

ラズパイ百葉箱 と クラウド(ambient)でグラフ表示 と ラズパイ実験ボード(アナログ入力) 紹介

●目次

- 0 はじめに
1. ラズパイ百葉箱システム構成
2. 温度・湿度・気圧センサ
3. 風速、風向、雨量センサ
4. システム管理、グラフ表示
5. 熱中症指数とは、絶対湿度とは
6. ラズパイ実験ボード(アナログ入力)
7. 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。

ラズパイ百葉箱

1. 構成

- サーバ兼デバイス:ラズパイ3、1台 **NODE-RED+MQTTブローカ**
風速、風向、雨量計
- デバイス:ESP-WROOM-02 2台(室外、室内)
BME280(**温度、湿度、気圧センサ**)
熱中症指数(WBGT)、インフルエンザ指数(絶対湿度)は、
温度、湿度より計算

2. 機能 クラウドのambientでグラフ表示

- 室外 温度、湿度、気圧、WBGT、絶対湿度**
↓←エアコンで制御→↓ ↓←加湿器で制御
- 室内 温度、湿度、気圧、WBGT、絶対湿度**

3. やりたかったこと

- 夏、室外の熱中症指数(WBGT)を見て、エアコンを制御し、**
快適な室内WBGTを作りだす
- 冬、室外のインフルエンザ指数(絶対湿度)を見て、加湿器を制御し、**
快適な室内絶対湿度を作りだす

皆さんのお役にたてれば嬉しいです。

1. ラズパイ百葉箱システム構成

- 1) 複数のホームサーバ兼デバイスが対等の関係で接続
- 2) PC、スマホの、WEBブラウザで、データが表示できる
- 3) cloudに接続

ホームネットワーク: Wifi AP

ホームサーバ間は、MQTT

Cloudに接続

GUIは、PC、スマホの、ブラウザ

ラズパイ百葉箱システム: 監視

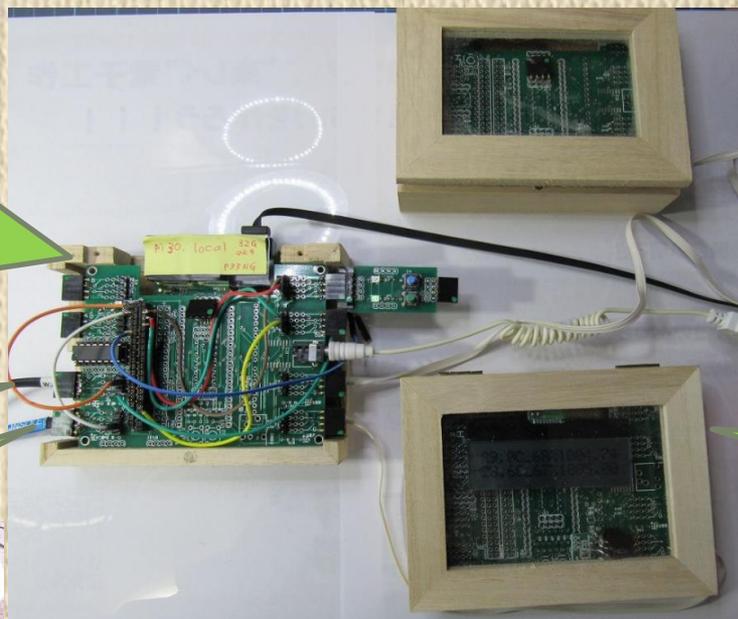
デバイス

ホームサーバ
GW, 兼デバイス
ラズパイ
NODE-RED
MQTTブローカ

風速・
風向計



雨量計



センサデバイス(屋外用)
防水: 太陽電池
低消費電力

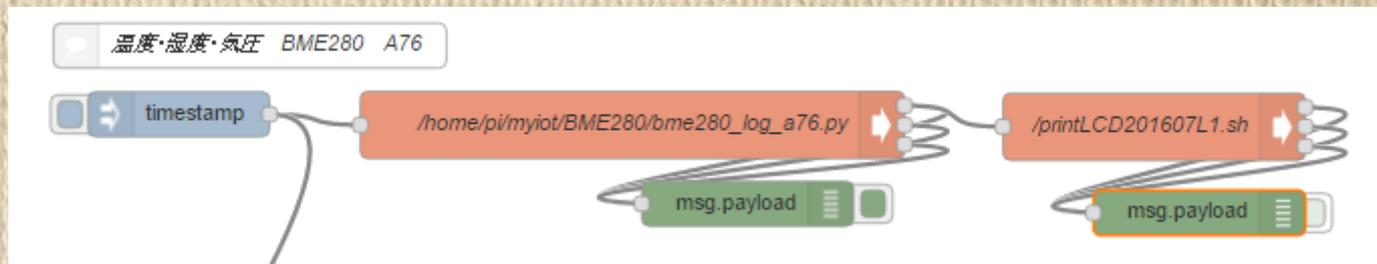


デバイス
I2C接続
屋内用: 表示付き
屋外用



2. 温度・湿度・気圧センサ

- ・フロー(流れ)とフロー間の情報は、グローバル変数
- ・変数を初期化、入力をカウント、雨量は、0.2794mm/カウントで計算



This is a screenshot of the 'Edit inject node' configuration window. The 'Payload' field is set to 'timestamp'. The 'Repeat' section is set to 'interval' with 'every 1 minutes'. There are 'Ok' and 'Cancel' buttons at the bottom.

定期起動
1分:自由に設定可

This is a screenshot of the 'Edit exec node' configuration window. The 'Command' field contains `sudo python /home/pi/myiot/BME280/bme280_log_a76.py`. The 'Append' section has next to 'msg.payload'. The 'Name' field contains `/home/pi/myiot/BME280/bme280_log_a76.py`. There are 'Ok' and 'Cancel' buttons at the bottom.

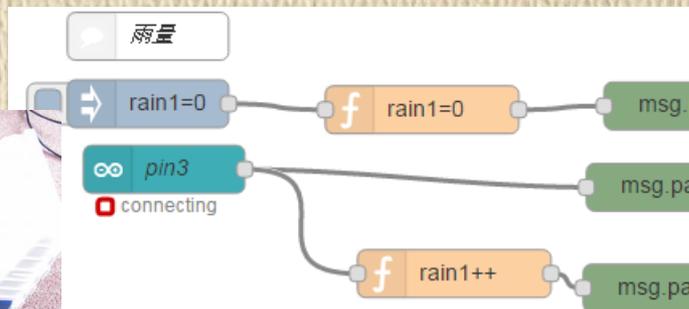
BME280のサンプルプログラムの、プリント文を数行修正

This is a screenshot of the 'Edit exec node' configuration window. The 'Command' field contains `sudo /home/pi/myiot/LCD/printLCD201607L1.sh`. The 'Append' section has next to 'msg.payload'. The 'Name' field contains `/printLCD201607L1.sh`. There are 'Ok' and 'Cancel' buttons at the bottom.

ApplePiのLCDプログラムを、1行目用と、2行目用に修正

3.1 雨量計

- フロー(流れ)とフロー間の情報は、グローバル変数
- 変数を初期化、入力をカウント、雨量は、0.2794mm/カウントで計算



Edit inject node

Payload: rain1=0

Topic:

Repeat: none

Inject once at start?

Name:

Note: "Interval between times" and "at a specific time" will use cron. See info box for details.

Ok Cancel

Edit function node

Name: rain1=0

Function

```
1 //グローバル変数に格納
2 context.global.rain1 = 0;
3 return msg;
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

Edit arduino in node

Arduino: /dev/ttyUSB0

Type: Digital pin

Pin: 3

Name: pin3

Note: You cannot use the same pin for both output and input.

Ok Cancel

Edit function node

Name: rain1++

Function

```
1 var rain = context.global.rain1;
2 //= parseInt(msg.payload.Sensor.rain);
3
4 // sum = context.global.rain
5 // 定義した入力数 cameたら、集計処理を行い、msg.payloadを返
6 if (msg.payload == true)
7 { // msg.payloadの合計値を算出
8   // 雨量をカウントする
9   // msg.payload = parseInt(msg.payload) + 1;
10  rain = rain + 0.2794;
11  context.global.rain1 = rain;
12  msg.payload = rain;
13
14  return msg;
15 }
16 return msg;
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

3.2 風速計

- 変数を初期化、入力をカウント、
- 単位時間(仮置き:5分)当たりのカウントから風速を計算(仮置き:1.0m/カウント)



Edit inject node

Payload: timestamp
Topic:
Repeat: interval
every 5 seconds
Inject once at start?
Name: 1min

Edit function node (wind_value)

```
1 //スタート時に初期化、初期値は 0
2 context.global.wind1 = context.global.wind1 || 0;
3 context.global.wind_value = context.global.wind_value || 0;
4 //グローバル変数に格納
5 context.global.wind_value = context.global.wind1 * 1.01;
6 msg.payload = context.global.wind_value;
7 //msg.payload = context.global.wind_value + context.global.wind_dir_var;
8 return msg;
```

Edit function node (wind1++)

```
1 //スタート時に初期化、初期値は 0
2 var wind = context.global.wind1 || 0;
3 // = parseInt(msg.payload.Sensor.rain);
4
5 // sum = context.global.rain
6 // 定義した入力数が来たら、集計処理を行い、msg.payloadを返す
7 if (msg.payload == true)
8 - { // msg.payloadの合計値を算出
9     // 雨量をカウントする
10    msg.payload = parseInt(msg.payload) + 1;
11    wind = wind + 1;
12    context.global.wind1 = wind;
13    msg.payload = wind;
14
15 }
16 - }
17 return msg;
```

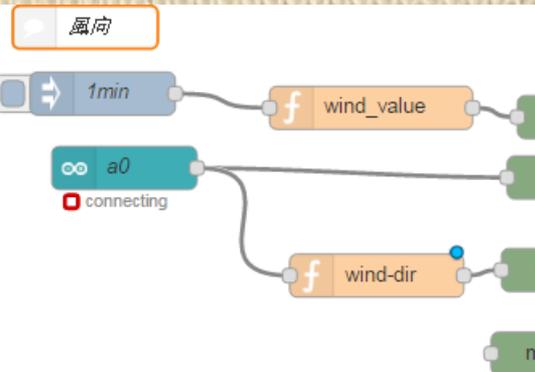
Edit arduino in node

Arduino: /dev/ttyUSB0
Type: Digital pin
Pin: 2
Name: pin2

Note: You cannot use the same pin for both output and input.

3.3 風向計

風向きごとに抵抗値が変わる



Edit function node

Name: wind-dir

Function

```
1 //スタート時に初期化、初期値は
2 var wind_max = context.global.wind_dir_max || 0;
3 var wind_min = context.global.wind_dir_min || 2000;
4 var wind_var = context.global.wind_dir_var || 0;
5 var wind_range = wind_max - wind_min;
6 // = parseInt(msg.payload.Sensor.rain);
7 var wind_count = parseInt(msg.payload);
8 //msg.payload = parseInt(wind_max);
9 //msg.payload = parseInt(wind_min);
10 // sum = context.global.rain
11 // 定義した入力数 cameたら、集計処理を行い、msg.payloadを返す
12 if (wind_count > wind_max)
13 - { // msg.payloadの合計値を算出
14   wind_max = wind_count;
15   context.global.wind_dir_max = wind_max;
16   //msg.payload = parseInt(wind_max);
17 }
18 - }
19 if (wind_count < wind_min)
20 - { // msg.payloadの合計値を算出
21   wind_min = wind_count;
22   context.global.wind_dir_min = wind_min;
23   //msg.payload = parseInt(wind_min);
24 }
25 - }
26 wind_var = wind_count - wind_min;
27 wind_var = wind_var / (wind_max - wind_min);
28 wind_var = wind_var * 360;
29 context.global.wind_dir_var = wind_var;
30 msg.payload = parseInt(wind_var);
31
32 return msg;
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

Edit inject node

Payload: timestamp

Topic:

Repeat: interval

every 5 seconds

Inject once at start?

Name: 1min

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

Ok Cancel

Edit function node

Name: wind_value

Function

```
1 //グローバル変数に格納
2 msg.payload = context.global.wind_dir_var;
3 return msg;
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

Edit arduino in node

Arduino: /dev/ttyUSB0

Type: Analogue pin

Pin: 0

Name: a0

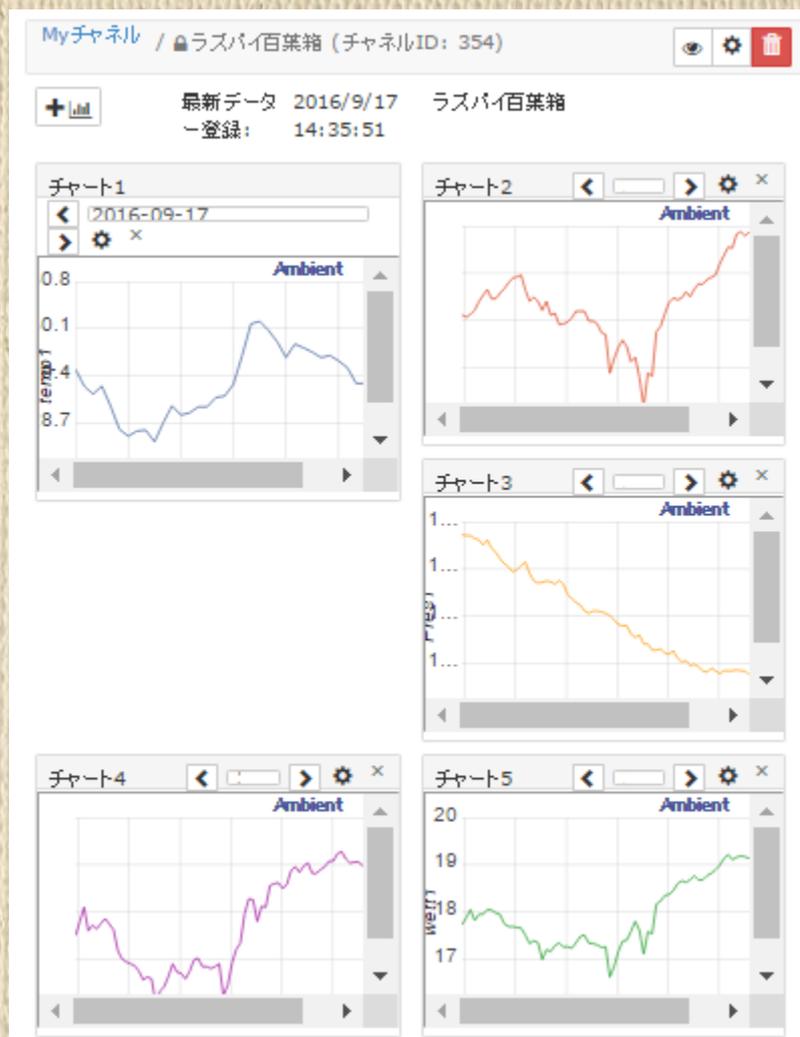
Note: You cannot use the same pin for both output and input.

Ok Cancel

入力の最大値、最小値を求め、その幅で正規化し、風向き(0~360°)を求める

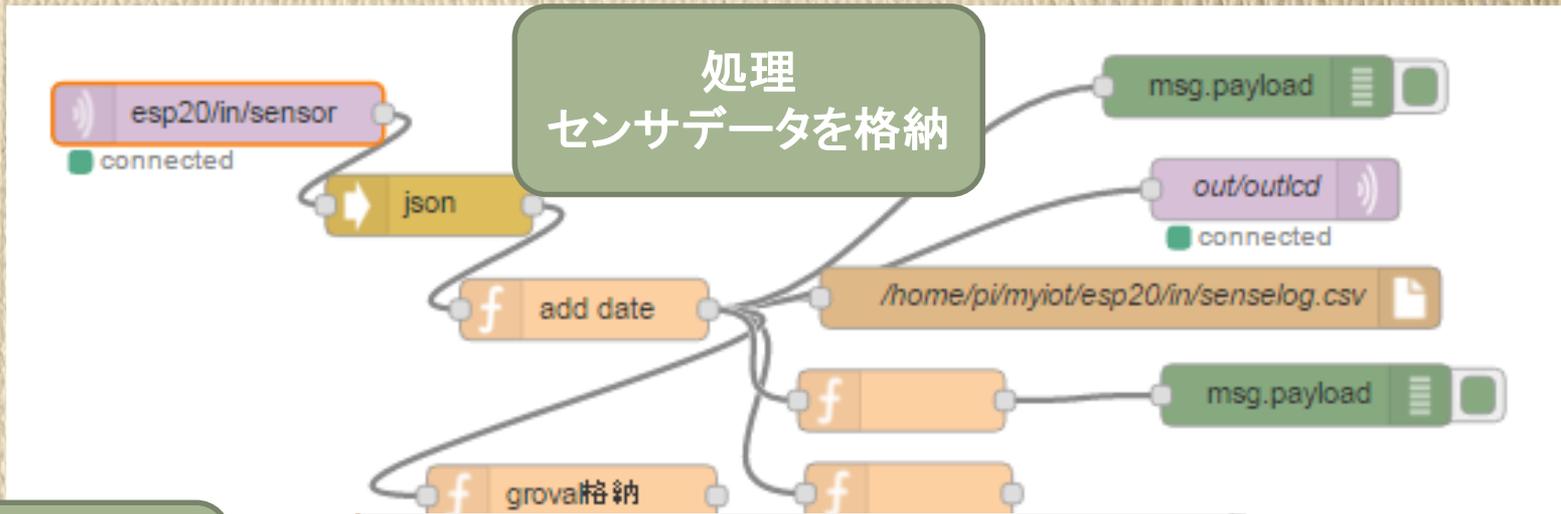
4.1 グラフ表示

クラウド・サービス、ambientでグラフ表示



4.2 システム管理 (Log保管)

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.global.temp1;
4 var humi1=context.global.humi1;
5 var esp10=context.global.esp10;
6 var esp20=context.global.esp20;
7 - var
8   getCurrentTime = function () {
9     var date = new Date();
10    date.setHours(date.getHours() + 0);
11    var d = date.getFullYear() + "-" +
12    d += ("0" + (date.getMonth() + 1)).slice(-2) + "-";
13    d += ("0" + date.getDate()).slice(-2) + "-";
14    d += ("0" + date.getHours()).slice(-2) + ":";
15    d += ("0" + date.getMinutes()).slice(-2) + ":";
16    d += ("0" + date.getSeconds()).slice(-2) + ".";
17    return d;
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

5.1 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	27	28	28	29	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



5.2 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 tempi = 30.5, humi = 60.0;
```

```
double Yi = 0.90739, Xi = 0.14775;
```

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

```
wbgt = (temp_act - tempi) * Yi;
```

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

```
wbgt = wbgt + (temp_act - tempi) * (hum_act - humi) * YXi + aveall;
```

5.3 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/m³以下では「注意」、7g/m³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 ³ 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 ³ を超えて11g/m ³ 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 ³ を超えて17g/m ³ 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 ³ 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。(※關エー・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 X W= 6.2

5.4 絶対湿度の求め方

//(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)
```

6.1 ラズパイ実験ボード(アナログ入力)ブロック図

シャットダウンスイッチ

ラズパイ
ZERO、2、3
GPIO: 40pin

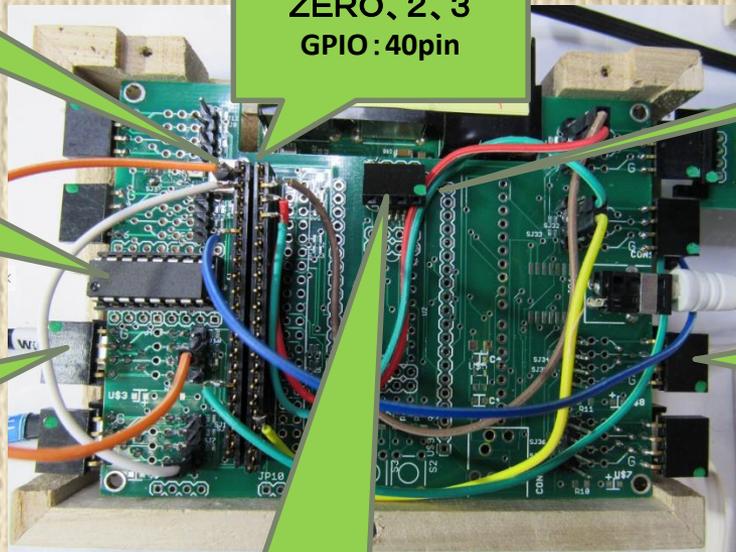
I2Cリピータ

アナログ入力
8入力:
• MCP3208-CI/P (12bit
8ch A/D)

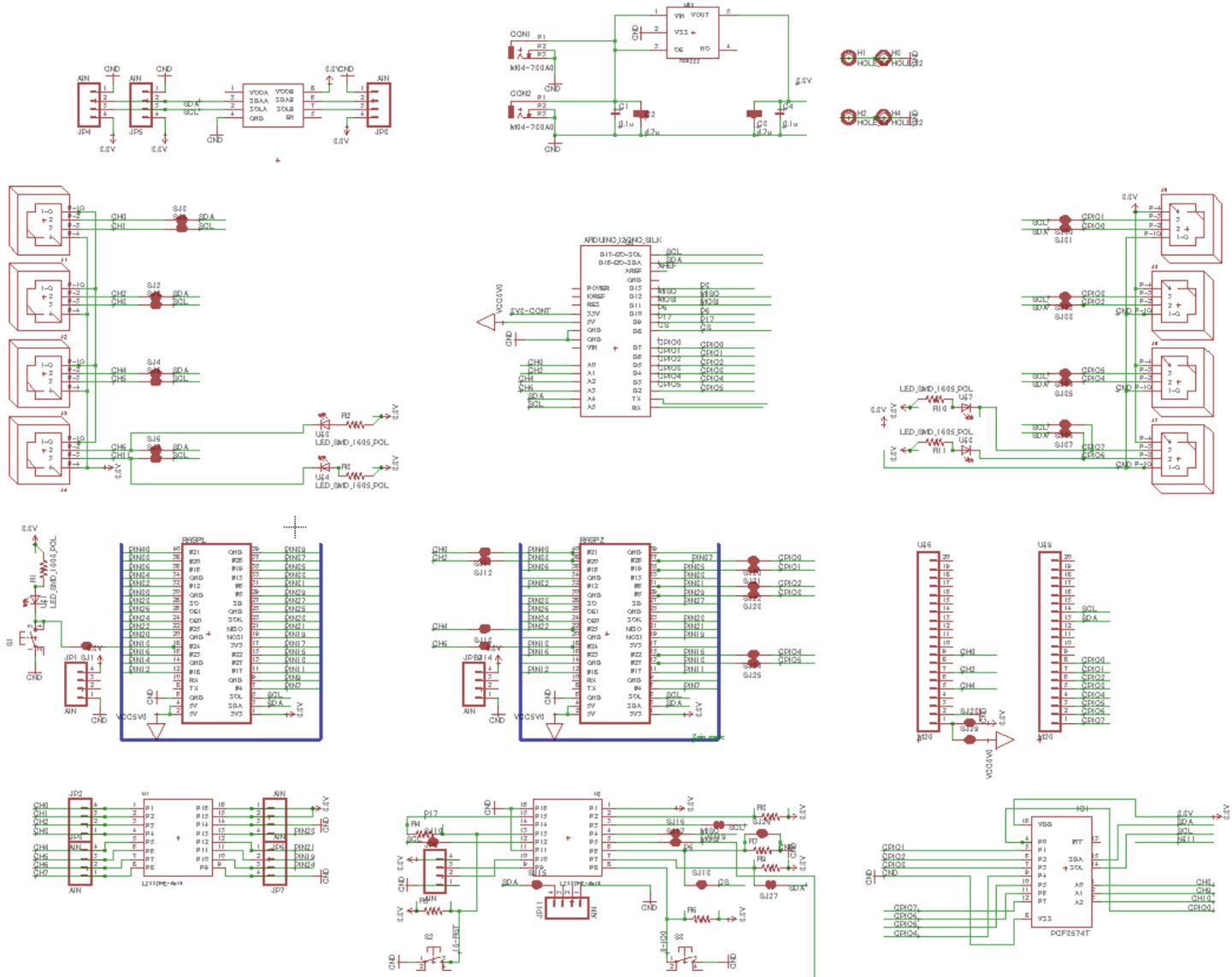
アナログ入力 兼
汎用GPIO入出力
コネクタ
4Pinコネクタ: 4ヶ

汎用GPIO入出力 兼
I2C コネクタ
4Pinコネクタ: 4ヶ

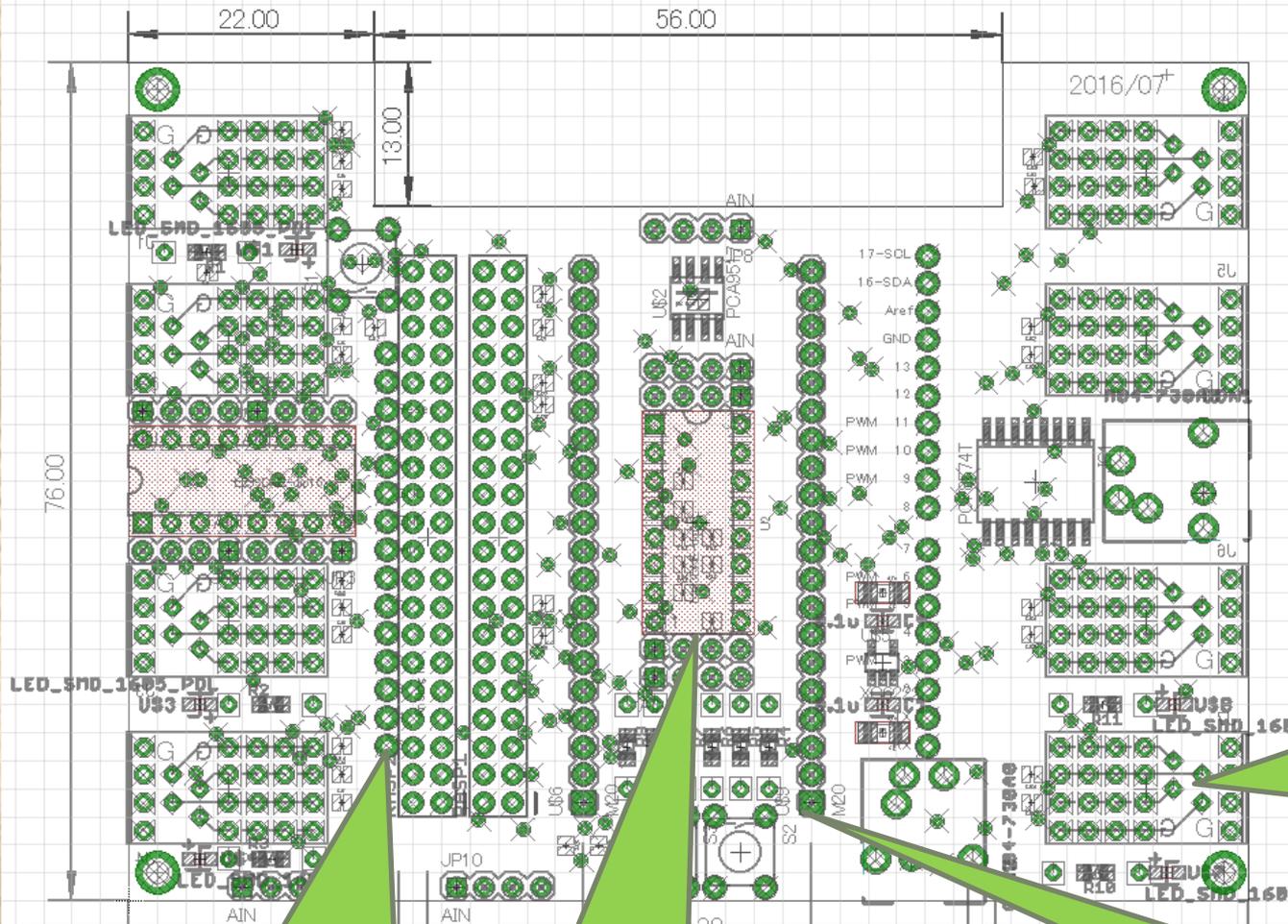
I2C LCD用 コネクタ
4Pinコネクタ: 1ヶ



6.2 ラズパイ実験ボード(アナログ入力)回路図



6.3 ラズパイ実験ボード(アナログ入力)基板図



100均の木製ケースに入るように設計しています。
76*100mm

4pinのモジュラーコネクタケーブルは100均で売っていますが実装できます

arduinoの標準コネクタが実装できます

秋月のESP-WROOM-02が実装できます

PSOCプロトタイプ基板が実装できます

7. 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: ラズパイ ZERO、2、3

センサデバイス部品: ESP-WROOM-02 (arduinoIDEで開発可)

・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た

**“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**