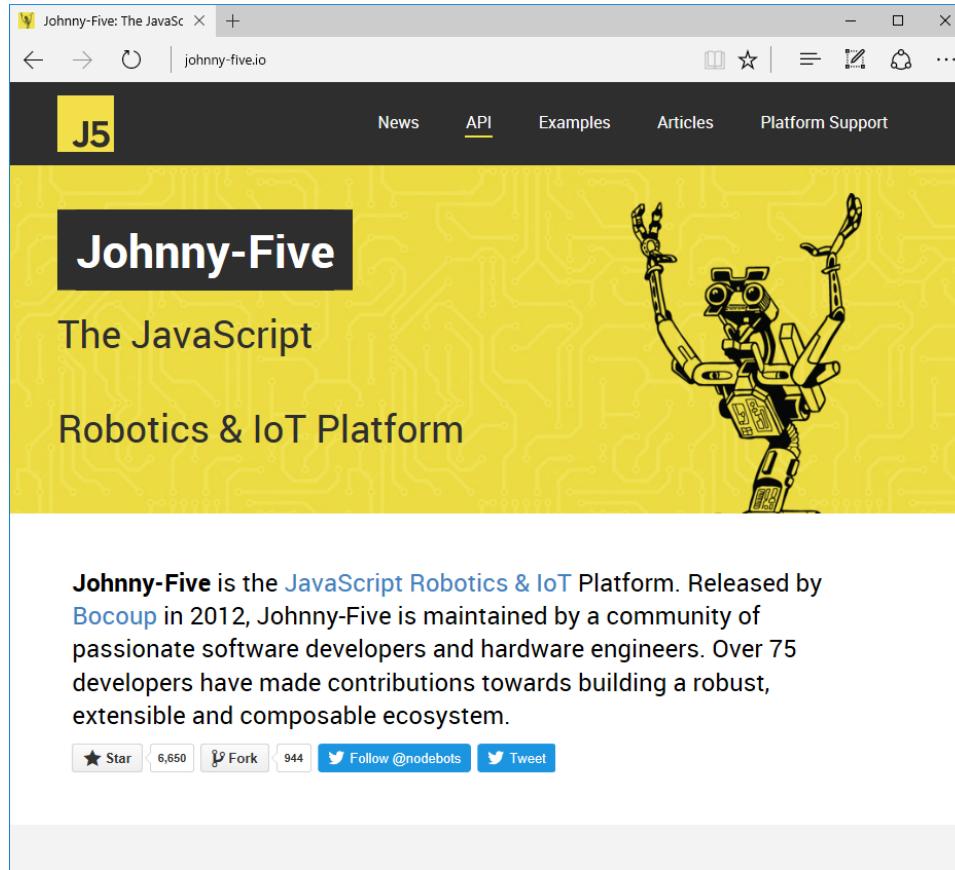
オレオレMIDIフォーマット

ステータスバイト		データバイト1	データバイト2
メッセージ値	チャネル番号		
1		0	0
電子ピアノなら		ノート番号 0~127	ベロシティ 0~127
PC → Arduino		サーボの角度 (deg) データバイト1 + データバイト2 (0~254)	
Arduino → PC		超音波距離センサーが計測した距離 (cm) データバイト1 + データバイト2 (0~180)	

Node.js + Johnny-Five + Firmata

- Johnny-Five
 - USB経由でFirmataを使ってArduinoと通信するNode.jsモジュール
 - さまざまなArduino互換ボードとセンサーに対応
- Firmata
 - PCとマイコンの間でやり取りするためのプロトコル
 - ライブラリはArduinoのIDEに付属
 - ArduinoにFirmataをビルド転送するだけ
 - 実際の通信はJohnny-Fiveに任せプロトコルを知る必要はない

Johnny-Five

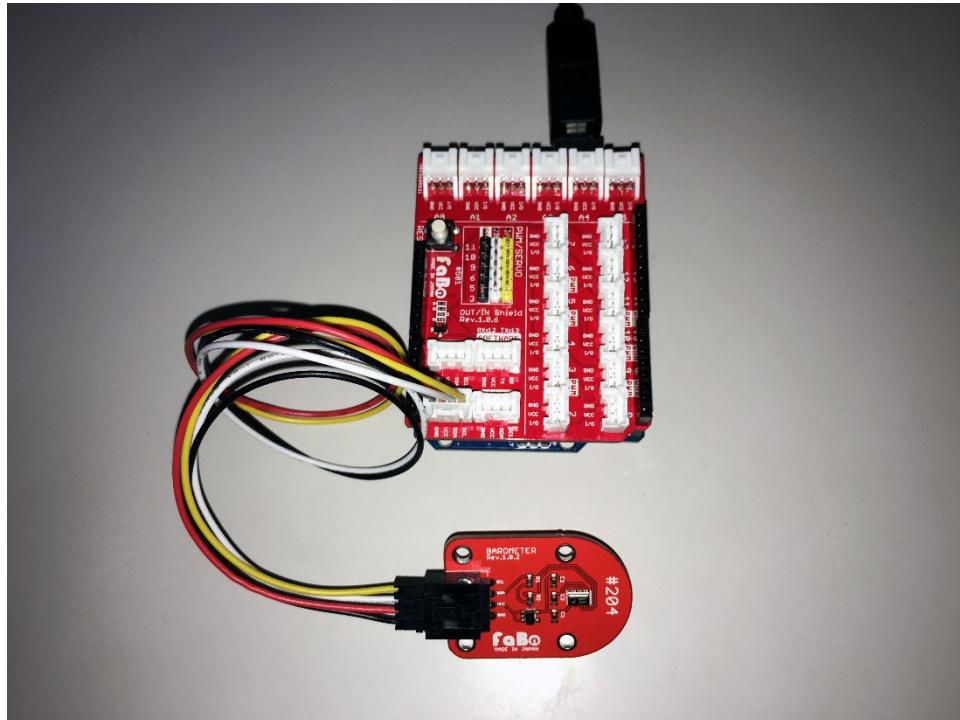


<http://johnny-five.io/>

The screenshot shows the "Multi" component page from the Johnny-Five API documentation. The top navigation bar is identical to the homepage. The main content area features a large "Multi" heading. Below it, there's a section titled "Component Classes" with a list of sensor classes: Accelerometer, Altimeter, Animation, Barometer, Board, Boards, Button, and Compass. To the right, there's a detailed description of the Multi class, explaining it constructs objects for breakout modules containing multiple sensors like thermistors, hygrometers, and altimeters. It also lists supported multi sensor modules: BMP180, BMP280, BME280, and HTU21D, each with links to Adafruit, SparkFun, and Grove components.

<http://johnny-five.io/api/multi/>

実験



```
コマンドプロンプト - node desktop/Johnny-Five-MPL115A2.js
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\太巳>node desktop/Johnny-Five-MPL115A2.js
1457760372058 Device(s) COM3,COM1
1457760372066 Connected COM3
1457760373764 Repl Initialized
>>
-----
Temperature : 22.4 celsius
Pressure    : 1011 hPa
Altitude    : 20.4 meter
-----
Temperature : 22.4 celsius
Pressure    : 1011 hPa
Altitude    : 20.4 meter
-----
```

FaBo社のFaBoを使って気圧センサー (Barometer I2C Brick: MPL115A2) をArduinoに接続
温度と気圧を取得したら高度 (理論値) を計算して表示

Node.jsのコード

```
var five = require("johnny-five");
var board = new five.Board();

board.on("ready", () => {
  var multi = new five.Multi({controller: "MPL115A2"});
  multi.on("change", () => {
    // 温度(摂氏)
    var tmp = this.temperature.celsius;
    // 大気圧(kPaをhPaに変換)
    var prs = this.barometer.pressure * 10;
    // 高度(メートル)：日本の場合
    var alt = ((Math.pow(1013.25/prs, 1/5.257) - 1) * (tmp + 273.15)) / 0.0065;
    // 結果表示
    ...
  });
});
```

EnOcean編

電源要らずのIoT

EnOceanの特徴

- エナジーハーベスティング
 - バッテリーレス無線発信技術
 - 電磁誘導や太陽光発電などで電池要らず
 - 主にビルオートメーションなどの業務用に活用される
 - 近年はスマートホームも視野に
- 周波数はサブギガ帯
 - 日本では928MHz
 - ローム社が日本市場向け無線モジュールを扱っている
- 製品
 - 各種センサー (温度, 湿度, CO₂, ドア開閉など)
 - ロッカースイッチ

製品例



仕様書

- EnOcean GmbH.が一般無料公開
 - <https://www.enocean.com/en/knowledge-base/>
- 仕様書の種類
 - EnOcean Serial Protocol (ESP) 3
 - テレグラムのフォーマット仕様
 - EnOcean Radio Protocol (ERP) 2
 - テレグラムのデータ部の仕様
 - EnOcean Equipment Profiles (EEP) 2.6.5
 - 機器種別ごとの仕様
 - EnOceanアライアンスにて策定

EnOceanのプロトコル体系

EnOcean Serial Protocol 3 (ESP3)

Offset	Size	Field	
0	1	Sync. Byte	
1	2	Header	Data Length
3	1		Optional Length
4	1		Packet Type
5	1	CRC8H	
6	x	Data	
6+x	y	Optional Data	
6+x+y	1	CRC8D	

If the *Packet Type* is 0x0A (RADIO_ERP2),
the *Data* field is specified in ERP2.

EnOcean Radio Protocol 2 (ERP2)

Size	Field
1	Header
0/1	Extended Header
0/1	Extended Telegram type
24/32/48	Originator-ID
0/32	Destination-ID
x	Data_DL
y	Optional Data
1	CRC

Data_DL field is specified in EEP2.6.5.

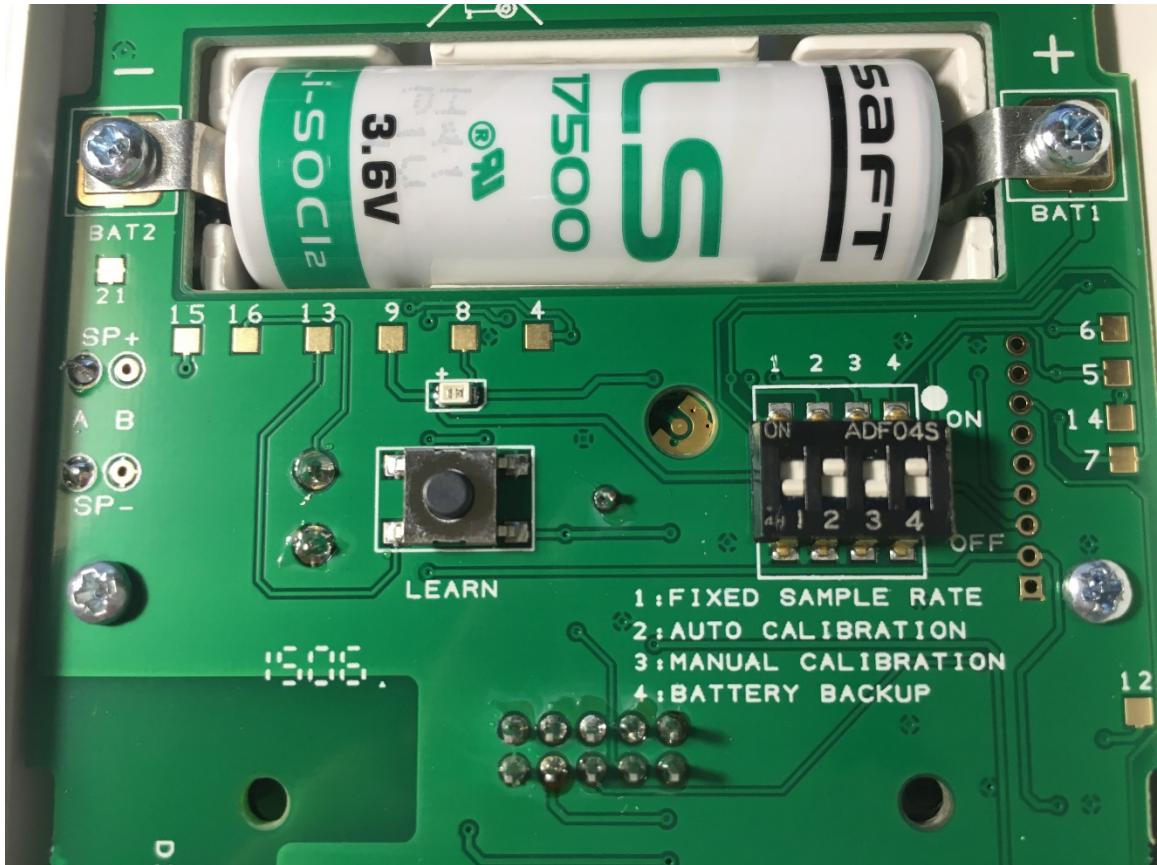
EnOcean Equipment Profiles (EEP)

Size	Field
x	Depends on the type of device (EEP)

EnOcean Equipment Profiles (EEP)

- デバイス種別ごとにEEPが割り振られている
 - A5-02-05: Temperature Sensor Range 0°C to +40°C
 - A5-07-01: Occupancy with Supply voltage monitor
 - A5-09-04: CO2 Sensor
 - F6-02-04: Rocker Switch, 2 Rocker, Light and blind control
- EEPの把握の仕方
 - EnOceanデバイスの説明書に記載されている(はず)
 - 記載されてなければ販売元に聞くしかない
 - 4BSというカテゴリのEEPならLearnボタンが用意されており、それを押すと自分自身のEEPを通知するテレグラムを飛ばしてくれる
 - ただし、製品によっては間違っていたりする...

Learnボタン



node-enocean-utils

- EnOceanを扱うためのNode.jsモジュール
 - <https://github.com/futomi/node-enocean-utils>
- 依存関係
 - Node.js 4.4以上
 - serialportモジュール
- インストール

```
$ npm install node-enocean-utils
```

EnOceanデバイスの登録 (teach-in)

```
var enocean = require('node-enocean-utils');

// EnOceanデバイスの登録
enocean.teach({
  'id' : '00 00 00 2C 86 5C',
  'eep' : 'F6-02-04',
  'name': 'ESM210R Rocker Switch Single'
});
enocean.teach({
  'id' : '00 00 04 01 31 95',
  'eep' : 'A5-02-05',
  'name': 'STM 431J Temperature Sensor'
});
```

EnOceanデバイスのモニター

```
// モニター開始  
enocean.startMonitor({ 'path': 'COM7', 'rate': 57600});  
  
// data-knownイベントのリッスン  
enocean.on('data-known', (telegram) => {  
    var message = telegram['message'];  
    console.log(message['device']['name'] + ': ' + message['desc']);  
});
```

出力例

```
ESM210R Rocker Switch Single: B0 pressed  
ESM210R Rocker Switch Single: B0 released  
ESM210R Rocker Switch Single: BI pressed  
ESM210R Rocker Switch Single: BI released  
STM 431J Temperature Sensor: 28.2°C  
STM 431J Temperature Sensor: 28.4°C
```

ECHONET Lite編

家電とスマートメーターを操作

ECHONET Liteの特徴

- 家電の種別ごとにクラス定義
 - エアコン, 空気清浄機, スマートメーターなど
- 通信の下位層はなんでもアリ
 - TCP, UDP, Bluetooth, Wi-SUNなど
 - セキュリティーも下位層任せ
- 製品
 - エアコン, シーリングライト, 空気清浄機, スマートメーターなど
 - エアコンは別途ネットワークモジュールを購入する必要あり
 - メーカーによってはホームゲートウェイも購入する必要あり

製品例



ダイキン 新うるさら7

ネットワークアダプタ



仕様書

- ECHONETコンソーシアムが一般無料公開
 - <http://echonet.jp/spec/>
- 仕様書の種類
 - ECHONET Lite規格書 Ver.1.12
 - http://echonet.jp/spec_v112_lite/
 - プロトコルの概要および共通部分の仕様
 - APPENDIX ECHONET機器オブジェクト詳細規定 Release H
 - http://echonet.jp/spec_object_rh/
 - 機器種別ごとの仕様

node-echonet-lite

- ECHONET Liteを扱うためのNode.jsモジュール
 - <https://github.com/futomi/node-echonet-lite>
- 依存関係
 - Node.js 4.4以上
 - serialportモジュール (Wi-SUN ルートBを扱う場合)
- インストール

```
$ npm install node-echonet-lite
```

初期化

```
// モジュールのロード
var EchonetLite = require('node-echonet-lite');

// EchonetLiteオブジェクトを生成
var el = new EchonetLite({ 'type': 'lan' });

// EchonetLiteオブジェクトを初期化
el.init((err) => {
  // ここで機器発見
});
```

機器発見

```
el.startDiscovery((err, res) => {
  var device      = res['device'];
  var address     = device['address']; // 機器のIPアドレス
  var eoj         = device['eoj'][0]; // EOJ
  var group_code  = eoj[0]; // グループコード
  var class_code  = eoj[1]; // クラスコード

  // 家庭用エアコンかどうかを判定
  if(group_code === 0x01 && class_code === 0x30) {
    // 機器発見プロセスを停止
    el.stopDiscovery();
    // ここからあれこれエアコン操作
    ...
  });
}
```

エアコンON/OFF

```
var epc = 0x80; // ON/OFFを表すプロパティコード

// ONかOFFかを調べる
el.getPropertyValue(address, ej, epc, (err, res) => {
  // ONだったならtrue, OFFだったならfalse
  var status = res['message']['data']['status'];
  // ONならOFFに, OFFならONにする
  var edt = { 'status': !status }; // EDTオブジェクト
  el.setPropertyValue(address, ej, epc, edt, (err, res) => {
    if(!err) {
      console.log('成功');
    }
  });
});
```

Wi-SUN ルートB用のUSB ドングル

ローム/Jorjin
WSR35A1-00



http://www.rohm.co.jp/web/japan/news-detail?news-title=2014-10-02_news&defaultGroupId=false

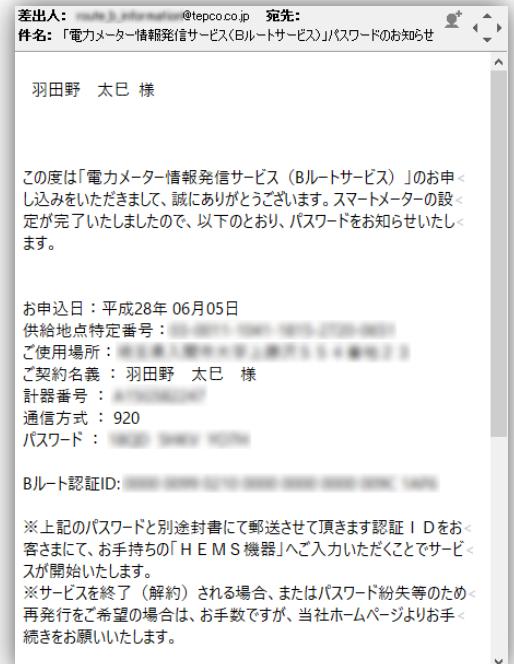
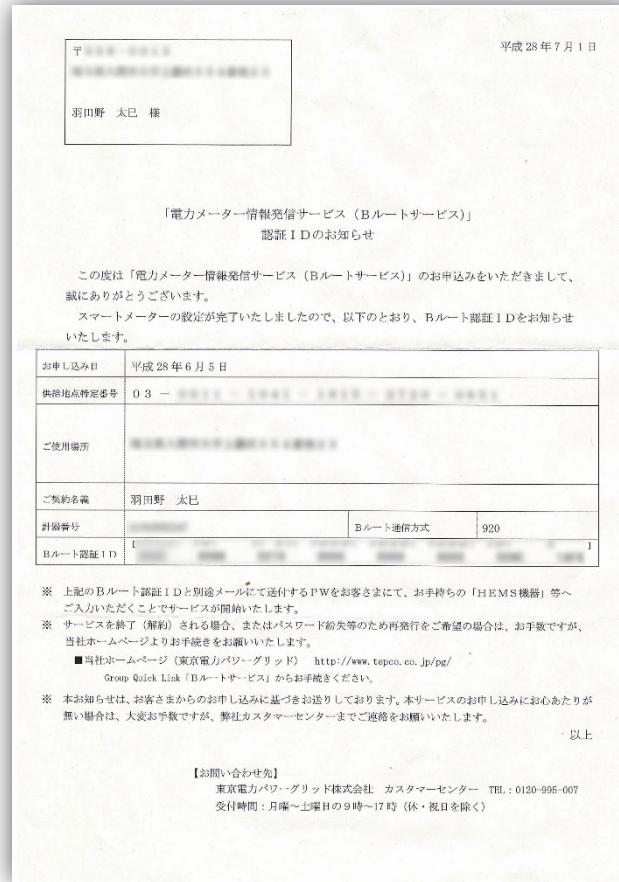
テセラ・テクノロジー
RL7023 Stick-D/DSS



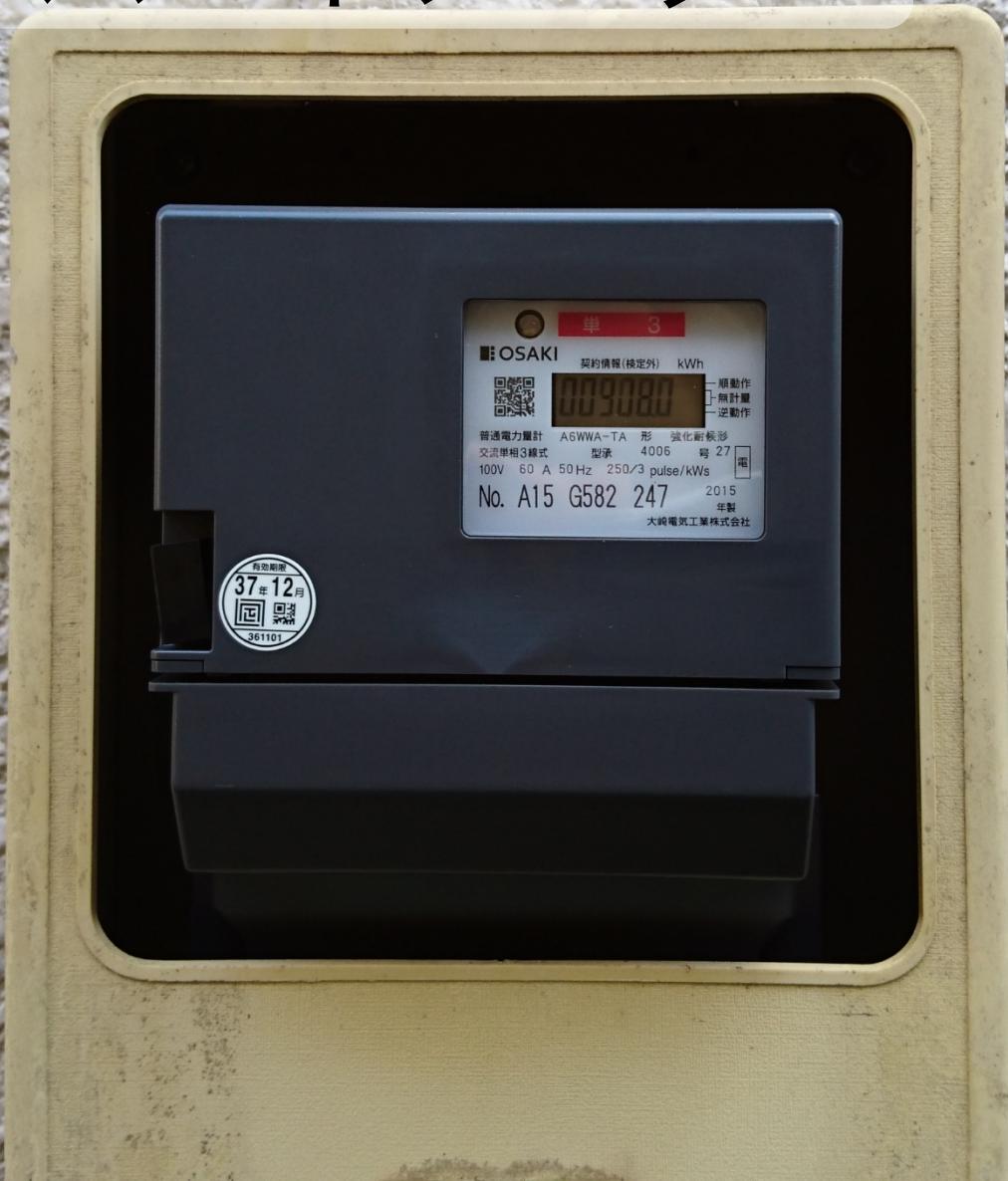
http://www.tessera.co.jp/r17023stick-d_dss.html

Bルートの申し込み(東電の場合)

- 電力メーター情報発信サービス
(Bルートサービス)
<http://www.tepco.co.jp/pg/consignment/liberalization/smartmeter-broute.html>
- 申し込みが終わると後日勝手にメータが交換される(停電なし)
- その後「認証ID」が郵送
- さらにその後メールでパスワードが通知



我が家の中古スマートメーター



Wi-SUN Bルートの実際のやり取り

> SKSENDTO 1 FE80:0000:0000:0000:0000:0000:0000:0000 0E1A 2 000E

1081001A05FF0102880162019E00

ECHONET Lite/パケット部分

EVENT 21 FE80:0000:0000:0000:0000:0000:0000:0000 00

OK

ERXUDP FE80:0000:0000:0000:0000:0000:0000:0000 FE80:0000:0000:0000:0000:0000:0000:0001

0E1A 0E1A 001C64000334809F E100121081001A02880105FF0172019E040381E5ED

> SKSENDTO 1 FE80:0000:0000:0000:0000:0000:0000:0000 0E1A 2 000E

1081001B05FF0102880162019F00

EVENT 21 FE80:0000:0000:0000:0000:0000:0000:0000 00

OK

※ 実際にはテキストとバイナリが混在したコマンドとレスポンスでECHONET Liteパケット部はバイナリ

ビデオ(非公開)

node-echonet-lite Wi-SUN Bルート初期化

```
var EchonetLite = require('node-echonet-lite');
var el = new EchonetLite({
  'lang'    : 'ja',
  'type'    : 'wisunb',
  'adapter' : 'bp35a1',
  'path'    : 'COM8',
  'id'      : '0123456789ABCDEF0123456789ABCDEF',
  'pass'    : 'ABCDEFGHIJKLM',
  'baud'    : 115200
});
```

- ・以降はUDPを扱う場合と同じ
- ・ただし、機器発見は時間がかかる（我が家環境では20秒ほど）
- ・基本的にレスポンスは遅い

UPnP編

テレビなどのAV機器にアクセス

UPnPの特徴

- 機器発見と録画映像再生制御など
 - SSDP, M-Search
 - HTTP POSTでSOAPをリクエスト
- 対応機器
 - ほとんどのテレビが対応
 - ただしUPnPで規定されていない操作はメーカー独自
 - パケットキャプチャーすればある程度は分かる(かも)

node-upnp-utils

- UPnPを扱うためのNode.jsモジュール
 - <https://github.com/futomi/node-upnp-utils>
- 依存関係
 - Node.js 4.4以上
 - xml2jsモジュール
- インストール

```
$ npm install xml2js
$ npm install node-upnp-utils
```

機器発見

```
var upnp = require('node-upnp-utils');

// addedイベントのリスナー
upnp.on('added', (device) => {
    // 欲しいデバイスが見つかれば発見処理を停止
    upnp.startDiscovery();
    // あとはお好きなように...
    ...
});

// 機器発見開始
upnp.startDiscovery();
```

アクション送信

```
var upnp = require('node-upnp-utils');

var soap = '';
soap += '<?xml version="1.0" encoding="utf-8"?>';
soap += '<s:Envelope ...>...</s:Envelope>';

var params = {
  'url' : 'http://192.168.10.4:20003/MediaServer/ContentDirectory/Control',
  'soap': soap
};

upnp.invokeAction(params, (err, obj, xml, res) => {
  ...
});
```

ビデオ(非公開)

まとめ

Web屋でもIoT

Web屋でもIoT

- Webディベロッパーでも関わられる分野はある
 - Arduino, Wi-SUN, EnOcean, etc..
 - IoTはモノだけでなく見える化も重要、そこはWeb屋の得意分野
- ハードは専門家に任せよう
 - ドットメイクのブームもあるがそんな簡単じゃない
 - ハルロック読みましょう



ご清聴ありがとうございました

有限会社 futomi 代表取締役

株式会社ニューフォリア 取締役 最高技術責任者

羽田野 太巳（はたの ふとみ）



@futomi



futomi.hatano