

アキラポニックス作ってみた

3B #25はせが"ありくお

目次

I. 序論

- 1) アクアポニックスとは
- 2) アクアポニックスの種類
- 3) メディアでの活躍とデメリット

II. 本論

- 1) システムアーキテクチャ図
- 2) 物理レイヤー
- 3) 電子レイヤー
- 4) 組み込み
- 5) バックエンド
- 6) フロントエンド

III. 結論

IV. 参考資料

I. 序論

1) アクアポニックスとは

アクアポニックスについて延々と語る前に、少しアクアポニックスの概要について述べる。アクアポニックスは新しい食生産の形で、近年少し人気が出てきている。アクアポニックスという言葉はアクアカルチャーとハイドロポニックスを合わせた複合語である。アクアカルチャーは魚の養殖ことである。ハイドロポニックスは水耕栽培という農法のことである。近年国内でも水耕栽培は大手電子機器メーカーが参入して業界としては急成長を遂げている。メディアなどでは「植物工場」と呼んでいることもある。水耕栽培では文字通り、露地栽培と違い、土を使わずに、水を「耕」して植物を育てている、人工の栄養素を水に混ぜ、植物に吸収させて育てている。

アクアポニックスはその両者の利点を保ちながら合成させようとした食生産の形である。

“Aquaponics is the cultivation of fish and plants together in a constructed, recirculating ecosystem utilizing natural bacterial cycles to convert fish waste to plant nutrients. (...) without the need to discard any water or filtrate or add chemical fertilizers.”

-Aquaponics Gardening Community, November 2010

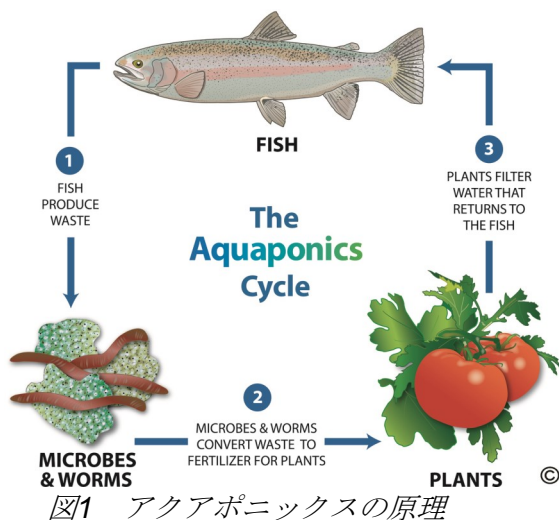


図1 アクアポニックスの原理

アクアポニックスは主に魚、野菜、バクテリアに構成された循環型農業である。魚に餌を与える部分が唯一の外部インプットである。魚は生きていく中でアンモニアを出しますが、そのアンモニアをバクテリアが硝酸塩や亜硝酸塩へ分解する。アンモニアは魚と植物にとっては有害であり、栄養にもならないが 硝酸塩は植物が吸収できるので植物が成長する。そして、アンモニアや硝酸塩が吸収されて魚たちには綺麗にフィルタリングされた水が戻る。この循環がアクアポニックスの核となる窒素循環である。[1][2]

2) アクアポニックスの種類

アクアポニックスシステムには数種類あるが、主に3つに分類できる。そのひとつはDWC (Deep Water Culture) タイプである。深さ30cmほどの水槽に発泡スチロールなどでできたラフトを浮かべ、植物をそれに刺し込んで育てる。根が水に浮かんでいるので根の観察がとても容易であり、根を洗う必要もない。[2][3][4]



図2 The Plant Chicago にあるDWCタイプシステム

ふたつ目はMFG (Media Filled Growbed) タイプである。MFGタイプのシステムには媒質があり、その媒質の表面にバクテリアが生息している。最も人気な媒質はハイドロトンという粘土の密度を下げ、焼いたものであるが、そのほかにも火山岩や砂利など色々なものが使われている。媒質がフィルターの代わりになるのでMFGタイプは他より設計しやすい場合が多い。そのためホビイストには人気である。

MFGタイプでは植物や媒質があるタンクの水位を定期的に上下させ、根が腐らなく、バクテリアが好気呼吸できるようにする必要がある。小規模のシステムの場合はMFGタイプが最も無難であり、人気もある。

[2][3][4]



図3 The Plant ChicagoにあるMFGタイプシステム

みつつ目はNFT(Nutrient Film Technique)タイプである。NFTタイプでは栄養が多い水をパイプ



に流して薄い膜を作り、その膜に根をたらしめて育てる。このタイプは水槽の水と植物タンクの水の割合の差が大きいので、メインとなるタイプよりは、上記の2タイプを補完するような使い方をする。もし栄養過多が起きているときにはNFTタイプを使って藻の大量発生を防げる。[2][3][4]

3) メディアでの活躍とデメリット

アクアポニックスはメディアで取り上げられていて、世間の農業への関心と共に徐々に認知度も上がってきている。[5][6][7] しかし、主な報道は興味や好奇心をそそるものであって、真剣に露地栽培に取って代わる農法としては扱われない。その理由は、アクアポニックスでは主に葉物野菜などしか生産できず、根野菜などを育てるのはとても困難であるためである。そのほかにも養殖できる魚が淡水魚に限られる点もある。

図4. NFTタイプシステム



Fig. 5 Micro Greens being raised in an aquaponics system at The Plant Chicago

II.本論

1) システムアーキテクチャ図

今回は前述のMFGシステムをWebから制御するプラットフォームを製作した。

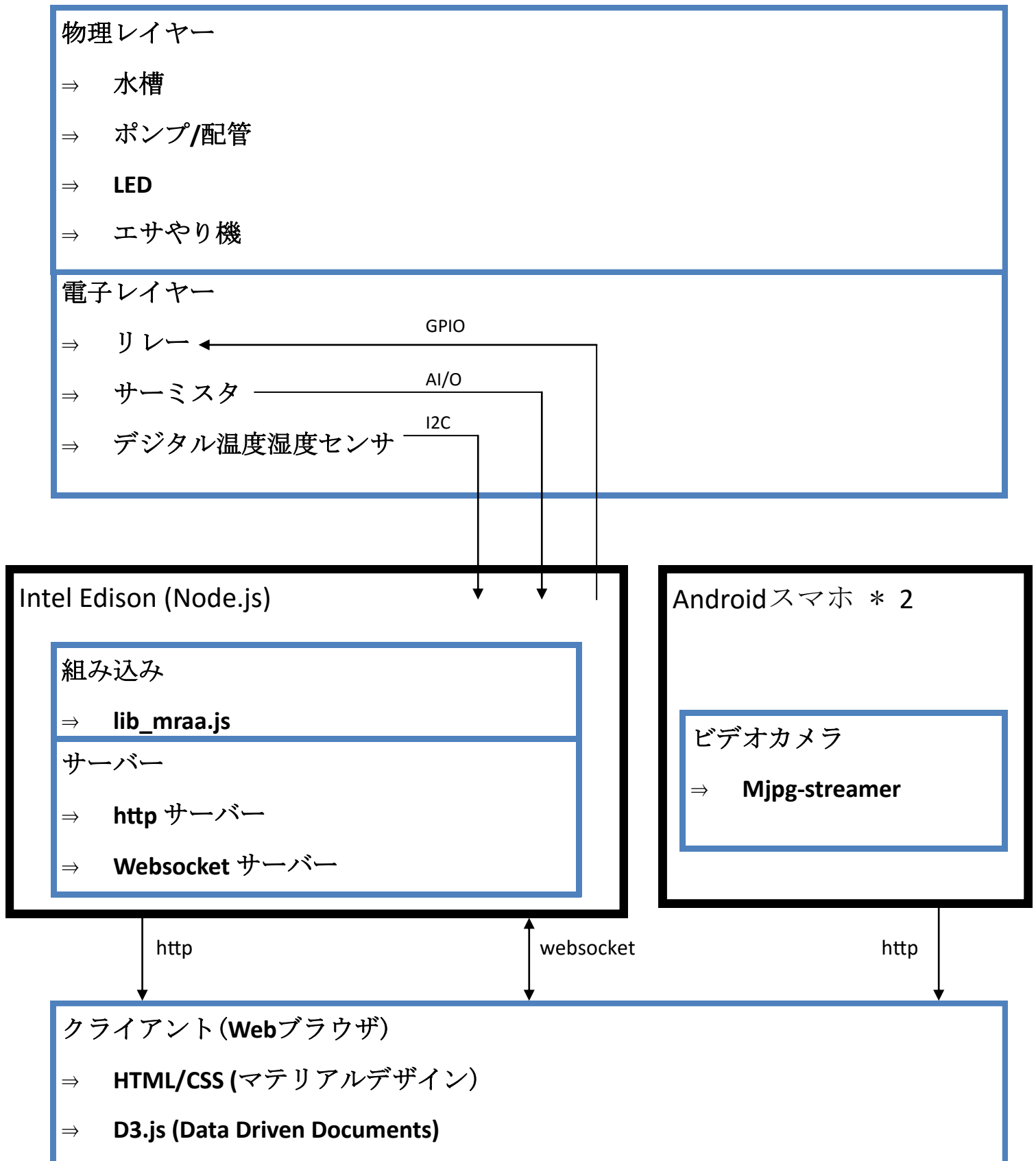


Fig. 6 システムアーキテクチャ図

2) 物理レイヤー

水槽/ポンプ/配管：

50ℓの水槽をふたつ用意し、上部の箱の底面に内径13mmと16mmの塩ビ管用の穴を開ける。その中に内径13mmと16mmの塩ビ管を入れ、Oリングで留める。13mmの塩ビ管の上部はMFGのGrowbed,プランター部分の水位の下限を定義するように配置する。下部の水槽端の13mmのパイプを切り、T字連結部位とボールバルブを付けて水流調節を可能にする。コスト面での理由によりこの部位は電氣的に制御できなかったが、技術的には容易である。

16mmの塩ビ管はGrowbedの水位の上限を定義するように配置する。

このような配管によって厳密に流量と時間を誤差なく定義して制御する必要がなくなる。

n = ポンプがONな時間

の場合、

$n > \text{Growbedが満ちる時間}$

であれば正しく機能する。

また、ハイドロトンや根がパイプを詰まらせないようにガードパイプを置き、水のみが循環する構造にする。

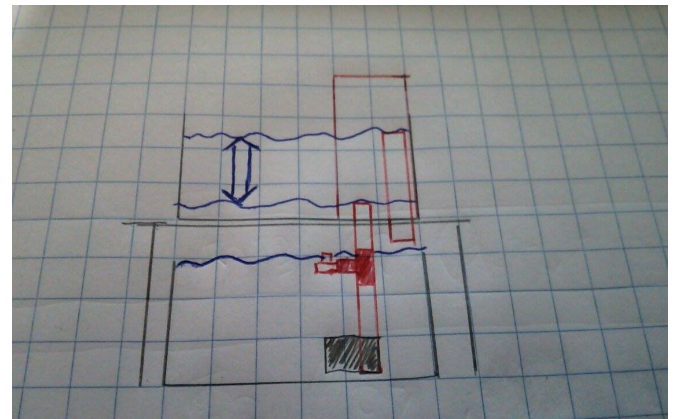


Fig 7 配管の構造

LED:

植物の光合成のための人工照明は最もエネルギー効率の高いLEDを用いる。本来植物の光合成のために必要な波長は、葉緑体の中あのクロロフィルの吸収スペクトルが集中する~460nm帯と~680nm帯であるが、複数のLEDテープを購入するとコストが2倍になるのでさまざまな波長を包括的に発す白色LEDを採用した。LEDは設備の中で最もコストのかかる部位で、AC/DCアダプタなどを含めて1万円ほどした。

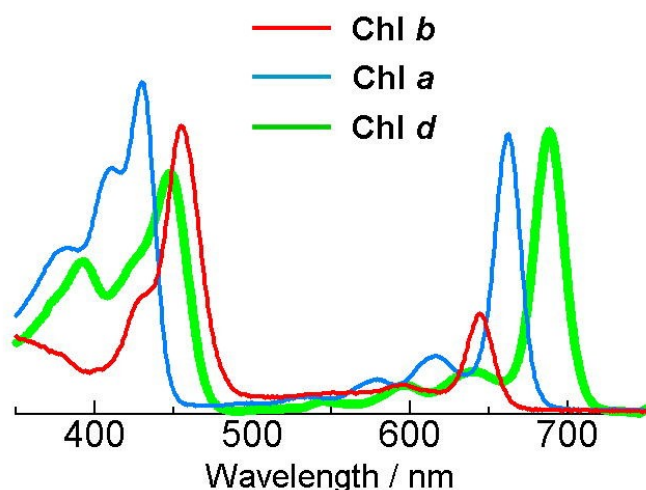


Fig 8 クロロフィルの吸収スペクトル



Fig 9 LED アレー

エサやり機：

コスト面と分解/改造のしやすさを考慮してAF-2003という型の自動エサやり機を採用。単純な回路であるため簡単に制御できた。仕組みとしては、エサが多く入っている室が回転して、小さな弁を通して一定の量のエサが出力される形。弁についているノブを回すことによってエサの量が調節できる。

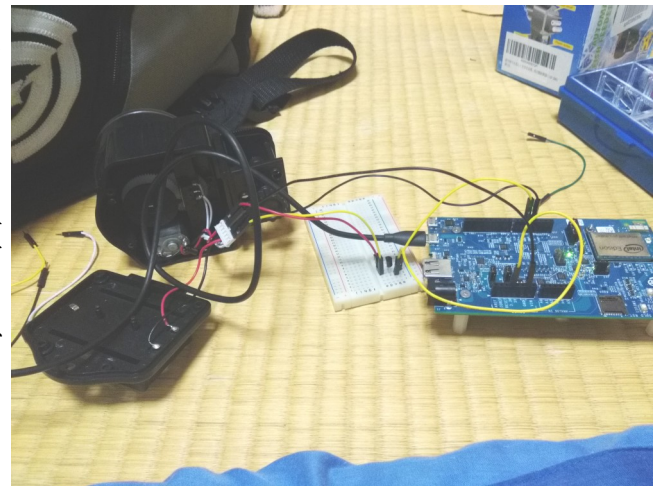


Fig 10 AF-2003 とIntel Edison

電子レイヤー

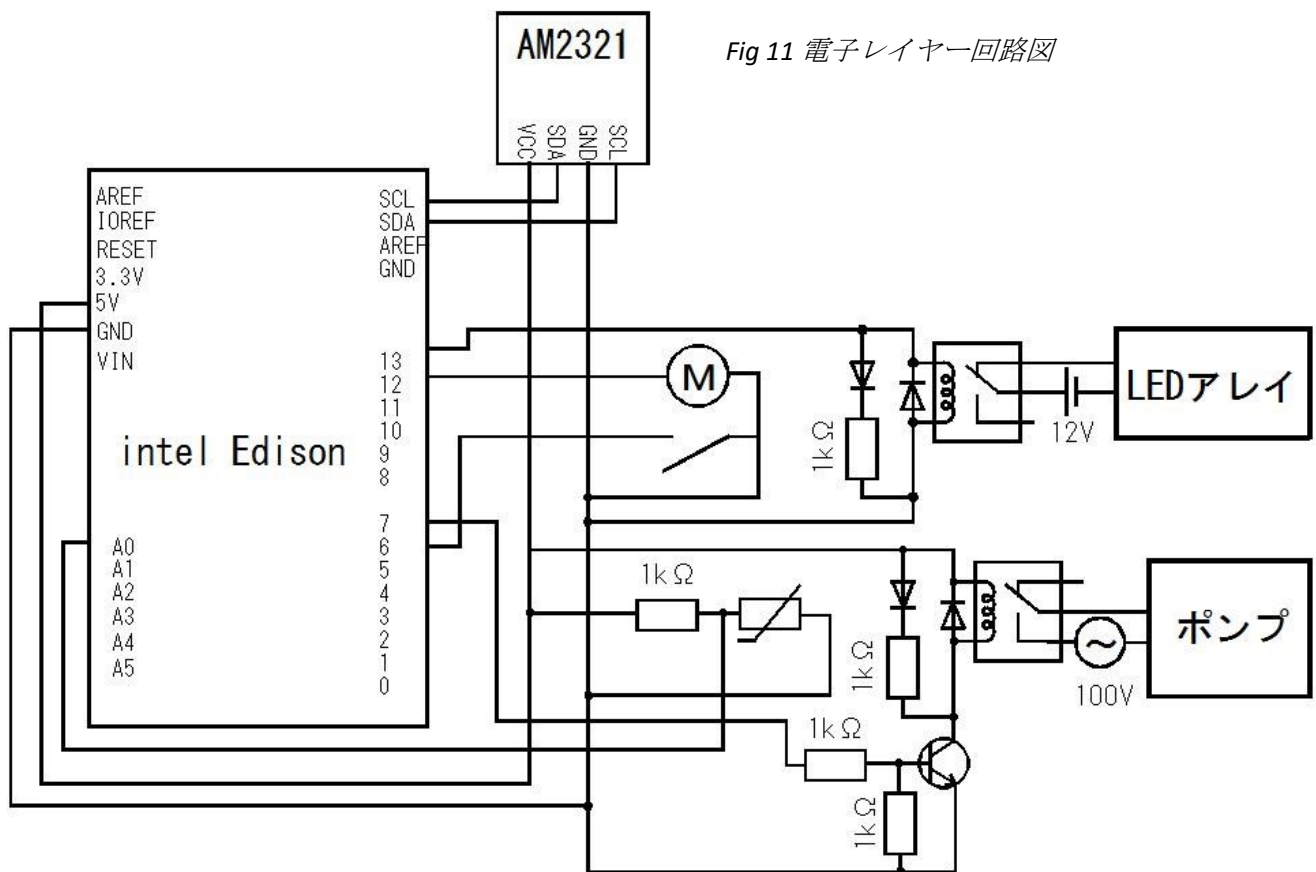


Fig 11 電子レイヤー回路図

センサ：デジタル温度湿度センサ、サーミスタ（水温センサ）

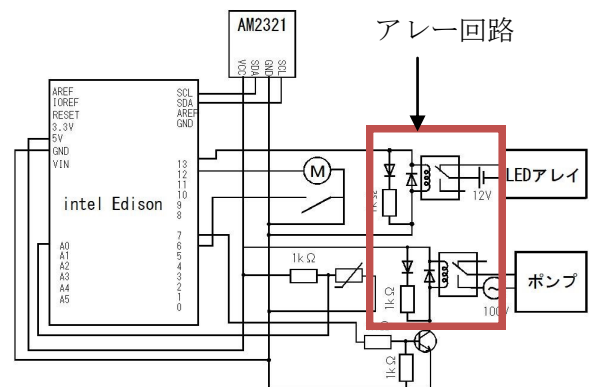
アクチュエータ：エサやり機、ポンプ、LEDアレイ

様々なセンサやアクチュエータをアナログ入出力(AI/O)、GPIO(General Purpose Input/Output), I2Cなどの規格に沿って制御した。

リレー：

「リレーとは、継電器ともいい、①電磁石 ②接点機構 で構成され「電磁石に、ある値以上の電流を流した時に生ずる電磁吸引力を利用して、接点機構を作動させるもので、コイルに与えられる電圧、電流（入力信号）により、接点の開閉を行なうもの」と定義しています。」[11]

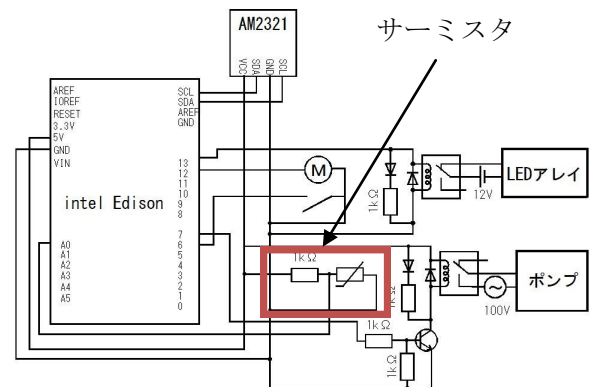
このような仕組みを使うことによって、低電圧(5VDC)なGPIOから高電圧な100VACや12VDCの制御が行える。よって、LEDアレーの制御やポンプの制御にはリレーを採用した。



サーミスタ：

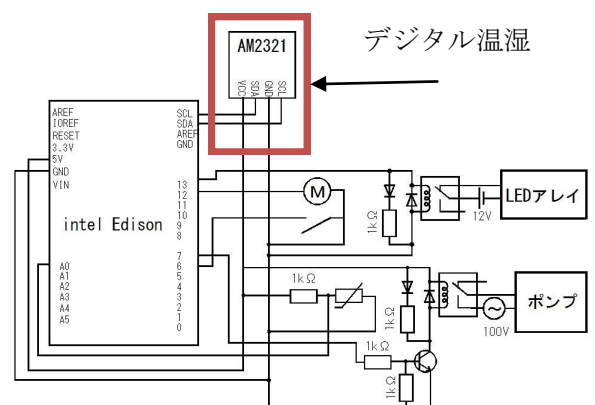
サーミスタは温度の変化に応じて抵抗値が変わる電子部品である。抵抗値の変化によってNTCサーミスタとPTCサーミスタの二種に分類される。[12]

今回は抵抗値がある関数上に変化するNTCサーミスタを水温センサとして採用した。サーミスタの読み込みにはアナログ入力力 (Analog Input)を利用した。



デジタル温度湿度センサ：

気温と湿度を計測するためにはAM-2321を利用しました。AM-2321はデジタルセンサであるが、I2Cという通信規格を利用している。I2CとはInter-Integrated Circuitの略称で、フィリップス社が提唱した周辺デバイスとコントローラーの通信の方式である。SlaveとMaster間の通信という形をとっていて、ひとつにMasterに複数のSlaveをつなげることが可能である。近距離の通信に向いていて、100kbps/400kbpsで通信するのが主である。[13]



Intel Edison (Node.js):

Intel Edison(以後Edison)とは、Intel社が販売している2コア、1GBRAM, 4GBフラッシュメモリを搭載した小型コンピュータの商標である。Edisonの端子はとても小さいため拡張基盤を使うのが主流である。Edisonではyocto-linuxというOSが動作するため、簡単に様々な機能が利用できる。今回はNode.jsを用いて組み込みとバックエンドを一括で開発した。



Fig 12 Intel Edison

Node.js:

JavaScript言語は元々ブラウザ内、フロントエンド用の言語であったが、2009年にNode.jsというJavaScriptで書くことのできるバックエンドプラットフォームが公開された。

Node.jsは「非同期イベント駆動」である。C言語などのプログラムは $A \rightarrow B \rightarrow C$ と順番にプログラムを実行していくが、それに対して非同期イベント駆動なNode.jsはイベント駆動である。イベント駆動あるイベント、例えばボタンが押される、情報を読み終わる、クライアントが情報を送信してくる、タイマーが鳴る、などに対して反応する。非同期というのは英語ではNon-Blocking, 「塞がない」という意味である。データの読み込み、ダウンロード中に他の処理をBlock, 塞ぐことがないためデータを扱う場合に指数関数的に処理が重くなることがない。

Node.jsではmoduleと呼ばれるライブラリのようなものを組み合わせてできていて、npm(node package manager)でmoduleを管理できる。

組み込み:

Edisonの入出力端子 (AI/O, GPIO, I2Cなど)を利用するためにIntelはlib_mraa.jsというmoduleを用意している。[14]

組み込みシステムとは、特定の機能を実装するために組むプログラムのことを指すが、一般的に電子機器を直接制御などするものを組み込みと呼ぶこともある。Linux上で電子機器を動かす場合は組み込みLinuxなどと呼んだりする。今回は組み込みLinuxシステムを開発した。

上記の電子レイヤーの制御、またセンサから意味のある値 (℃の温度など) を抽出のが主となる。

サーバー :

Edison上で動くNode.jsのプログラムでは2種類のサーバーを利用している。ひとつは静的ファイルの供給のためのhttpサーバーである。もうひとつは動的データ送受信のwebsocketサーバー(厳密にはsocket.io)である。それぞれmoduleがあり、`require('http');`のように呼び出すことができる。

http:

ウェブサイトのURLの冒頭に“http://”などと書かれているhttpと同じhttpである(最近のブラウザではほとんどの場合省略できるが)。httpとは、HyperText Transfer Protocolの略称であり、その名の通り通信規格である。HTTP GETリクエストに応じてサーバーがHTML/CSS/JSなどのファイルを送信してそれを受け取ったクライアント(ブラウザ)が解釈してレンダリングするのが普通のhttpのやり取りである。PUSHやPOSTなどもあるがそこにはこの際は触れない。

コードを読んでいただければわかるが、httpサーバーでは/publicディレクトリ内の静的ファイルを供給している。

WebSocket/socket.io:

WebSocketプロトコルはhttpと違い、双方向リアルタイム通信を目的に作られている。なおhttpもWebsocketも両方ともTCP上で動いているため「インターネット技術」であることに違いはない。WebSocketプロトコルはやや新しい技術であるため、古いブラウザなどは対応していないことがある。

そこでSocket.io というmoduleが便利である。Socket.ioはデフォルトでWebSocketプロトコルを使おうとするが失敗した際には別の通信規格で代用する機能を搭載したWebSocketサーバーmoduleである。

今回はセンサーのデータの送信、ユーザーからの制御パターン依頼の受信などはSocket.ioを介して行っている。

ルートディレクトリ内のserver.jsでほとんどのsocket.io機能を実装している。

Android端末 (mjpg-streamer):

動画の生配信のためにmjpg形式(motion-jpeg)で送信する。mjpgは文字通り、動画の毎フレームをjpg形式で保存するためHTMLのタグで利用できる。

Androidアプリでmjpgサーバーを建てることのできたのでとても楽な実装だった。Android端末にカメラが搭載されていけばよいので、格安スマホなどを利用すればカメラの数を無数に増やすことが可能である。

クライアント：

今まで構築してきたバックエンド（サーバー/通信）から得た情報の表示方法がクライアントサイドである。今回はWebアプリという形で開発したためブラウザでの表示を目的とする。ブラウザは HTML (Hyper Text Markup Language)で記された DocumentをもとにCSS (Cascading Style Sheets)やクライアントサイドJavaScriptを読み込みサイトをレンダリングしていく。つまりUIデザインはHTML/CSS/JSで記される。

クライアントサイドのロジックは主にボタンなどが押されたときに応じてのイベント駆動プログラミングで行った。Socket.io通信に関するロジックが主になる。

マテリアルデザイン：

今回のUIは「マテリアルデザイン」というデザイン手法を取り入れた。

マテリアルデザインでは「物」、マテリアルのようにUIが動くようにデザインする。そのためにはアニメーションや影をつけ、平面の画面の中で生き生きと立体が存在するように作る。このような贅沢なレンダリングは処理能力とネットワーク速度に余裕が出てきた近年のコンピューターだからこそできるものである。

D3.js：

過去データを可視化するためのグラフはD3.js (Data Driven Documents)を使った。

D3.jsはNHKなどの大手ニュース番組のデータ可視化ツールとしても使われるオープンソースライブラリである。HTML5の<svg>に対応しており、とても優れた可視化ツールである。

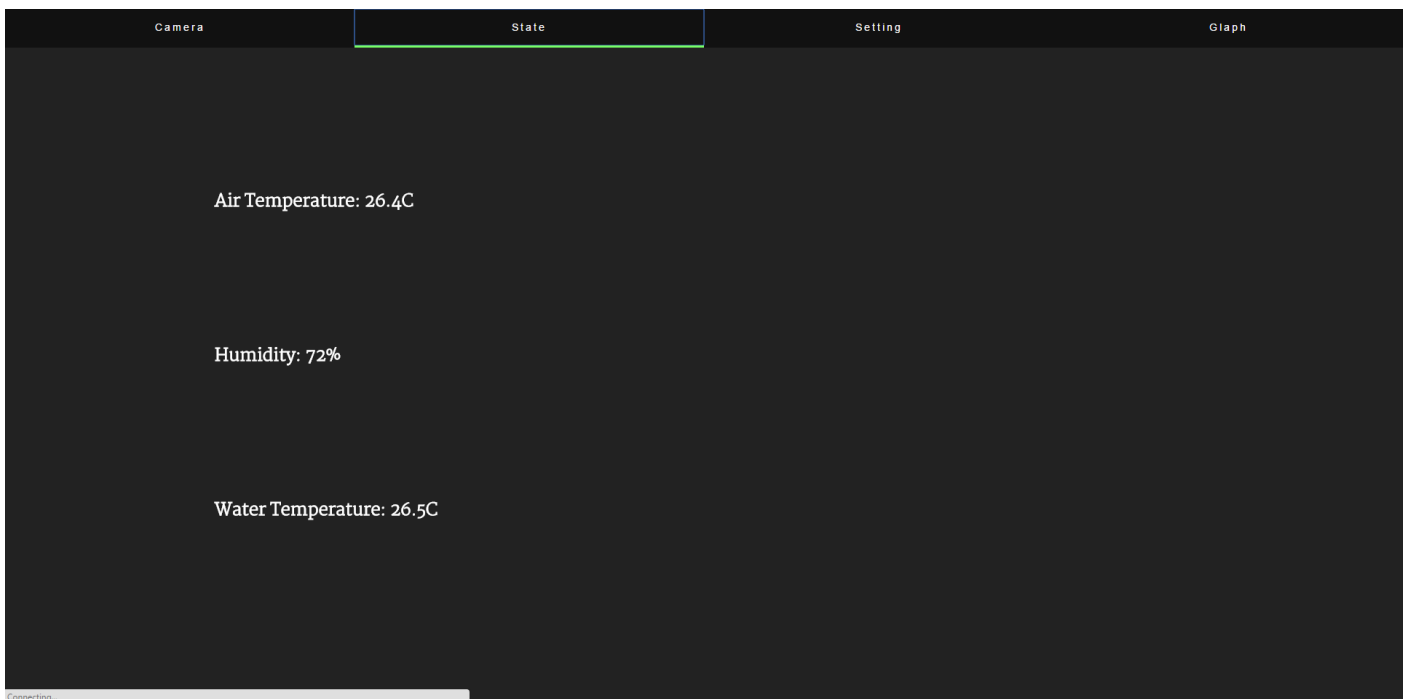


Fig 13 フロントエンドUI

III. 結論

このプロジェクトは先日、8月29日にYahoo Japanで開催された開発コンテストInnovator's Summerに出場し、無事Yahoo基金賞とSoftbank賞を受賞できた。関係者各位に感謝の言葉を贈りたい。

このように自分の思い描いたシステム設計に沿って開発を進め、思い通りに様々な部品が各々の定位置にはまって行く感覚はエクスタシーに他ならない高揚感を感じた。また、他人と2週間ほど連続で終電ギリギリまで共同開発する限界状態も適度なアドレナリンが発されて心地よいものだった。

実際に0からシステムを構築していけるという自信もつき、激しく成長できたとも思うので今後も開発を進めていきたい。



Fig 14 完成した瞬間

ソースコード

ディレクトリ構造

ルートディレクトリ/

node_modules/

- mime/
- mraa-js/
- path/
- socket.io/

public/

js/

- client.js
- animation.js
- socket.io.js
- air_temp_graph.js
- water_temp_graph.js
- humidity_graph.js

- index.html
- style.css

- index.js
- mraa.js
- server.js
- package.json

* このコードは https://github.com/Spaghet/aquaponics_edison/ で随時更新されていく

14 index.js

```
var socketio = require('socket.io');

var io = require('./server.js');

var mraa = require('./mraa.js');

var http = require('http');

var path = require('path');

var fs = require('fs');

var mime = require('mime');

var cache = {};


//sends 404 not found response

function send404(response) {

    response.writeHead(404, { 'Content-type': 'text/plain' });

    response.write('Error 404: resource not found\n');

    response.end();

}


//sends file by looking up mime type from file extension

function sendFile(response, filePath, fileContents) {

    response.writeHead(

        200,

        { "Content-type": mime.lookup(path.basename(filePath)) }

    );

    response.end(fileContents);

}
```

```

//caches file or sends 404 if file not found
function serverStatic(response, cache, absPath) {
  if (cache[absPath]) {
    sendFile(response, absPath, cache[absPath]);
  } else {
    fs.exists(absPath, function (exists) {
      if (exists) {
        fs.readFile(absPath, function (err, data) {
          if (err) { send404(response); }
          else {
            cache[absPath] = data;
            sendFile(response, absPath, data);
          }
        });
      } else {
        send404(response);
      }
    });
  }
}

//da http server
var server = http.createServer(function (request, response) {
  var filePath = false;
  if (request.url == '/') {
    filePath = './public/index.html';
  } else {
    filePath = 'public' + request.url;
  }
  var absPath = './' + filePath;
  serverStatic(response, cache, absPath);
});

server.listen(80, function () {
  console.log('Server listening on http://localhost/¥n');
});

io.listen(server);

```

```

16 server.js
var socketio = require('socket.io');

var mraa = require('./mraa.js');

function state(socket) {

    socket.emit("waterTemp" , mraa.waterTemp());

    socket.emit('tempHum', mraa.tempTest());

}

var ioServer = function (server) {

    io = socketio.listen(server); //run the socket.io server on top of the http server
    io.sockets.on('connection', function (socket) {

        socket.on('toggle', function () {

            console.log("socket io toggle");

            mraa.feed();

        });

        setInterval(state, 10000, socket);

        socket.on('control', function (data) {

            console.log(data);

            switch (data.LED) {

                case 'on': mraa.led.on(); break;

                case 'equinox': mraa.led.equinox(); break;

                case 'summer': mraa.led.summer(); break;

                case 'winter': mraa.led.winter(); break;

                case 'off': mraa.led.off(); break;

            }

            setInterval(mraa.feed(), parseInt(data.feed) * 3600000);

            setInterval(mraa.pump(), parseInt(data.pump) * 60000);

        });

    });

};

module.exports.listen = ioServer;

```


17 mraa.js

```
var m = require('mraa'); //require mraa
console.log(m.getVersion());
```

```
var sunrise = new Date();
var sunset = new Date();
//get pin 13
var waterTempPin = new m.Aio(0);
var feeder = new m.Gpio(12);
var ledPin = new m.Gpio(13);
var sw1 = new m.Gpio(6);
var wet = new m.Gpio(11);
var pumpPin = new m.Gpio(7);
```

```
//set direction
sw1.dir(m.DIR_IN);
ledPin.dir(m.DIR_OUT);
feeder.dir(m.DIR_OUT);
wet.dir(m.DIR_IN);
pumpPin.dir(m.DIR_OUT);
```

```
var analogValueFloat = waterTempPin.readFloat();
var ledState = true; //led state bool
```

```
var led = {
  on: function () { ledPin.write(1); },
  off: function () { ledPin.write(0); },
  equinox: function () {
    setInterval(function () { led.write(ledState ? 1:0); ledState = !ledState; }, 43200000);
  },
  summer: summer,
  winter: winter,
};
```

```
//feed the fish yo
function feed()
{
  var sw = sw1.read();
  if (sw) {
    feeder.write(1);
    setTimeout(function () { feeder.write(0); }, 1000);
  } else {
    feeder.write(1);
    setTimeout(feed, 50);
  }
}
```

```

//return water temperature
function waterTemp() {
    analogValueFloat = waterTempPin.readFloat(); //read the pin value as a float
    var v = (analogValueFloat * 5); //voltage
    var r = (10 * v) / (5 - v); //resistance
    var temp = (-21.19 * Math.log(r) + 74.08);

    return temp;
};

//read temperture&humid
function Am2321(bus, address, bufsize) {
    this.x = new m.I2c(bus);
    this.x.address(address);
    this.buff = new Buffer(bufsize);

    this.wakeupSensor = function () {
        this.x.writeReg(0, 0);
    }

    this.sendCommand = function () {
        get_command = new Buffer([0x03, 0x00, 0x04]);
        this.x.write(get_command);
    }

    this.receiveData = function () {
        this.buff = this.x.read(8);
    }

    this.checkCrc = function () {
        var crc = this.buff[6] + this.buff[7] * 256;
        return crc == crc16(this.buff, 6);
    }

    this.checkRecieveData = function () {
        return (this.buff[0] == 0x03) && (this.buff[1] == 0x04);
    }

    this.getHumidity = function () {
        var high = this.buff[2];
        var low = this.buff[3];
        return (high * 256 + low) / 10;
    }
}

```

```

this.getTemperature = function () {
    var temperature = 0;
    var high = this.buff[4];
    var low = this.buff[5];

    if (high & 0x80) {
        temperature = -1 * (high & 0x7F) * 256 + low;
    }
    else {
        temperature = high * 256 + low;
    }
    return temperature / 10;
}
}

function tempTest(){
    var sensor = new Am2321(6, 0x5C, 8);
    sensor.wakeupSensor();
    sensor.sendCommand();
    sensor.receiveData();
    if (sensor.checkRecieveData() == true) {
        var humidity = sensor.getHumidity();
        var temperature = sensor.getTemperature();
    }
    return [temperature, humidity];
}

var continueLoop = true;
//timerLed
function timerLed(sunrise, sunset) {
    var currentDate = new Date();
    var isDay = true;

    if (isDay && currentDate.getHours() === sunset.getHours() && currentDate.getMinutes() === sunset.getMinutes()) {
        ledPin.write(0);
    }
    if (!isDay && currentDate.getHours() === sunrise.getHours() && currentDate.getMinutes() === sunrise.getMinutes()) {
        ledPin.write(1);
    }

    if(continueLoop){ setTimeout(timerLed, 10000, sunrise, sunset);}

}

function summer() {
    continueLoop = false;
    setTimeout(null, 10010);
    continueLoop = true;
    sunrise.setHours(4, 26);
    sunset.setHours(19, 1);
    timerLed(sunrise, sunset);
}

```

```
function winter() {  
  continueLoop = false;  
  setTimeout(null, 10010);  
  continueLoop = true;  
  sunrise.setHours(6,47);  
  sunset.setHours(16,32);  
  timerLed(sunrise, sunset);  
}  
var isPump = true;  
function pump(){  
  pumpPin.write(isPump ? 1:0);  
}  
//export functions  
module.exports.feed = feed;  
module.exports.led = led;  
module.exports.waterTemp = waterTemp;  
module.exports.tempTest = tempTest;  
module.exports.pump = pump;
```



```

<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="style.css">
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
  <script src="js/socket.io.js"></script>
  <script src="js/animation.js"></script>
  <script src="js/client.js"></script>
  <script type="text/javascript">
    var io = io();
  </script>
  <title>Plannet</title>
</head>
<body>
  <div id="Header">
    <div id="HeaderScroll">
      <button class="HeaderBtn" onclick="ChangePageTo1()">
        Camera
      </button>
      <button class="HeaderBtn" onclick="ChangePageTo2()">
        State
      </button>
      <button class="HeaderBtn" onclick="ChangePageTo3()">
        Setting
      </button>
      <button class="HeaderBtn" onclick="ChangePageTo4()">
        Glaph
      </button>
    </div>
    <div id="HeaderBorderBox">
      <div id="HeaderBorder">
      </div>
    </div>
  </div>
  <div id="Main">
    <div id="MainScrollBox">

    <div class="MainContent">
      <button id="ChangeCameraBtn" onclick="ChangeCamera()">Change Camera</button>
      <div id="CameraBox">
        <div id="CameraScrollBox">
          
          
        </div>
      </div>
    </div>
  </div>
</div>

```

```

<div class="MainContent">
  <div class="StateContent">
    <h3 class="temp">Temperature</h3>
  </div>
  <div class="StateContent">
    <h3 class="hum">Humidity</h3>
  </div>
  <div class="StateContent">
    <h3 class="waterTemp">Water Temperature</h3>
  </div>
</div>
<form name="set" class="MainContent" style="background:#111;">
  <div class="MainForm">
    <div class="MainFormHeader">
      現在の様子
    </div>
    <select class="MainFormSelect" name="LED">
      <option value="on">常時ON</option>
      <option value="equinox">春分、秋分</option>
      <option value="summer">夏至</option>
      <option value="winter">冬至</option>
      <option value="off">常時OFF</option>
    </select>
  </div>
  <div class="MainForm">
    <div class="MainFormHeader">
      餌やりのやる間隔
    </div>

    <select class="MainFormSelect" name="feed">
      <option value="48">48時間おき</option>
      <option value="24">24時間おき</option>
      <option value="12">12時間おき</option>
    </select>

  </div>

  <div class="MainForm">

    <div class="MainFormHeader">
      現在、次の満ち引き
    </div>

    <select class="MainFormSelect" name="pump">
      <option value="25">25分おき</option>
      <option value="20">20分おき</option>
      <option value="15">15分おき</option>
      <option value="10">10分おき</option>
    </select>

  </div>
  <button type="button" id="MainFormButton" onclick="setting()"> 更新</button>
</form>

```

```
<div class="MainContent">
  <svg id="air_temp"></svg>
  <svg id="humidity"></svg>
  <svg id="water_temp"></svg>
  </div>
</div>
</div>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script src="js/air_temp.js" charset="utf-8"></script>
<script src="js/water_temp.js" charset="utf-8"></script>
<script src="js/humidity.js" charset="utf-8"></script>
</body>
</html>
```

```

@import url(http://fonts.googleapis.com/css?family=Martel:200);
html,body{
    font-family: 'Martel', serif;
    height: 100%;
    margin: 0;
    overflow: hidden;
}
#Header{
    width: 100%;
    height: 53px;
    background: #111;
    position: relative;
}
#HeaderScroll{
    width: 100%;
    height: 50px;
    background: #111;
    position: relative;
}
.HeaderBtn{
    width: 25%;
    height: 50px;
    border: none;
    background: 0;
    color: #fff;
    float: left;
    letter-spacing: 3px;
    font-size: 14px;
}
#HeaderBorderBox{
    width: 100%;
    height: 3px;
    background: #111;
}
#HeaderBorder{
    width: 25%;
    height: 3px;
    background: #7f6;
    position: absolute;
    left: 0;
}
#Main{
    width: 100%;
    height: calc(100% - 53px);
    background: #222;
    position: relative;
    overflow-x: hidden;
}
#MainScrollBox{
    width: 400%;
    height: 100%;
    position: absolute;
    left: 0;
}
.MainContent{
    width: 25%;
    height: 100%;
    position: relative;
    float: left;
    background: #222;
    overflow: hidden;
}
#ChangeCameraBtn{
    width: 100%;
    height: 40px;
    background: #444;
    text-align: center;
    border: none;
    color: #fff;
}
#CameraBox{
    width: 100%;
    height: calc(100% - 40px);
    position: relative;
    overflow: hidden;
}
#CameraScrollBox{
    width: 100%;
    height: 200%;
    position: absolute;
}
#Camera1{
    width: 100%;
    height: 50%;
    vertical-align: bottom;
}

```



```

#Camera2{
    width: 100%;
    height: 50%;
    vertical-align: bottom;
}

.StateContent{
    width: 80%;
    height: 20px;
    font-size: 20px;
    margin-left: 15%;
    margin-top: 10%;
    color: #fff;
}

.MainForm{
    width: 80%;
    margin-left: 15%;
    margin-top: 7%;
}

.MainFormHeader{
    width: 300px;
    color: #fff;
    font-size: 25px;
}

.MainFormSelect{
    width: 150px;
}

#MainFormButton {
    margin-left: 15%;
    margin-top: 3.5%;
    width: 150px;
}

@-webkit-keyframes Btn1to2{
    0%{ left: 0;}
    100%{ left: 25%;}
}

@-webkit-keyframes Btn1to2{
    0%{ left: 0;}
    100%{ left: 25%;}
}

@-webkit-keyframes Btn1to3{
    0%{ left: 0;}
    100%{ left: 50%;}
}

@-webkit-keyframes Btn1to3{
    0%{ left: 0;}
    100%{ left: 50%;}
}

@-webkit-keyframes Btn1to4{
    0%{ left: 0;}
    100%{ left: 75%;}
}

@-webkit-keyframes Btn1to4{
    0%{ left: 0;}
    100%{ left: 75%;}
}

@-webkit-keyframes Btn2to1{
    0%{ left: 25%;}
    100%{ left: 0%;}
}

@-webkit-keyframes Btn2to1{
    0%{ left: 25%;}
    100%{ left: 0%;}
}

@-webkit-keyframes Btn2to3{
    0%{ left: 25%;}
    100%{ left: 50%;}
}

@-webkit-keyframes Btn2to3{
    0%{ left: 25%;}
    100%{ left: 50%;}
}

@-webkit-keyframes Btn2to4{
    0%{ left: 25%;}
    100%{ left: 75%;}
}

@-webkit-keyframes Btn2to4{
    0%{ left: 25%;}
    100%{ left: 75%;}
}

```

```

@-webkit-keyframes Btn3to1{
  0%{ left: 50%;}
  100%{ left: 0%;}
}

@-webkit-keyframes Btn3to1{
  0%{ left: 50%;}
  100%{ left: 0%;}
}

@-webkit-keyframes Btn3to2{
  0%{ left: 50%;}
  100%{ left: 25%;}
}

@-webkit-keyframes Btn3to2{
  0%{ left: 50%;}
  100%{ left: 25%;}
}

@-webkit-keyframes Btn3to4{
  0%{ left: 50%;}
  100%{ left: 75%;}
}

@-webkit-keyframes Btn3to4{
  0%{ left: 50%;}
  100%{ left: 75%;}
}

@-webkit-keyframes Btn4to1{
  0%{ left: 75%;}
  100%{ left: 0%;}
}

@-webkit-keyframes Btn4to1{
  0%{ left: 75%;}
  100%{ left: 0%;}
}

@-webkit-keyframes Btn4to2{
  0%{ left: 75%;}
  100%{ left: 25%;}
}

@-webkit-keyframes Btn4to2{
  0%{ left: 75%;}
  100%{ left: 25%;}
}

@-webkit-keyframes Btn4to3{
  0%{ left: 75%;}
  100%{ left: 50%;}
}

@-webkit-keyframes Btn4to3{
  0%{ left: 75%;}
  100%{ left: 50%;}
}

@-webkit-keyframes Box1to2{
  0%{ left: 0;}
  100%{ left: -100%;}
}

@-webkit-keyframes Box1to2{
  0%{ left: 0;}
  100%{ left: -100%;}
}

@-webkit-keyframes Box1to3{
  0%{ left: 0;}
  100%{ left: -200%;}
}

@-webkit-keyframes Box1to3{
  0%{ left: 0;}
  100%{ left: -200%;}
}

@-webkit-keyframes Box1to4{
  0%{ left: 0;}
  100%{ left: -300%;}
}

@-webkit-keyframes Box1to4{
  0%{ left: 0;}
  100%{ left: -300%;}
}

@-webkit-keyframes Box2to1{
  0%{ left: -100%;}
  100%{ left: 00%;}
}

@-webkit-keyframes Box2to1{
  0%{ left: -100%;}
  100%{ left: 00%;}
}

```

```

@-webkit-keyframes Box2to3{
  0%{ left: -100%;}
  100%{ left: -200%;}
}

@-webkit-keyframes Box2to3{
  0%{ left: -100%;}
  100%{ left: -200%;}
}

@-webkit-keyframes Box2to4{
  0%{ left: -100%;}
  100%{ left: -300%;}
}

@-webkit-keyframes Box2to4{
  0%{ left: -100%;}
  100%{ left: -300%;}
}

@-webkit-keyframes Box3to1{
  0%{ left: -200%;}
  100%{ left: 00%;}
}

@-webkit-keyframes Box3to1{
  0%{ left: -200%;}
  100%{ left: 00%;}
}

@-webkit-keyframes Box3to2{
  0%{ left: -200%;}
  100%{ left: -100%;}
}

@-webkit-keyframes Box3to2{
  0%{ left: -200%;}
  100%{ left: -100%;}
}

@-webkit-keyframes Box3to4{
  0%{ left: -200%;}
  100%{ left: -300%;}
}

@-webkit-keyframes Box3to4{
  0%{ left: -200%;}
  100%{ left: -300%;}
}

@-webkit-keyframes Box4to1{
  0%{ left: -300%;}
  100%{ left: 00%;}
}

@-webkit-keyframes Box4to1{
  0%{ left: -300%;}
  100%{ left: 00%;}
}

@-webkit-keyframes Box4to2{
  0%{ left: -300%;}
  100%{ left: -100%;}
}

@-webkit-keyframes Box4to2{
  0%{ left: -300%;}
  100%{ left: -100%;}
}

@-webkit-keyframes Box4to3{
  0%{ left: -300%;}
  100%{ left: -200%;}
}

@-webkit-keyframes Box4to3{
  0%{ left: -300%;}
  100%{ left: -200%;}
}

@-webkit-keyframes Camera1to2{
  0%{ top: 0%;}
  100%{ top: -100%;}
}

@-moz-keyframes Camera1to2{
  0%{ top: 0%;}
  100%{ top: -100%;}
}

@-webkit-keyframes Camera2to1{
  0%{ top: -100%;}
  100%{ top: 00%;}
}

```

```
@media screen and (min-width: 901px) {  
  body{  
    width: 100%;  
  }  
}
```

```
@media screen and (max-width: 900px) {  
  body{  
    width: 900px;  
  }  
}
```

```
var NowCamera = 1;
```

```
var NowBox = 1;
```

```
function ChangeCamera () {
  if (NowCamera == 1) {
    $("#CameraScrollBox").css("-webkit-animation","Camera1to2 0.8s ease 0s forwards");
    NowCamera = 2;
  }else{
    $("#CameraScrollBox").css("-webkit-animation","Camera2to1 0.8s ease 0s forwards");
    NowCamera = 1;
  }
}
```

```
function ChangePageTo1 () {
  if (NowBox == 2) {
    $("#HeaderBorder").css("-webkit-animation","Btn2to1 1s ease 0s forwards");
    $("#MainScrollBox").css("-webkit-animation","Box2to1 1s ease 0s forwards");
    NowBox = 1;
  }else if (NowBox == 3){
    $("#HeaderBorder").css("-webkit-animation","Btn3to1 1s ease 0s forwards");
    $("#MainScrollBox").css("-webkit-animation","Box3to1 1s ease 0s forwards");
    NowBox = 1;
  }else if (NowBox == 4){
    $("#HeaderBorder").css("-webkit-animation","Btn4to1 1s ease 0s forwards");
    $("#MainScrollBox").css("-webkit-animation","Box4to1 1s ease 0s forwards");
    NowBox = 1;
  }else{};
}
```

```
function ChangePageTo2 () {
  if (NowBox == 1) {
    $("#HeaderBorder").css("-webkit-animation","Btn1to2 1s ease 0s forwards");
    $("#MainScrollBox").css("-webkit-animation","Box1to2 1s ease 0s forwards");
    NowBox = 2;
  }else if (NowBox == 3){
    $("#HeaderBorder").css("-webkit-animation","Btn3to2 1s ease 0s forwards");
    $("#MainScrollBox").css("-webkit-animation","Box3to2 1s ease 0s forwards");
    NowBox = 2;
  }else if (NowBox == 4){
    $("#HeaderBorder").css("-webkit-animation","Btn4to2 1s ease 0s forwards");
    $("#MainScrollBox").css("-webkit-animation","Box4to2 1s ease 0s forwards");
    NowBox = 2;
  }else{};
}
```

```

function ChangePageTo3 () {
    if (NowBox == 2) {
        $("#HeaderBorder").css("-webkit-animation","Btn2to3 1s ease 0s forwards");
        $("#MainScrollBar").css("-webkit-animation","Box2to3 1s ease 0s forwards");
        NowBox = 3;
    }else if (NowBox == 1){
        $("#HeaderBorder").css("-webkit-animation","Btn1to3 1s ease 0s forwards");
        $("#MainScrollBar").css("-webkit-animation","Box1to3 1s ease 0s forwards");
        NowBox = 3;
    }else if (NowBox == 4){
        $("#HeaderBorder").css("-webkit-animation","Btn4to3 1s ease 0s forwards");
        $("#MainScrollBar").css("-webkit-animation","Box4to3 1s ease 0s forwards");
        NowBox = 3;
    }else{};
}

function ChangePageTo4 () {
    if (NowBox == 2) {
        $("#HeaderBorder").css("-webkit-animation","Btn2to4 1s ease 0s forwards");
        $("#MainScrollBar").css("-webkit-animation","Box2to4 1s ease 0s forwards");
        NowBox = 4;
    }else if (NowBox == 3){
        $("#HeaderBorder").css("-webkit-animation","Btn3to4 1s ease 0s forwards");
        $("#MainScrollBar").css("-webkit-animation","Box3to4 1s ease 0s forwards");
        NowBox = 4;
    }else if (NowBox == 1){
        $("#HeaderBorder").css("-webkit-animation","Btn1to4 1s ease 0s forwards");
        $("#MainScrollBar").css("-webkit-animation","Box1to4 1s ease 0s forwards");
        NowBox = 4;
    }else{};
}

```

31 public/js/water_temp_graph.js

```
(function () { // ----- これより下、関数型プログラムであったほうが楽な関数s などで、無視しておく
// valの存在確認プレディゲート と valがtrueを返すリテラルかを確認するプレディゲート
function existy (val) { return val != null}
function truthy (val) { return (val != false) && existy(val)} //valがtrueなやつかどうか確認プレディゲート
function fail (msg) { throw new Error(msg)} //エラーを吐く

function always(VALUE) { //valueを常に返すクロージャ
  return function() {
    return VALUE;
  };
}

function doWhen (cond, func) { //ifをカプセル化したやつ
  if (truthy(cond))
    return func();

  else
    return undefined;
}

// -----
//objectの宣言

function Graph(obj) {
  var self = this;
  self.width = obj.width | 800;
  self.height = obj.height | 400;
  self.margin = 40;
  if(!existy(obj.data))
    fail("GRAPH.DATA::UNDEFINED")
  self.data = obj.data;

  return {
    set : function (key, value){
      doWhen(!existy(self[key]), function(){
        fail(key + ":: NOT FOUND");
      });
      self[key] = value;
    },
    get : function(key){return self[key]}
  }
}

function LG () {
  LG.graph;
  return {
    init: function(obj){
      LG.graph = new Graph(obj);
      return this;
    },
    render:10
  }
}

function auto_set () {
}

// 表示サイズを設定
var margin = {
  top : 40,
  right : 40,
  bottom: 40,
  left : 40
};

var size = {
  width : 1366,
  height: 590/3
};

// アニメーション判定フラグ
var isAnimated = false;

// 元の表示サイズを保持しておく
margin.original = clone(margin);
size.original = clone(size);

// 縦横比率と現在の倍率を保持しておく
size.scale = 1;
size.aspect = size.width / size.height;
```



```

// 表示するデータ
var data = [
  {value: 24.335236261486692,
   date: "2015-01-28"},
  {value: 24.086511834531663,
   date: "2015-02-28"},
  {value: 24.16942832729206,
   date: "2015-03-28"},
  {value: 23.50580313888993,
   date: "2015-04-28"},
  {value: 24.00358614316591,
   date: "2015-05-28"},
  {value: 24.16942832729206,
   date: "2015-06-28"},
  {value: 24.252336257667963,
   date: "2015-07-28"},
  {value: 23.75474750870748,
   date: "2015-08-28"},
  {value: 24.00358614316591,
   date: "2015-09-28"},
  {value: 23.92065061655103,
   date: "2015-10-28"},
  {value: 24.666769715603294,
   date: "2015-11-28"},
  {value: 24.666769715603294,
   date: "2015-12-28"}
];

// 時間のフォーマット
var parseDate = d3.time.format("%Y-%m-%d").parse;

// SVG、縦横軸などの設定
var win = d3.select(window);
var svg = d3.select("#water_temp");
var g = svg.append("g");
var x = d3.time.scale();
var y = d3.scale.linear();

var xAxis = d3.svg.axis()
  .scale(x)
  .orient("bottom")
  .tickFormat(d3.time.format("%m"));

var yAxis = d3.svg.axis()
  .scale(y)
  .orient("left");

var line = d3.svg.line()
  .x(function(d){ return x(d.date); })
  .y(function(d){ return y(d.value); })
  .interpolate("basis");

// 描画
function render(){
  data.forEach(function(d){
    d.date = parseDate(d.date);
    d.value = +d.value;
  });

  x.domain(d3.extent(data, function(d){ return d.date; }));
  y.domain(d3.extent(data, function(d){ return d.value; }));

  g.append("g")
    .attr("class", "x axis");

  g.append("g")
    .attr("class", "y axis")
    .append("text")
    .attr("transform", "rotate(-90)")
    .attr("dy", 6)
    .attr("dy", ".7em")
    .style("text-anchor", "end")
    .text("値の単位");

  g.append("path")
    .attr("class", "line");
}

```

33 public/js/water_temp_graph.js

```
// グラフサイズの変更
function update(){
    // SVGのサイズを取得
    size.width = parseInt(svg.style("width"));
    size.height = size.width / size.aspect;

    // 現在の倍率を元に余白の量も更新
    // 最小値がそれぞれ30pxになるように調整しておく
    size.scale = size.width / size.original.width;
    margin.top = Math.max(30, margin.original.top * size.scale);
    margin.right = Math.max(30, margin.original.right * size.scale);
    margin.bottom = Math.max(30, margin.original.bottom * size.scale);
    margin.left = Math.max(30, margin.original.left * size.scale);

    // <svg>のサイズを更新
    svg
    .attr("width", size.width)
    .attr("height", size.height);

    // 縦横の最大幅を新しいサイズに合わせる
    x.range([0, size.width - margin.left - margin.right]);
    y.range([size.height - margin.top - margin.bottom, 0]);

    // 中心位置を揃える
    g.attr("transform", "translate(" + margin.left + "," + margin.top + ")");

    // 横軸の位置
    g.selectAll("g.x")
    .attr("transform", "translate(0, " + (size.height - margin.top - margin.bottom) + ")")
    .call(xAxis);

    // 縦軸の位置
    g.selectAll("g.y")
    .call(yAxis);

    // 折れ線の位置
    if (isAnimated){
        g.selectAll("path.line")
        .datum(data)
        .attr("d", line);
    }
}

// アニメーションを実行
function animate(){
    // アニメーション用のダミーデータ
    var dummy = [];
    data.forEach(function(d, i){
        dummy[i] = clone(d);
        dummy[i].value = 23.5;
    });
    g.selectAll("path.line")
    .datum(dummy)
    .attr("d", line)
    .transition()
    .delay(500)
    .duration(1000)
    .ease("back-out")
    .attr("d", line(data))
    .each("end", function(){
        isAnimated = true;
        update();
    });
}

// オブジェクトのコピーを作成する簡易ヘルパー
function clone(obj){
    var copy = {};
    for( var key in obj ){
        if( obj.hasOwnProperty(key) ) copy[key] = obj[key];
    }
    return copy;
}

// extra
(function(e){function t(n){if(a[n])return a[n].exports;var r=a[n]={exports:{},id:n,loaded:!1};return e[n].call(r.exports,r,t,e.exports,t),r.loaded=!0,r.exports}var n=window.webpackJsonp;window.webpackJsonp=function(o,c){for(var s,i,l=0,p=[];l<o.length;l++)i=o[l],r[i]&&p.push.apply(p,r[i]),r[i]=0;for(s in c)e[s]=c[s];for(n&&n(o,c);p.length;)p.shift().call(null,t);return c[0]?(a[0]=0,t(0);void 0):var a={},r={0:0};return t.e=function(e,n){if(0===r[e])return n.call(null,t);if(void 0!==r[e])r[e].push(n);else{r[e]=n;var a=document.getElementsByTagName("script"),o=document.createElement("script");o.type="text/javascript",o.charset="utf-8",o.async=!0,o.src=t.p+""+e+""+(1:"app")[e][e]+".js",a.appendChild(o)},t.m=e,t.c=a,t.p=""},t(0)}(function(e,t,n,a,r,o,c){e.GoogleAnalyticsObject=r,e[r]=e[r]||function(){(e[r].q=e[r].q||[]).push(arguments)},e[r].l=1*new Date,o=t.createElement(n),c=t.getElementsByTagName(n)[0],o.async=1,o.src=a,c.parentNode.insertBefore(o,c)}(window,document,"script","//www.google-analytics.com/analytics.js","ga"),ga("create","UA-40730305-1","auto"),ga("send","pageview"))});

// 初期化
$( "body" ).append( '<meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0,user-scalable=no">' )
.append( '<style>.axis path,.axis line {fill:none;stroke:#666;stroke-width:1.5px;stroke-dasharray: 5px 5px;} .x.axis path {display:none;} .line {fill:none;stroke:#1572F9;stroke-width:1.5px;stroke-dasharray: 5px 5px;} .text {font-size:8px;} .svg {width:100%;}@media screen and (max-width:700px){.text {font-size:8px;}.line {stroke-width:1px;}}</style>' );

render();
update();
animate();
win.on("resize", update);
})();
```

```
$(function ()  
{  
  io.on('waterTemp', function (waterTemp)  
  {  
    $('#waterTemp').html('Water Temperature: ' + waterTemp.toFixed(1) + 'C');  
  });  
  io.on('tempHum', function (tempHum)  
  {  
    $('#temp').html('Air Temperature: ' + tempHum[0] + 'C');  
    $('#hum').html('Humidity: ' + tempHum[1] + '℥%');  
  });  
});  
  
function setting()  
{  
  var frm = document.forms["set"];  
  var led = frm.elements["LED"].selectedIndex;  
  var feed = frm.elements["feed"].selectedIndex;  
  var pump = frm.elements["pump"].selectedIndex;  
  led = frm.elements["LED"].options[led].value;  
  feed = frm.elements["feed"].options[feed].value;  
  pump = frm.elements["pump"].options[pump].value;  
  io.emit('control', {LED: led, feed: feed, pump: pump});  
}
```

IV. 参考資料

1. Bernstein, S.(2011). *Aquaponic Gardening A Step-By-Step Guide to Raising Vegetables and Fish Together*. Gabriola Island, BC, Canada: New Society Publishers
2. Japan Aquaponics. (2014). *General Information About Aquaponics*. Retrieved from <http://www.japan-aquaponics.com/aquaponics.html>
3. Aquaponics Nation. (2014). *Aquaponics Acronyms - Frequently Asked Questions - Articles - Articles - Aquaponics Nation*. Retrieved from http://aquaponicsnation.com/articles.html/_/articles/frequently-asked-questions/aquaponics-acronyms-r15
4. Backyard Aquaponics. (2014). *Type of Systems—Backyard Aquaponics*. Retrieved from <http://www.backyardaquaponics.com/guide-to-aquaponics/running-of-the-system/>
5. Singh, M. (2014, May 24). If local farms aren't local enough, buy from the rooftop. *NPR*. Retrieved from <http://www.npr.org/blogs/thesalt/2014/05/24/313097487/if-local-farms-arent-local-enough-buy-from-the-rooftop>
6. Tortorello, M. (2010, February 18). The Spotless Garden. *The New York Times*. pp. D1 Retrieved from <http://www.nytimes.com/2010/02/18/garden/18aqua.html>
7. Donnelly, S. (Video Journalist), Duff, C. (Producer), (2010). *Aquaponics: Using Fish Poop to Grow Vegetables* [Video Tape]. United States: Time Magazine. Retrieved from http://content.time.com/time/video/player/0,32068,658160623001_2030619,00.html
8. Nate. (December 02, 2008). Controllable Power Outlet. *Sparkfun*. Retrieved from <https://www.sparkfun.com/tutorials/119>
9. Hasegawa, R. (2014) Is Aquaponics Viable?
10. *Spectra_of_chls*. Digital image. N.p., n.d. Web.
11. “リレー入門 - リレーとは。” *リレー入門 - リレーとは / 制御機器入門 / 制御機器 / 電子デバイス・産業用機器 / Panasonic*. Panasonic. Web. Retrieved from <https://www3.panasonic.biz/ac/j/corp/nyumon/relay/index.jsp>
12. “サーミスタとは？” *村田製作所* Web. Retrieved from <http://www.murata.com/ja-jp/products/sensor/temperature/basic/thermistor>
13. “I2C通信の使い方” Web. Retrieved from <http://www.picfun.com/c15.html>
14. “mraa” Intel Web. Retrieved from <https://github.com/intel-iot-devkit/mraa>
15. “D3.js—Data-Driven-Documents” Web. Retrieved from <http://d3js.org/>

Special Thanks!

竹田聖 : D3.js

Daichi Hirota: Frontend Development

Takuto Narisawa: Electronics, Embedded