

# ネットワーク(NW)百葉箱紹介 (Raspberry ZERO+ESP-WROOM-02)

## ●目次

- 0 はじめに
1. NW百葉箱システム構成
2. ホームサーバ  
システム、HW、SWについて
3. センサデバイス  
システム、HW、SWについて
4. 熱中症指数とは、絶対湿度とは
5. まとめ

# 0 はじめに

“**楽しく、安く、早く**”電子工作を心がけています。

長年、色々な物に手を出してきましたが、やっと皆さんにお勧めできるシステムが出来てきたと考えているので、紹介します。

## 色々なもの

- ラズパイ+Python+BLE4.0+Sensortag+Xively  
BLE4.0で5分おきにして**低消費電力が出来なかった**
- ESP-WROOM-02+wifi+blynk  
blynkは、スマホを接続しているときだけしか**log保管してくれない**

## ネットワーク百葉箱システム

サーバ:**Raspberry ZERO** node-red、MQTTブローカ

**複数のデバイスのlog保管**

デバイス:**ESP-WROOM-02** arduino—IDE、MQTT、wifi

BME280、**deepsleep**で、5分おきにMQTTパブリッシュ

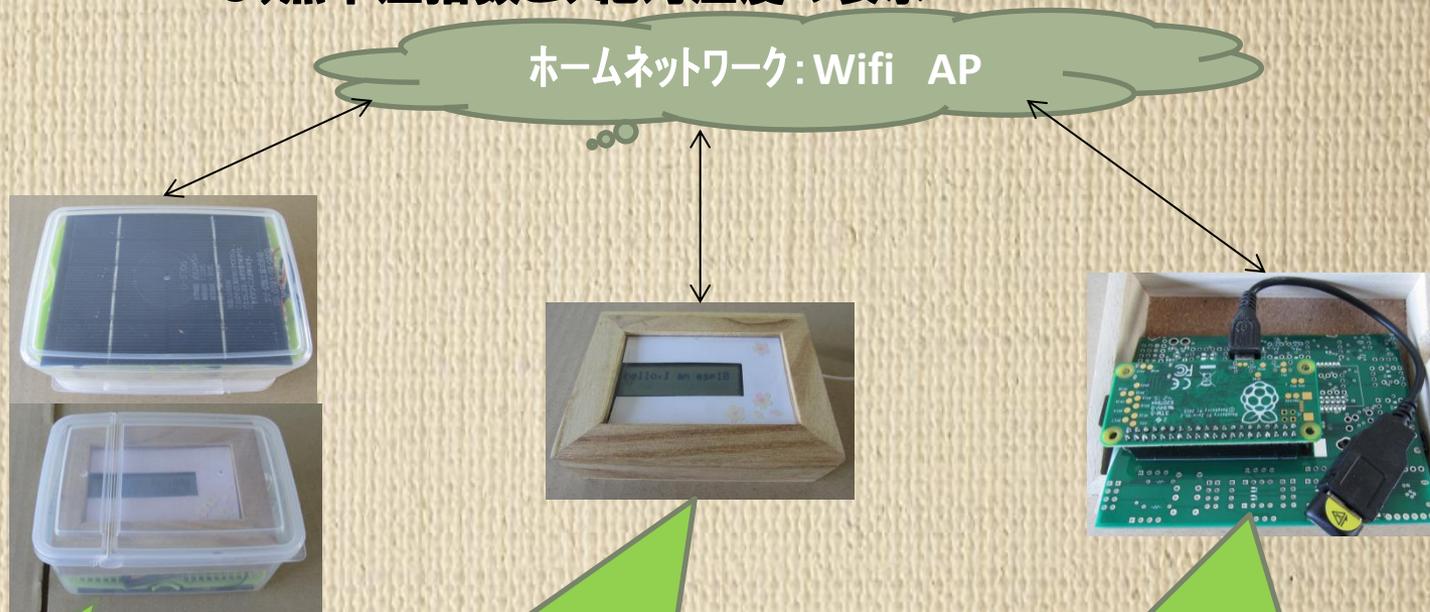
**低消費電力**(平均10mA以下)

パワポのノートに**全ソース**を入れてあります。

皆さんのお役にたてれば嬉しいです。

# 1.1 NW百葉箱システム構成

- 1) logが蓄積できる
- 2) WEBブラウザで、データが表示できる
- 3) 熱中症指数と、絶対湿度の表示



センサデバイス(屋外用)  
防水:太陽電池  
低消費電力

センサデバイス(室内用)  
ESP-WROOM-02  
Wifi、MQTT  
低消費電力

ホームサーバ  
ラズパイ ZERO  
NODE-RED  
MQTTブローカ

# 1.2 ネットワーク百葉箱システムで苦勞した点

## 課題1. 適切なサーバの選択

Xively、Blynkとクラウドサービスを試行してきましたが、なかなか自分の思うようなシステムが出来ずに困っていました。

サーバを自分で構築するには、荷が重いと考えていたのですが、ラズパイにNODE-REDがインストールされるようになり、MQTTを、割と簡単に扱えるようになり、最小限の下記機能が実現できました。

- 1) logが蓄積できること
- 2) WEBブラウザで、データが表示できること

## 課題2. システムを作るには費用が高い

ラズパイZERO(5ドル)、ESP-WROOM-02(500円)と、ここ1年ほどで、価格破壊とも思われるデバイスを手に入れ活用できるようになりました。

## 2.1 ホームサーバ構成

ホームサーバ  
ラズパイ ZERO  
NODE-RED  
MQTTブローカ

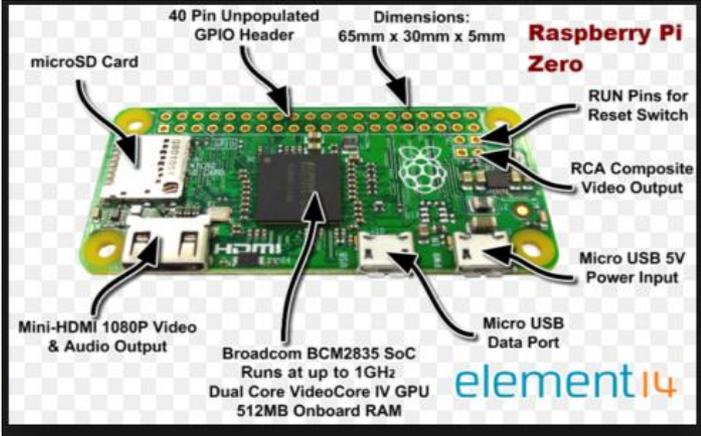
ラズパイ ZERO

マイクロSD: 8GB  
NODE-RED  
MQTTブローカ

Wifi dongle



## 2. 2ホームサーバ(HW) 部品代:2.5千円程度

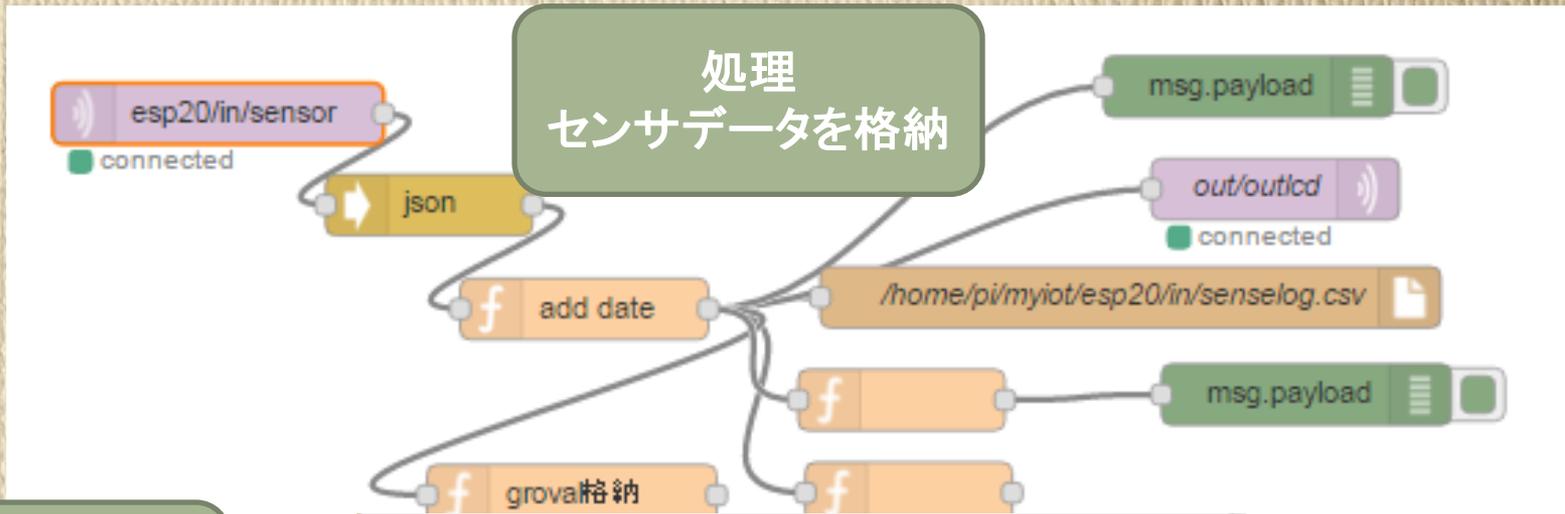
No.	品名	概要	価格	画像
1	Raspberry Pi ZERO	1Ghz, Single-core CPU 512MB RAM Mini HDMI and USB On-The-Go ports Micro USB power HAT-compatible 40-pin header Composite video and reset headers	5ドル	 <p>40 Pin Unpopulated GPIO Header Dimensions: 65mm x 30mm x 5mm Raspberry Pi Zero microSD Card RUN Pins for Reset Switch RCA Composite Video Output Micro USB 5V Power Input Micro USB Data Port Broadcom BCM2835 SoC Runs at up to 1GHz Dual Core VideoCore IV GPU 512MB Onboard RAM element14 Mini-HDMI 1080P Video &amp; Audio Output</p>
2	Wifi ドンクル WLI-UC-GNM	BUFFALO 11n対応 11g/b 無線LAN子機 親機-子機デュアルモード対応モデル WLI-UC-GNM2	1,000	
3	USBケーブル		200	
5	ユニバーサル 自作基板	I2Cの4本の信号線を2.54mmピン・ソケットで接続 Elecrow:	250	
6	プラスチックケース 100円ショップの木製ケース?	<a href="http://www.maroon.dti.ne.jp/satodenki/etc.html">W65H38D100 ケース加工済み http://www.maroon.dti.ne.jp/satodenki/etc.html</a>	100~200	

## 2. 3ホームサーバ(SW)

No.	品名	概要	Url他
1	開発環境 NODE-RED	Node-REDはハードウェアデバイス/APIおよびオンラインサービスを接続するためのツールです。	<a href="https://nodered.jp/">https://nodered.jp/</a>
2	MQTTブローカ mosquitto	MQTTとは、TCP/IPネットワークで利用できる通信プロトコルの一つで、多数の主体の間で短いメッセージを頻繁に送受信する用途に向けた軽量なプロトコル。多数のセンサーの遠隔監視などM2MネットワークやIoT(Internet of Things)分野での普及が見込まれている。	<a href="http://dotnsf.blog.jp/archives/1034832455.html">http://dotnsf.blog.jp/archives/1034832455.html</a>
3	開発言語 javascript	JavaScriptは、Webブラウザ上で動かすことのできる簡単なプログラム言語です。	<a href="http://www.atmarkit.co.jp/ait/articles/0911/18/news110.html">http://www.atmarkit.co.jp/ait/articles/0911/18/news110.html</a>

# 2.4 ホームサーバ ソフト NODE-RED

log蓄積:MQTTで送られてきたセンサデータをTEXTファイルに蓄積



入力  
MQTT:センサデータ

出力  
Logファイル出力

Edit mqtt in node

Server: 127.0.0.1:1883

Topic: esp20/in/sensor

Name: Name

Ok Cancel

Edit function node

```
1 var myfoo = global.get("foo");
2 // this should now be "bar"
3 var temp1=context.globals.temp1;
4 var humi1=context.globals.humi1;
5 var esp10=context.globals.esp10;
6 var esp20=context.globals.esp20;
7 - var getCurrentTime = function () {
8   var date = new Date();
9   date.setHours(date.getHours() + 0);
10  var d = date.getFullYear() + "-" +
11  d += ('0' + (date.getMonth() + 1)).slice(-2) + "-";
12  d += ('0' + date.getDate()).slice(-2) + "-";
13  d += ('0' + date.getHours()).slice(-2) + ":";
14  d += ('0' + date.getMinutes()).slice(-2) + ":";
15  d += ('0' + date.getSeconds()).slice(-2) + ".";
16  d += ('0' + context.globals.temp);
17 - };
18
19
20 msg.payload.timestamp = getCurrentTime();
21 msg.payload = {
22   // "timestamp" : timestamp ,
23   "esp10" : esp10 ,
24   "esp20" : esp20 ,
25   "temp1" : temp1 ,
26   "humi1" : humi1 ,
27   "avgtemp" : getAverage( temp ) ,
28   "avghumidity" : getAverage( humidity )
29 };
30
31
32 return msg;
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

Edit file node

Filename: /home/pi/myiot/esp20/in/senselog.csv

Action: append to file

Add newline (\n) to each payload?

Create directory if it doesn't exist?

Name: /home/pi/myiot/esp20/in/senselog.csv

Ok Cancel

# 2.5 ホームサーバ ソフト NODE-RED

WEBブラウザで、データ表示:log蓄積機能からグローバル変数で、データを受け取る



入力  
http:get

出力  
http:応答

Edit http in node

Method: GET

URL: /red/test

Name: /red/test

Ok Cancel

Edit function node

Name: grova格納

```
1 var temp1 = msg.payload.Sensor.Temp;
2 var humi1 = msg.payload.Sensor.Humi;
3 var esp20 = msg.payload.Sensor;
4 //var discomfort = Math.round(getDiscomfortIndex( temp , humidity ))
5
6 //グローバル変数に不快指数を格納
7 context.global.temp1 = temp1;
8 context.global.humi1 = humi1;
9 context.global.esp20 = esp20;
10 //context.global.discomfort = discomfort;
11
12 //msg.payload = context.global.temp;
13 msg.payload = {
14   //timestamp" : timestamp ,
15   "temp1" : temp1 ,
16   "humi1" : humi1 ,
17   //avgtemp" : getAverage( temp ) ,
18   //avghumidity" : getAverage( humidity )
19 };
20 return msg;
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

Edit http response node

Name: Name

The messages sent to this node must originate from an http input node

Ok Cancel

# 3.1 センサデバイス構成



センサデバイス(室内用)  
ESP-WROOM-02  
Wifi、MQTT  
低消費電力



センサデバイス(屋外用)  
防水:太陽電池  
低消費電力



## 3.2 センサデバイス(HW) 部品代:3千円

No.	品名	概要	価格	画像
1	ESP-WROOM-02	ESP8266EX(SoC、32 bit MCU)を搭載したWiFiモジュール	550 (10個以上 450:秋月 実装済み 650)	
2	BME280使用 温湿度・気圧センサモジュールキット	I2C 温度、湿度、気圧 分解能 温度: 0.01°C 湿度: 0.008% 気圧: 0.18 Pa <a href="http://akizukidenshi.com/catalog/g/gK-09421/">http://akizukidenshi.com/catalog/g/gK-09421/</a>	1080	
3	I2C接続小型キャラクタLCDモジュール AE-AQM1602A	16文字×2行のキャラクタ液晶モジュール 電源電圧は3.1~5.5V <a href="http://akizukidenshi.com/catalog/g/gK-08896/">http://akizukidenshi.com/catalog/g/gK-08896/</a>	550	
4	電源IC	高速応答のもの XC6266	200	
5	Arduino用ユニバーサル自作基板	I2Cの4本の信号線を2.54mmピン・ソケットで接続 Elecrow: 310円	250	
6	プラスチックケース 100円ショップの木製ケース?	<a href="http://www.maroon.dti.ne.jp/satodenki/etc.html">W65H38D100 ケース加工済み http://www.maroon.dti.ne.jp/satodenki/etc.html</a>	100~200	  

## 3.3 センサデバイス(SW)

No.	品名	概要	Url他
1	開発環境 Arduino ide	Arduino ide ARDUINO 1.6.5	<a href="https://www.arduino.cc/en/Main/Software">https://www.arduino.cc/en/Main/Software</a>
2	BME280使用 温湿度・気圧セン サ用 Arduinoライブラ リ	Arduinoライブラ リを実装	<a href="http://trac.switch-science.com/wiki/BME280">http://trac.switch-science.com/wiki/BME280</a>
3	I2C接続小型キャ ラクタLCDモ ジュール AE-AQM1602A用 Arduinoライブラ リ	Arduinoライブラ リを実装	<a href="http://www.geocities.jp/zattouka/GarageHouse/micon/Arduino/LCD/I2CLCD.htm">http://www.geocities.jp/zattouka/GarageHouse/micon/Arduino/LCD/I2CLCD.htm</a>
4	スケッチを少し修 正	・ライブラリをインク ルード ・スケッチを修正	<pre>#include "skI2CLCDlib.h" #define BME280_ADDRESS 0x76</pre>

## 3.4 センサデバイス: arduinoスケッチ

- `/* name:20160406-mqtt-sensor.ino`
- `author:hiroshi matsumoto rev1:2016/04/06`
- `function in:bme280(temp,humi,pres)`
- `process:temp,humi,pres,wbgt,wet`
- `out:LCD`
- `mqtt broker dev1: esp10/in/sensor`
- `mqtt broker dev2: esp20/in/sensor*/`
- `#include <Wire.h>`
- `#include <SPI.h>`
- `#include <ESP8266WiFi.h>`
- `#include <PubSubClient.h> // MQTT Client`
- `#include <ArduinoJson.h> // JSON parser`
- `#include <stdio.h>`
- `#include "ST7032.h"`

```
int sleeptime = 300; //normal=300,test=30
```

```
Serial.println("Go to deepSleep!");
```

```
ESP.deepSleep(sleeptime * 1000 * 1000 , WAKE_RF_DEFAULT);
```

```
delay(1000); // zzzzz
```

## 3.5 センサデバイスで苦労した点

---

### 課題1. ESP-WROOM-02の電源問題

- ・十分余裕のある3端子レギュレータ:不十分
  - ・大容量コンデンサ:数百マイクロでも不十分
- 高速応答タイプの3端子レギュレータ 安定

### 課題2. 低消費電力化

ESP-WROOM-02のdeepsleepモードで解決  
ただし、Resetと同じ条件で復帰するため、毎回  
初期設定から始まる(前回の状況を記憶できない)  
間欠動作となるため、連続使用のセンサ(雨量計)  
には使いにくい

### 課題3. 熱中症指数と、絶対湿度の表示

豊富なメモリのある、ESP-WROOM-02のお蔭で、  
何とか計算することが出来た

# 4.1 熱中症指数とは

## ・暑さ指数

(WBGT: Wet Bulb Globe Temperature)とは

WBGT(湿球黒球温度)とは、人体の熱収支に影響の大きい湿度、輻射熱、気温の3つを取り入れた指標で、乾球温度、湿球温度、黒球温度の値を使って計算します。

表4・WBGT値と気温、相対湿度との関係

WBGT値	相対湿度(%)																		
	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100		
40	29	30	31	32	33	34	35	35	36	37	38	39	40	41	42	43	44		
39	28	29	30	31	32	33	34	35	35	36	37	38	39	40	41	42	43		
38	28	28	29	30	31	32	33	34	35	35	36	37	38	39	40	41	42		
37	27	28	29	29	30	31	32	33	34	35	35	36	37	38	39	40	41		
36	26	27	28	29	29	30	31	32	33	34	34	35	36	37	38	39	39		
35	25	26	27	28	29	29	30	31	32	33	33	34	35	36	37	38	38		
34	25	25	26	27	28	29	29	30	31	32	33	33	34	35	36	37	37		
33	24	25	25	26	27	28	28	29	30	31	32	32	33	34	35	35	36		
32	23	24	25	25	26	27	28	28	29	30	31	31	32	33	34	34	35		
31	22	23	24	24	25	26	27	27	28	29	30	30	31	32	33	33	34		
30	21	22	23	24	24	25	26	27	27	28	29	29	30	31	32	32	33		
29	21	21	22	23	24	24	25	26	27	28	29	29	30	31	31	32	32		
28	20	21	21	22	23	23	24	25	26	27	28	28	29	30	30	31	31		
27	19	20	21	21	22	23	23	24	25	26	27	27	28	29	29	30	30		
26	18	19	20	20	21	22	22	23	24	24	25	26	26	27	28	28	29		
25	18	18	19	20	21	22	22	23	23	24	25	25	26	27	27	28	28		
24	17	18	18	19	19	20	21	21	22	22	23	24	24	25	26	26	27		
23	16	17	17	18	19	19	20	20	21	22	22	23	23	24	25	25	26		
22	15	16	17	17	18	18	19	19	20	21	21	22	22	23	24	24	25		
21	15	15	16	16	17	17	18	19	19	20	20	21	21	22	23	23	24		

WBGT値 注意 25℃未満 警戒 25℃～28℃ 厳重警戒 28℃～31℃ 危険 31℃以上

(日本生気象学会「日常生活における熱中症予防指針」Ver.1 2008.4 から)

### 【日常生活における熱中症予防指針】

温度基準(WBGT)	注意すべき生活活動の目安	注意事項
危険(31℃以上)	すべての生活活動でおこる危険性	高齢者においては安静状態でも発生する危険性が大きい。外出はなるべく避け、涼しい室内に移動する。
厳重警戒(28～31℃)		外出時は炎天下を避け、室内では室温の上昇に注意する。
警戒(25～28℃)	中等度以上の生活活動でおこる危険性	運動や激しい作業をする際は定期的に十分に休息を取り入れる。
注意(25℃未満)	強い生活活動でおこる危険性	一般的には危険性は少ないが、激しい運動や重労働時には発生する危険性がある。

注) 28～31℃は28℃以上31℃未満の意味 (日本生気象学会「日常生活における熱中症予防指針」より)

### ※WBGT(湿球黒球温度)の算出方法

屋外:WBGT = 0.7×湿球温度+0.2×黒球温度+0.1×乾球温度

屋内:WBGT = 0.7×湿球温度+0.3×黒球温度

室内に限れば、気温と湿度から、右の図のように求められます



## 4.2 熱中症指数(wbgt)の求め方

Tableから求める方法は2種類考えられます。

1) tableをデータベースとして扱い、検索する方法

2次元配列にして、温度、湿度を整数化(コード化)すれば...

2) 近似式を求め、計算で求める方法

excelで何とか実用レベルの近似式を求めました。

(多変量の測定データに対して重回帰分析)

excelのソルバーではよい結果が得られず、品質工学専門家にやってもらいました。

$Y_i * \text{気温} + X_i * \text{湿度} + YX_i * \text{気温} * \text{湿度}$

```
double wbgt = 0.0;
```

```
double temp_i = 30.5, humi = 60.0;
```

```
double Y_i = 0.90739, X_i = 0.14775;
```

```
double YX_i = -0.003665, aveall = 27.77;
```

```
wbgt = (temp_act - temp_i) * Y_i;
```

```
wbgt = wbgt + (hum_act - humi) * X_i;
```

```
wbgt = wbgt + (temp_act - temp_i) * (hum_act - humi) * YX_i + aveall;
```

## 4.3 絶対湿度とは

### ●乾燥指数(絶対湿度)とは

空気の体積1立方メートルあたりに含まれる水蒸気の質量を表す指数です。季節性インフルエンザの流行に関して、乾燥指数(絶対湿度)が11g/m<sup>3</sup>以下では「注意」、7g/m<sup>3</sup>以下になると「警戒」となり、どちらも空気が乾燥した状態で適度な加湿を勧めています。

#### 【空気の乾燥状態とインフルエンザの流行】

インフルエンザ対策の目安	湿度基準 絶対湿度(乾燥指数※)	空気の乾燥状態	空気の乾燥状態とインフルエンザの流行
警戒	7g/m <sup>3</sup> 以下	乾燥	空気が特に乾燥して、インフルエンザが流行しやすい状態(インフルエンザの流行に適した湿度)
注意	7g/m <sup>3</sup> を超えて11g/m <sup>3</sup> 以下	やや乾燥	空気が乾燥してきて、インフルエンザが流行してよい状態
ほぼ安全	11g/m <sup>3</sup> を超えて17g/m <sup>3</sup> 以下	湿潤	空気が湿っていて、インフルエンザの流行はしにくい状況
	17g/m <sup>3</sup> を超える	非常に湿潤	空気が大変湿っていて、インフルエンザの流行は非常にしにくい状況

財団法人宮城県地域医療情報センターホームページ:全国インフルエンザ流行予測

<http://www.mmic.or.jp/flu/flu-list.php>より転記、一部追記。(※勝エー・アンド・デイの製品において、空気の乾燥状態を表す絶対湿度を「乾燥指数」として表しています。この「乾燥指数」は、ド=マルトンヌの提唱する「乾燥度指数」とは定義と意味が異なります。)

20.4C 35.0% 1013  
0 wb=14 × w= 6.2

## 4.4 温度と湿度から水蒸気量の求め方

//(1) 温度tから飽和水蒸気圧eを求める

```
double e-wet = 0.0 ,a-wet = 0.0;
```

```
e-wet= 7.5 * tempi/(tempi+ 237.3)
```

```
e-wet= 6.11 × pow( 10,e-wet)
```

//(2) eとtと相対湿度rhから水蒸気量a(グラム/立方m)を求める

```
a-wet=217.0 × e-wet / (tempi+273.15) x (humi/100.0)
```

# 5. まとめ

---

## ・早く

ホームサーバ: NODE-RED、MQTT

**素早く開発、ライブラリが豊富**

センサデバイス: arduino-IDE、MQTT

**素早く開発、ライブラリが豊富**

## ・安く

ホームサーバ部品代: ラズパイZERO      2.5千円程度

センサデバイス部品代: ESP-WROOM-02      3千円程度

## ・楽しく

MQTTという汎用のプロトコルを使っているので

IN/OUTの色々なデバイスをつなぎやすい

実用的な、高い性能のものができる

**“楽しく、安く、早く”電子工作が出来ました、  
参考にしてもらえると嬉しいです**

# おまけ

## Makerfaire2016に出展したい

- この“ネットワーク(NW)百葉箱”では、うけないかな???
- 前回のモールス練習機を改良してできないかな  
出来れば、モールスENCODER、decoderも実装したいけど・・・
- 今まで、PICのモールス練習機はたくさんあるので、流用したいが、WIFIで、MQTTに送れるかな？PICだと大変かな???
- やはり、ESP-WROOM-02で、arduinoのソフトを探した方が早いかな???

“楽しく、安く、早く“電子工作を心がけています。

今年、JARLでは、「モールス符号を無形文化遺産へ！」というキャンペーンをしています。

Makerとして、協力するべく、ユニークな器械を作りました。

### ラズパイ・無線モールス通信器

Raspberry ZERO2台で対向 node-red、MQTTブローカ

Raspberry ZEROというわずか5ドルの部品と

GUIの開発環境ですので、小中学生から作れます。

うまくすれば、インターネットでモールスが復活するかもしれません。