IoTエンジニアへの第一歩! 若手エンジニア向けスキルアップイベントシリーズ

XIDT

メイカーズ・チャレンジ

ハンズオン講習会テキスト

2018.2.24

inte





「Web×IoT メイカーズチャレンジ 2017 in横須賀」では、 小型ボードコンピューター Raspberry Pi とセンサーキットを使って、 JavaScriptによるハードウェア制御をハンズオン形式で学びます

本テキストの対象・内容レベル

Webを触ったり、ちょっとしたプログラミングなら やったことがあるけれど、Raspberry Pi は触った ことがない…… という方

組み込み開発の知識はあるけれど、 Web技術での制御は未経験・・・・・という方

※本テキストに記載されているブランド名、会社名、製品名等は、それぞれ各社の登録商標または商標です。

目次

1. 使用機材	P4		6. 実習① LEDを光 らせてみる P11
2. システム構成	Ρ7		実習② 超音波セ 7. ンサーを使ってみ P12 る
3. センサーを使った 3. 簡単なアイデア例	P8		8. 実習③ 測定結果 8. を表示してみる P13
4. Raspberry Pi の 初期セットアップ	P9		9. 付録 P14
5. Raspberry Pi の リモート操作	P10		
		T	後半は実機演習です。





使用機材 | Raspberry Pi

Raspberry Pi (ラズベリーパイ)とは?

通称「ラズパイ」。超小型、低価格でありながら、高い可能性を秘めたLinux PC。 現行モデル(Raspberry Pi 3 Model B)は、HDMI出力、USBポート、LAN、 Wi-Fi、Bluetooth、microSDスロットなどを搭載。



使用機材 GrovePi+ スターターキット

厳選された12種類のセンサー類を用意。はんだ付け不要で簡単にボードへ接続可。



GrovePi+ボード

使用機材 | GrovePi+ ボード

GrovePi+は、Raspberry Pi の上に取り付けてセンサー類を使えるようにする 拡張ボード。デジタルポート、アナログポート、I2Cポートを装備。





使用する機材・ソフトウェアのシステム構成は以下の通り。 各種センサーからNode.jsに対応したライブラリまで、主催者側で用意。



センサーを使った簡単なアイデア例



Raspberry Pi の初期セットアップ

Raspberry Pi 3 スターターキットに付属のmicroSDには、OS(Raspbian)や Node.js、Node.jsに対応したライブラリが予めインストール済み。

Raspberry Pi のmicroSDスロットにmicroSDを差し込み、電源を入れると すぐに使用できる状態になっています。

参考

参考までに、自分で初期セットアップする場合は以下参照

- 1. 公式サイトからOSイメージファイルをダウンロード <u>https://www.raspberrypi.org/downloads/raspbian/</u>
- 2. 作業用PCでOSイメージファイルをmicroSDカードへコピー Windows, Mac共通で使える「Etcher」というアプリを使うと便利 <u>https://etcher.io/</u>
- 3. Raspberry Pi のmicroSDスロットにmicroSDを差し込み、 HDMIのモニター・USBキーボード・マウスを繋いで電源を入れる

デスクトップ画面が表示されたら 起動完了!





Raspberry Pi のリモート操作

PCから Raspberry Pi をリモート操作する。

PCと Raspberry Pi が同じネットワークに参加していれば、 HDMIのモニター・USBキーボード・マウスがなくても、PCから操作可。

Raspberry Pi 側

1. Raspberry Pi でVNCを利用可能に設定

192.168.0.45 (raspberrypi) - VNC Viewer		Ra	spberry Pi の設定		_ = ×
 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	システム	インターフェイス	パフォーマンス	ローカライゼーション	
1 7772 ·	カメラ:		○ 有効	• 無効	
× 7-4 ,	SSH:		○ 有効	⊙ 無効	
 アクセサリ ・ 塗 システムツール ・ 	VNC:		• 有効	○ 無効	
	SPI:		○ 有効	• 無効	
Add / Remove Software	120:		○ 有効	• 無効	
Run	シリアル:		○ 有効	⊙ 無効	
Shutdown	1-Wire:		〇 有効	④ 無効	
Main Menu Editor	リモートGPIO:		○ 有効	• 無効	
 ・・ボードとマウス ・・ボードとマウス ・・ボードとマウス ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・				キャンセル(C)	OK(0)

PC 側

3. PCにVNC Viewerのアプリをインストール

VNC Viewerは、Windows, Mac共通で使える https://www.realvnc.com/en/connect/download/viewer/



2. Raspberry Pi のホスト名を調べる

	Ra	spberry Pi の設定	-	
システム	インターフェイス	パフォーマンス	ローカライゼーション	
パスワード:			パスワードを変更(P)	
ホスト名:		raspberryp	și	
ブート:		⊙ デスクトップ	O CLI	
自動ログイン			☑ 現在のユーザとしてログイン	す
ネットワークプート:			□ ネットワークを待つ	
スプラッシュ画面		⊙ 有効	○ 無効	
解像度			解像度を設定(R)	
オーバースキャン		● 有効	○ 無効	
			(キャンセル(C)) OK(O)	

4. PCでVNC Viewerを起動、 Raspberry Pi のホスト名を入力して接続



実習① | LEDを光らせてみる

Raspberry Pi をJavaScriptで制御してLEDを光らせてみよう。

1. GrovePi+ボードのD4にLEDを挿す



2. ターミナル(コマンドプロンプト) から以下コマンドを実行

node led.js

【コマンドの意味】 Node.jsで「led.js」というJavaScript言語 のプログラムを実行。「led.js」の中身は右 ■

1秒間隔でLEDが点滅したら 成功!

// LED点滅

var GrovePi = require('node-grovepi').GrovePi; var Board = GrovePi.board; var LedDigital = GrovePi.sensors.base.Digital; var blink = false; var board = new Board({ debug: true, onError: function(err){ console.log('ERROR'); console.log(err) }, onInit: function(res){ if(res){ var ledDigital = new LedDigital(4); // D4ポートを指定 setInterval(function(){ ledDigital.write(blink); // LEDを点滅 if(blink){ blink = false; }else{ blink = true; },1000); // 1000ミリ秒(=1秒)のインターバル }); board.init();

実習②|超音波センサーを使ってみる

Sy オフィス

ゲーム

↓ 電子工学

Help

盖 設定

Run...

Thutdown.

超音波センサーを使って距離を測定してみよう。 Node-RED を使うと、GUIで直感的・簡単にプログラミングできる。

1. GrovePi+ボードのD3に 超音波センサーを挿す



2. Node-REDを記動

Raspberrypi.local (ras	pberrypi) - VNC Viewer	
۱ 🔁 🕒 🌘 🍯	× 🚯	URLは、
🚺 プログラミング	> 🥯 Arduino IDE	総告に
🤦 オフィス	> 🐔 BlueJ Java IDE	酸ぐ塩の
() インターネット	> 🥹 Geany	を作成。
26 J-4	> 嘴 Greenfoot Java IDE	CITIZA
📢 アクセサリ	> 🜞 Mathematica	
👰 システムツール	> 🔁 Node-RCC	ファイル(F) 編集(E) タブ(T) ヘルプ(H)
┿ 電子工学	> 🥐 Python 2 (IDLE)	Once Node-RED has started, point a browser at http://102.168.0.2 Dn P1 Node-RED works better with the Firefox or Chrome browser
Help	, 🥐 Python 3 (IDLE)	Use node-red-stop to stop Wode-RED a Use node-red-start to start Node-RED a use node-red-start to use the recent Use sudo systemati enable nodered.service to autostart Node-R Use sudo systemati disable modered.service to disable autostar
🧱 設定	Scratch	To find more nodes and example flows - go to http://flows.noderes starting as a systemd service. Started Woole:RD graphical event wiring tool 17 Jan 11(5):40 - { [info] welcome to Node-RD
Run	Sense HAT Emulator	17 Jan 11:53:40 [info] Node-RED version: v0.17.5 17 Jan 11:53:40 [info] Node-RED version: v0.17.5 17 Jan 11:53:40 [info] Node-Jo Version: v0.17.4 17 Jan 11:53:40 [info] Loading palette nodes 17 Jan 11:53:46 [info] Casabbard version 2.7.0 started at /ui
Shutdown	(II))) Sonic Pi	<pre>Info orower.loard orower is initing I7 Jan 11:53:50 - [info] Stitings file : /home/pi/.node-red/set: 17 Jan 11:53:50 - [info] User directory : /home/pi/.node-red/flow rend</pre>
No. of Concession, Name	Th Thonny (Simple Mode)	17 Jan 11:53:50 - [info] Starting flows 17 Jan 11:53:50 - [info] Started flows 17 Jan 11:53:50 - [info] Server now running at http://127.0.0.1:
and the second s	Th Thonny Python IDE	
	K Wolfram	server now running at htt と出力されたら Node



. . .

ダッシュボ

▼全てのフロー 首

実習③|測定結果を表示してみる

超音波センサーを使って測定した距離を Web画面に表示してみよう。 Node-RED のダッシュボードを使うと、グラフィカルにリアルタイム描画できる。



付録|初期セットアップの詳細①



付録|初期セットアップの詳細②

Raspberry Pi 初期セットアップの詳細(つづき) 1. OS(Raspbian) のアップデート

sudo apt-get update sudo apt-get upgrade

2. Node.jsとnpmの最新版インストール

sudo apt-get install -y nodejs npm sudo npm cache clean sudo npm install n -g sudo n stable

3. GrovePi+ソフトウェアのインストール

cd /home/pi/Desktop sudo git clone https://github.com/DexterInd/GrovePi.git cd /home/pi/Desktop/GrovePi/Script sudo chmod +x install.sh sudo ./install.sh

4. GrovePi+のNode.jsに対応したライブラリのインストール

sudo npm install node-grovepi

5. Node-REDの最新版とダッシュボードのインストール

update-nodejs-and-nodered sudo npm install -g node-red-dashboard

6. Node-REDのGrovePi+ノードのインストール

sudo npm install -g node-red-grovepi-nodes

7. 日本語入力メソッド fcitx-mozcのインストール

sudo apt-get install fcitx-mozc





Node-RED に登録されている GrovePi+ 12種類のセンサー類



付録|センサーのサンプルコード①

1. 超音波センサー(ultrasonic.js)

```
// 超音波センサー
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var UltrasonicDigitalSensor = GrovePi.sensors.UltrasonicDigital;
var board = new Board({
 debug: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var ultrasonicSensor = new UltrasonicDigitalSensor(3); //
D3ポートを指定
   ultrasonicSensor.stream(1000, // 1000ミリ秒(=1秒)のイン
ターバル
   function(res){
    console.log(res) // 値を出力
   });
  }
 }
});
board.init();
```

```
2. 回転角センサー(rotary.js)
```

```
// 回転角センサー
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var RotaryAnalogSensor = GrovePi.sensors.RotaryAnalog;
var board = new Board({
 debug: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var rotarySensor = new RotaryAnalogSensor(1); // A1\#-h
を指定
   rotarySensor.start();
   rotarySensor.on('data', function(res){
    console.log(res) // 値を出力
   });
});
board.init();
```

付録 | センサーのサンプルコード②

3. 光センサー(light.js)

```
// 光センサー
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var LightAnalogSensor = GrovePi.sensors.LightAnalog;
var board = new Board({
 debug: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var lightSensor = new LightAnalogSensor(0); // A0ポートを
指定
   lightSensor.stream(1000, // 1000ミリ秒(=1秒)のインターバル
   function(res){
    console.log(res) // 値を出力
   });
  }
 }
});
board.init();
```

4. ボタン(button.js)

```
// ボタン
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var DigitalButtonSensor = GrovePi.sensors.DigitalButton;
var board = new Board({
 debug: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var buttonSensor = new DigitalButtonSensor(7); // D7\pi-
トを指定
   buttonSensor.on('down', function(res){ // ボタンが押された時
    console.log(res) // 値を出力
   });
   buttonSensor.watch();
});
```

```
board.init();
```

付録 | センサーのサンプルコード③

5. 音センサー(sound.js)

// 音センサー

```
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var SoundAnalogSensor = GrovePi.sensors.LoudnessAnalog;
var board = new Board({
 debug: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var soundSensor = new SoundAnalogSensor(2); // A2ポート
を指定
   soundSensor.stream(1000, // 1000ミリ秒(=1秒)のインターバ
ル
   function(res){
    console.log(res) // 値を出力
  });
  }
 }
});
board.init();
```

6. 温度・湿度センサー(dht.js)

```
// 温度・湿度センサー
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var DHTDigitalSensor = GrovePi.sensors.DHTDigital;
var board = new Board({
debug: true,
onError: function(err){
  console.log('ERROR');
  console.log(err)
},
onInit: function(res){
  if(res){
   var dhtSensor = new DHTDigitalSensor(2); // D2ポートを指定
   dhtSensor.on('change', function(res){ // 値が変化した時
    console.log(res) // 値を出力
   });
   dhtSensor.watch(1000); // 1000ミリ秒(=1秒)のインターバル
 }
});
```

```
board.init();
```

付録 | センサーのサンプルコード④

7. リレー(relay.js)

```
// リレー
var GrovePi = require('node-grovepi').GrovePi;
var Board = GrovePi.board;
var RelayDigital = GrovePi.sensors.base.Digital;
var relay = false;
var board = new Board({
 debug: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var relayDigital = new RelayDigital(6); // D6ポートを指定
   setInterval(function(){
    relayDigital.write(relay); // リレースイッチをオン・オフ
     if(relav){
      relav = false;
     }else{
      relay = true;
     3
   },1000); // 1000ミリ秒(=1秒)のインターバル
  }
 }
});
board.init();
```

8. ブザー(buzzer.js)

```
// ブザー
```

var GrovePi = require('node-grovepi').GrovePi; var Board = GrovePi.board; var BuzzerDigital = GrovePi.sensors.base.Digital; var buzzer = false; var board = new Board({ debug: true, onError: function(err){ console.log('ERROR'); console.log(err) }, onInit: function(res){ if(res){ var buzzerDigital = new BuzzerDigital(5); // D5ポートを指定 setInterval(function(){ buzzerDigital.write(buzzer); // ブザーを鳴らす・止める if(buzzer){ buzzer = false; }else{ buzzer = true; } },1000); // 1000ミリ秒(=1秒)のインターバル }); board.init();

付録 センサーのサンプルコード5

9. LCD RGBバックライト(lcdrgb.js)

```
// LCD RGBバックライト
var GrovePi = require('node-grovepi').GrovePi;
var i2c = require('i2c-bus');
var sleep = require('sleep/');
var Commands = GrovePi.commands;
var Board = GrovePi.board;
var DISPLAY RGB ADDR = 0x62;
var DISPLAY TEXT ADDR = 0x3e;
function setRGB(i2c1, r, g, b) {
i2c1.writeByteSync(DISPLAY_RGB_ADDR,0,0)
 i2c1.writeByteSync(DISPLAY RGB ADDR,1,0)
 i2c1.writeByteSync(DISPLAY RGB ADDR,0x08,0xaa)
 i2c1.writeByteSync(DISPLAY RGB ADDR,4,r)
 i2c1.writeByteSync(DISPLAY RGB ADDR,3,g)
 i2c1.writeBvteSvnc(DISPLAY RGB ADDR.2,b)
}
function textCommand(i2c1, cmd) {
i2c1.writeByteSync(DISPLAY TEXT ADDR, 0x80, cmd);
}
function setText(i2c1, text) {
textCommand(i2c1, 0x01) // clear display
 sleep.usleep(50000);
 textCommand(i2c1, 0x08 | 0x04) // display on, no cursor
 textCommand(i2c1, 0x28) // 2 lines
 sleep.usleep(50000);
 var count = 0;
 var row = 0;
```

```
for(var i = 0, len = text.length; i < len; i++) {
  if(text[i] === '¥n' || count === 16) {
   count = 0;
   row ++;
    if(row === 2)
      break:
   textCommand(i2c1, 0xc0)
   if(text[i] === '¥n')
    continue;
  }
  count++;
  i2c1.writeByteSync(DISPLAY TEXT ADDR, 0x40,
text[i].charCodeAt(0));
}
}
var board = new Board({
 debua: true,
 onError: function(err){
  console.log('ERROR');
  console.log(err)
 },
 onInit: function(res){
  if(res){
   var i2c1 = i2c.openSync(1); // I2Cポートを指定
   setRGB(i2c1, 255, 255, 255); // バックライトのRGB値を指定
   setText(i2c1, "Hello¥nWorld!"); // 表示テキストを指定
   i2c1.closeSvnc();
  }
 }
});
board.init();
```

END

ハンズオン講習会テキスト

