

AI/IoTセンサのしくみを知ろう(基礎編)

第4回温度・湿度をはかる

桑折 範彦
2019 /06/01

今日のテーマ

- 前回の復習: JJ7ボード上のLEDを点滅させる
- LEDを点滅をさせる
前回のLEDのスケッチ Example202aを復習
- 温度・湿度センサ HTU21D 取り付ける
スケッチにより測定する
値の表示方法(シリアルプリント、シリアルプロット)
測定値の検討
- 温度センサ DS18B20 取り付ける
スケッチにより測定する
値の表示方法(シリアルプリント、シリアルプロット)
測定値の検討

プログラミングの復習

前回基本を学んだ

Setup() スケッチの初期設定部分

Loop() プログラムの本体部分

繰り返し: for文{ }繰り返しの構文

条件判定: if(条件式){ 1 } else { 2 }

条件式が成立していれば{1}の部分を実行、そうでなければ{2}の部分を実行する

関数の定義: プログラムの一部をまとまりとして

呼び出ししやすくする、プログラムを分かりやすくする

配列の定義: 複数の変数をまとめて1つの変数名とする

スケッチの復習 LEDの点灯・消灯 少し拡張する

変数の表し方 関数の中のfor文の表現

// Example202a

```
const int LED_PIN[2]={2,18};      //LED 2:Blue, LED 18:Green
```

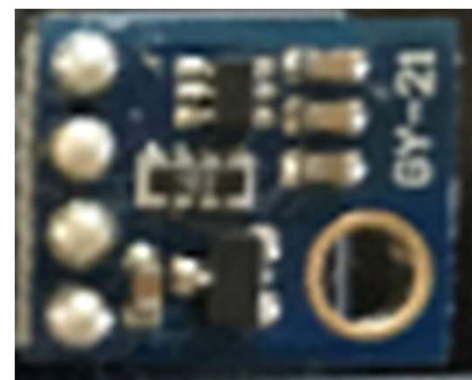
```
void setup() {  
  pinMode(LED_PIN[0], OUTPUT);    // LED_PIN[0]=2は出力 青  
  pinMode(LED_PIN[1], OUTPUT);    // LED_PIN[1]=18は出力 緑  
}
```

```
void loop() {  
  myLed(0);                       //LEDの番号0に出力 青  
  myLed(1);                       //LEDの番号1に出力 緑  
}
```

```
void myLed(int ledNum){           //関数ledNumを指定して実行  
  for(int i=1; i>=1; i--){       // i=1にして以下を実行、i=i-1つまりi=0にして以下を実行  
    digitalWrite(LED_PIN[ledNum], i); //引数ledNumのLED_PINにiを書く。  
    delay(1000);                 //i=1なら点灯、i=0なら消灯 //1秒待つ  
  }  
}
```

温度・湿度センサ

- 温度・湿度センサ(小さなボードGY-21)はJ7を上にして右の方に、取り付けてある。
- 温度・湿度センサ:
Te Connectivity, HTU21D
小さなボードGY-21



温度センサ

- Maxim, DS18B20 取り付ける
- ケーブルになっている
- コネクタの向きに注意
印字: RE(赤) BL(黒)
YE(黄)がコネクタの線の色に合うように挿す

DS18B20

防水温度センサ

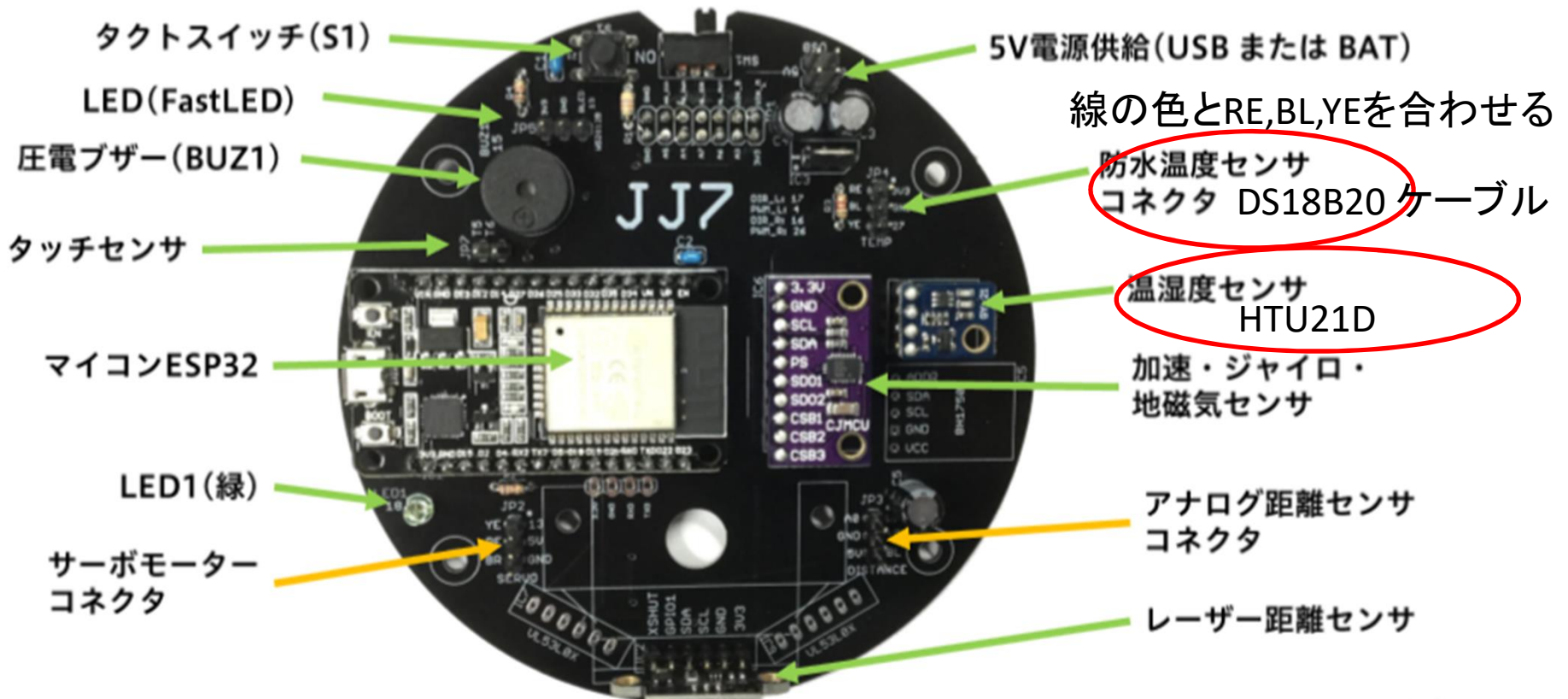
(ステンレス容器封入タイプ)

- 高精度計測
- デジタル出力
- 複数並列接続OK
- データラインから給電可



センサーの取付け

マイコンボード(JJ7)



温度と湿度をはかる

- 温度によって空気 1m^3 が含むことのできる水蒸気は決まっている
 - 温度が低い場合は水蒸気は少ない
 - 温度が高い場合は多い
- 飽和水蒸気 露点
- グラフを参照

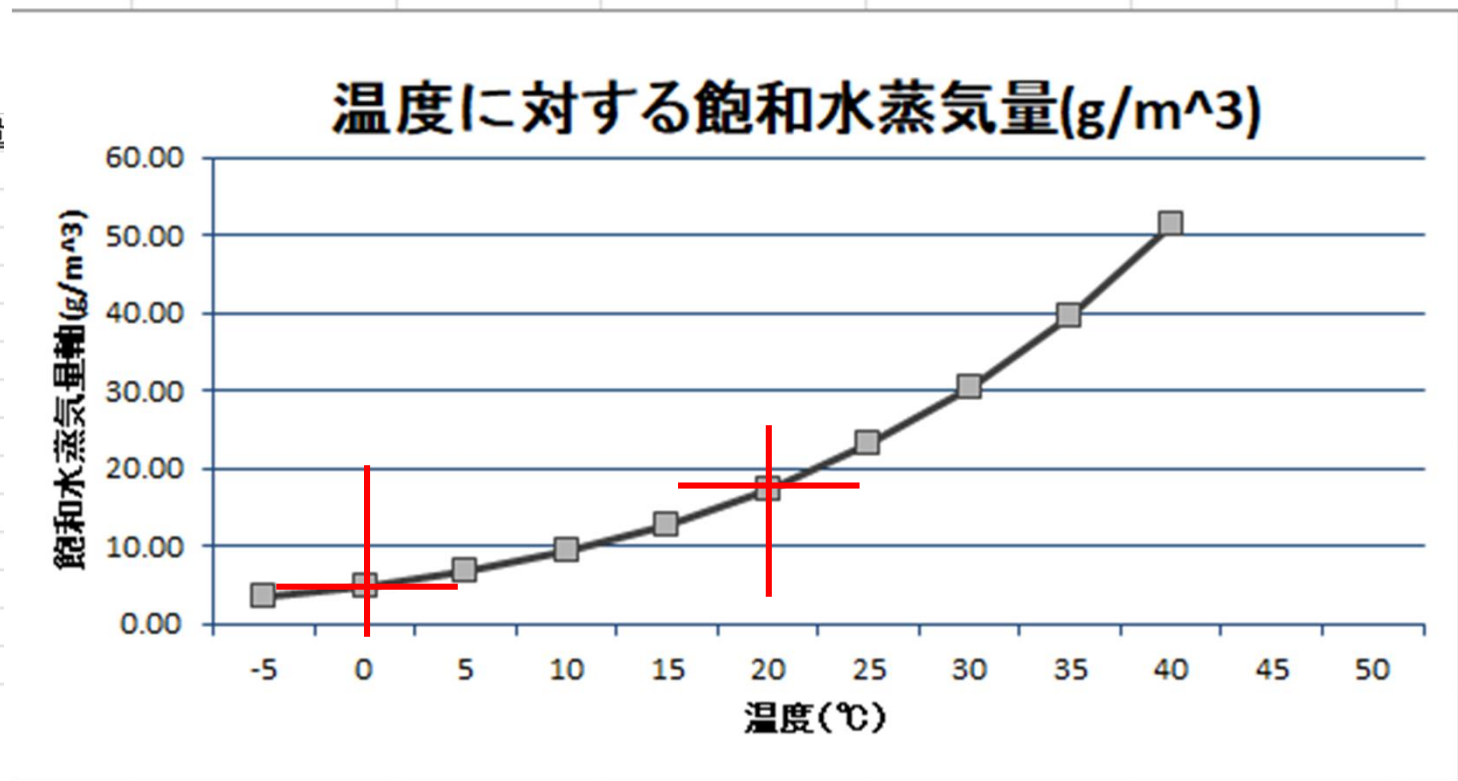
空気の温度と湿度

空気 1m^3 がその時の温度(T)で含むことのできる水蒸気量(飽和水蒸気量、 g/m^3)に対して含まれている水蒸気量 g/m^3 の割合を(相対)湿度という(%RH)。

飽和水蒸気量はTetensの式を用いて計算できる(後述)。

20度では $17.32\text{g}/\text{m}^3$ の水蒸気を含む、そのまま0度にとすると $4.85\text{g}/\text{m}^3$ しか含めない。 $17.32 - 4.85 = 12.47\text{g}/\text{m}^3$ が水滴になる。

温度に対する飽和水蒸気量	
Tetensの式による飽和水蒸気量	
室温 T	$T_{\text{sat}} (\text{g}/\text{m}^3)$
-5	3.41
0	4.85
5	6.81
10	9.41
15	12.85
20	17.32
25	23.08
30	30.40
35	39.64
40	51.17
45	
50	



温度・湿度の測定

HTU21Dの利用

既に接続されているセンサーHTU21Dを利用してみる。
家庭でも温度と湿度を測定したり、室内の湿度を気にしたりする。

ここでは、HTU21Dを利用して、温度と湿度の関係などを考えてみよう。

測定した値を数値でみる方法

ツールのメニューにあるシリアルモニタを使う。

測定値をグラフにする方法

ツールのメニューにあるシリアルプロッタを使う。

HTU21D スケッチの説明

配布されたExample03の中にHTU21Dがあるので、ファイル→開くからこれを開く

```
// htu21d.ino
#include <WiFi.h>
#include "SparkFunHTU21D.h"
```

```
HTU21D htu21d;
```

```
void setup() {
// WiFi.mode(WIFI_OFF);
  Serial.begin(115200);
  delay(100);
  htu21d.begin();
}
```

```
// はコメント文でメモなどを書く
//必要なヘッダーを読み込む
```

```
Setup()は1度実行される
コメントにする(WIFIはoffにする)
通信速度を115200とする
100ms待つ
Htu21dの実行開始
```

```
void loop() {  
  float humd = htu21d.readHumidity();  
  float temp = htu21d.readTemperature();  
  
  Serial.print(temp, 2);  
  Serial.print(", ");  
  Serial.println(humd, 2);  
  
  // Serial.print("Time:");  
  // Serial.print(millis());  
  // Serial.print(" Temperature:");  
  // Serial.print(temp, 1);  
  // Serial.print("C");  
  // Serial.print(" Humidity:");  
  // Serial.print(humd, 1);  
  // Serial.print("%RH");  
  // Serial.println();  
  delay(100);  
}
```

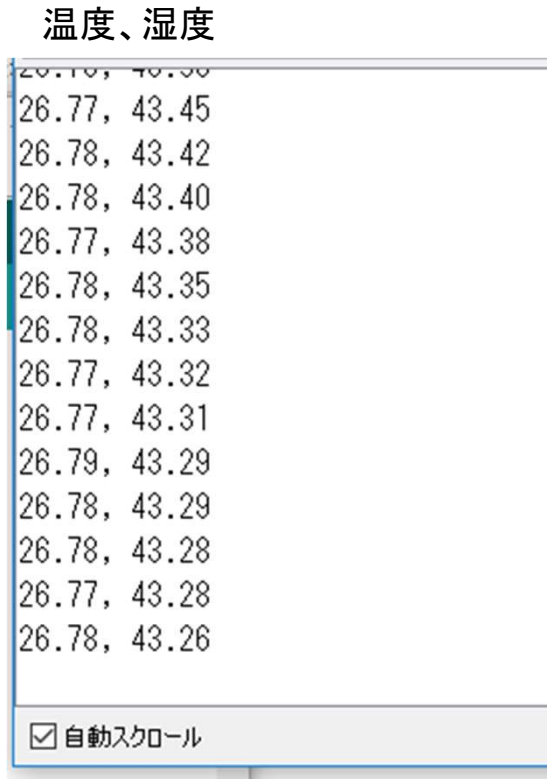
loopの{から終わりの}までを繰り返す
変数の定義 humd, temp

Serial.printは出力
(変数名,少数点以下の数値)
表示はツールで選択
シリアルモニタに表示される
シリアルプロッタにグラフ表示
表示の時の通信速度を
115200とする
//の行はコメント

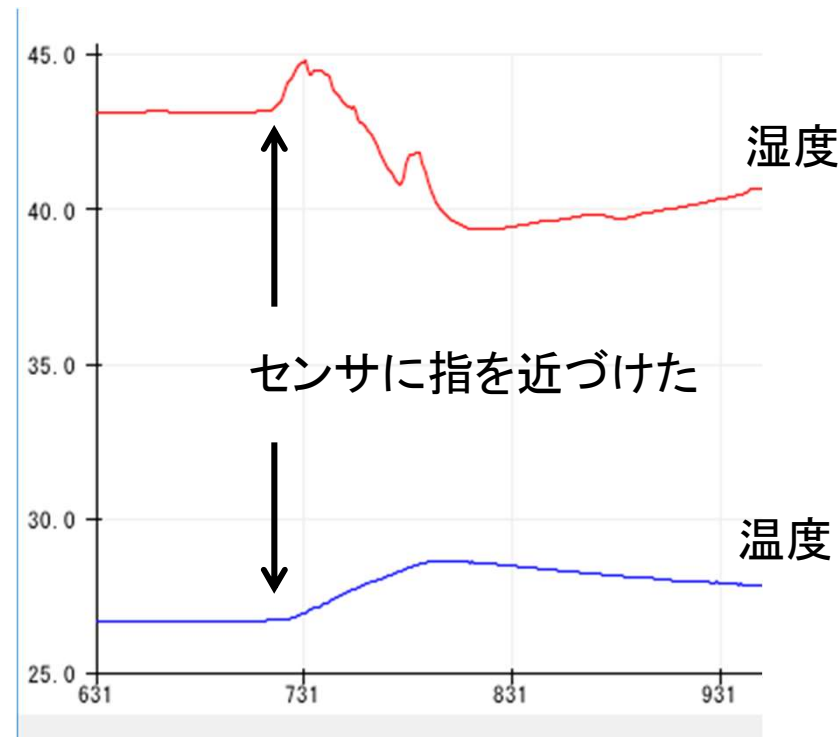
最後に100ms待つ

HTU21Dで測定

シリアルモニタの出力例



シリアルプロッタの出力例



記録の取り方: 表示の途中で、Wordで挿入、部分選択とする

湿度の値はばらつきが大きい。

精度は $\pm 0.3^{\circ}\text{C}$ 、容差は0.3度 \sim 0.5度($5^{\circ}\text{C}\sim 60^{\circ}\text{C}$)。

参考) 温度・湿度計

温度はほぼ同じであるが、湿度は値が異なることがある。



温湿度・気圧計 BA-9116



EMPEXの温度・湿度計

参考) 温度・湿度計2



精度は $\pm 0.3^{\circ}\text{C}$ 、容差は 0.3 度 ~ 0.5 度($5^{\circ}\text{C}\sim 60^{\circ}\text{C}$)。

100円ショップの温度・湿度計

湿度計の原理

A

【湿度計について】

湿度計にも主に3つの計り方があります。

・バイメタル式湿度計

原理は温度計と同じで、金属(真ちゅう)に湿気を吸いやすい、収縮率の異なる感湿剤を張り合わせ、湿度変化によって曲がるようにゼンマイ巻になっています。その時の感湿剤の変化によってゼンマイが巻いたり戻ったりした動きで針が動いています。

●湿度が上昇した時の動き



・デジタル湿度計

デジタル式ではくしをかみ合わせたような電極の上に感湿剤を間に挟み込まれており、感湿剤が湿気を多く含むと電流が流れやすく、乾燥すると流れにくくなるのでその時の流れやすさの抵抗の変化で計っています。

●デジタル湿度計



空気の温度と湿度

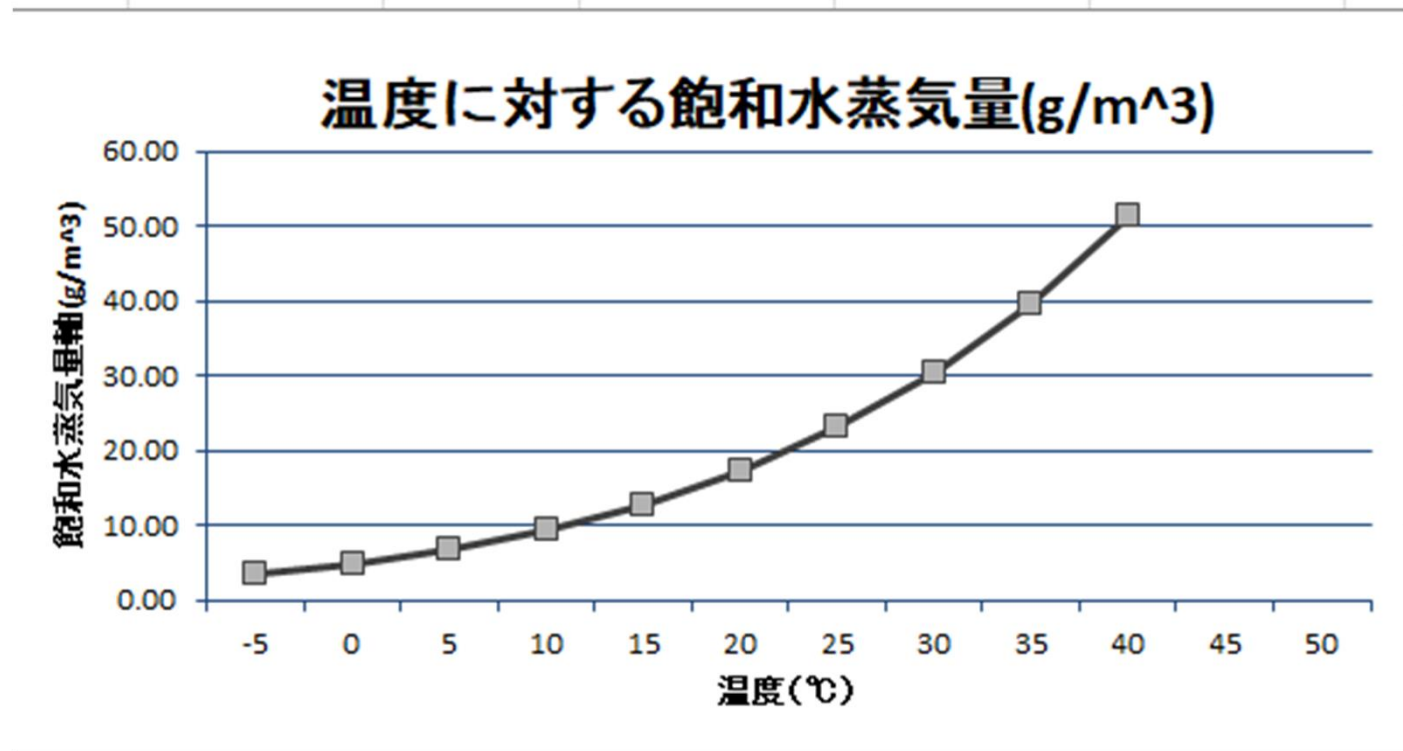
空気1m³がその時の温度(T)で含むことのできる水蒸気量(飽和水蒸気量、g/m³)に対して含まれている水蒸気量g/m³の割合を(相対)湿度という(%RH)。

飽和水蒸気量はTetensの式を用いて計算できる。Excelで計算する。

飽和水蒸気量 $a(T) = (217 \times e(T)) / (T + 273.15)$

飽和水蒸気圧 $e(T)$ は近似的に、 $e(T) = 6.1078 \times 10^{(7.5T / (T + 237.3))}$

温度に対する飽和水蒸気量	
Tetensの式による飽和水蒸気量	
室温 T	T _{ea} T (g/m ³)
-5	3.41
0	4.85
5	6.81
10	9.41
15	12.85
20	17.32
25	23.08
30	30.40
35	39.64
40	51.17
45	
50	



課題：湿度や水蒸気量・露点の計算問題

FUkuuExamから

<http://exam.fukuumedia.com/rikamon-7>

湿度や水蒸気量・露点の計算問題

次の表は、気温と飽和水蒸気量との関係を示している。現在の気温は 20°C で、空気 1m^3 中に 11.4g の水蒸気が含まれている。問いに答えよ。

1. この空気は、 1m^3 あたりあと何gの水蒸気を含むことができるか。(5.9g)
2. この時の湿度は何%か。(65.9%RH)
3. この空気を冷やしていったとき、水滴が生じ始めるのは、何 $^{\circ}\text{C}$ か。(13 $^{\circ}\text{C}$)
4. この温度を空気の何というか。(露点)

気温 [$^{\circ}\text{C}$]	飽和水蒸気量 [g]
10	9.4
11	10.0
12	10.7
13	11.4
14	12.1
15	12.8
16	13.6
17	14.5
18	15.4
19	16.3
20	17.3

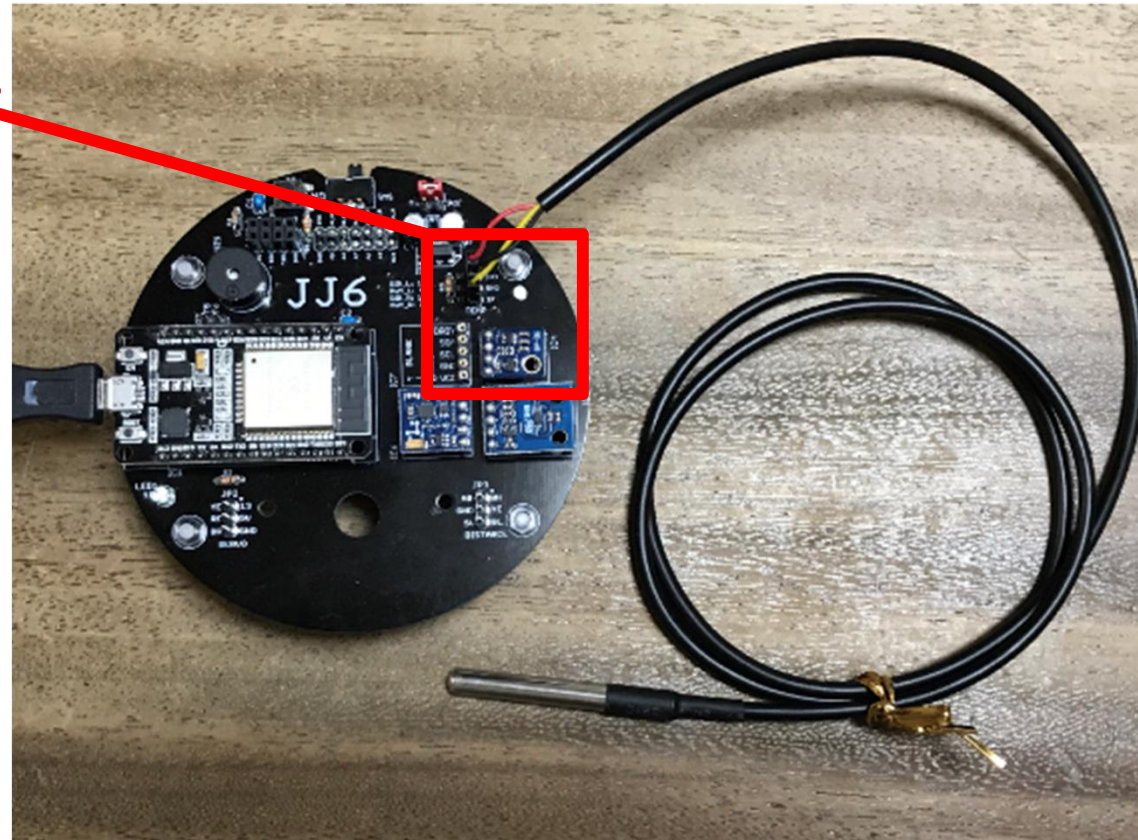
DS18B20で温度を測定する

DS18B20温度センサを取付ける。コネクタの位置を確認する(先に示した)。
温度センサは防水されている。温水に漬けて測定できる。

配布されたExample04からスケッチDs18b20onewireを読み込む。



DS18B20温度センサを取付けたところ



DS18B20 スケッチ

arduinoでコンパイル、転送する(実行する)

```
// ds18b20, one wire
#include <DallasTemperature.h> // DS18B20
```

```
const int DS_PIN = 27;
OneWire dsw(DS_PIN);
DallasTemperature ds(&dsw);
```

```
void setup(void) {
  Serial.begin(115200);
  delay(100);
  ds.begin();
}
```

```
void loop(void) {
  ds.requestTemperatures();
  Serial.println(ds.getTempCByIndex(0));
  // delay(500);
}
```

DS18B20

DallsTemperature.hをインクルード

DS_PINを27とする

OneWire接続

DallsTemperature. | このようにする

setup関数

Serial.begin 115200

| シリアル通信速度

ds.begin

繰り返しloop

dsで温度を要求

ds.getTempCByIndex(0)で値を読み

Serial.print

500ms 待つ

温度センサの比較

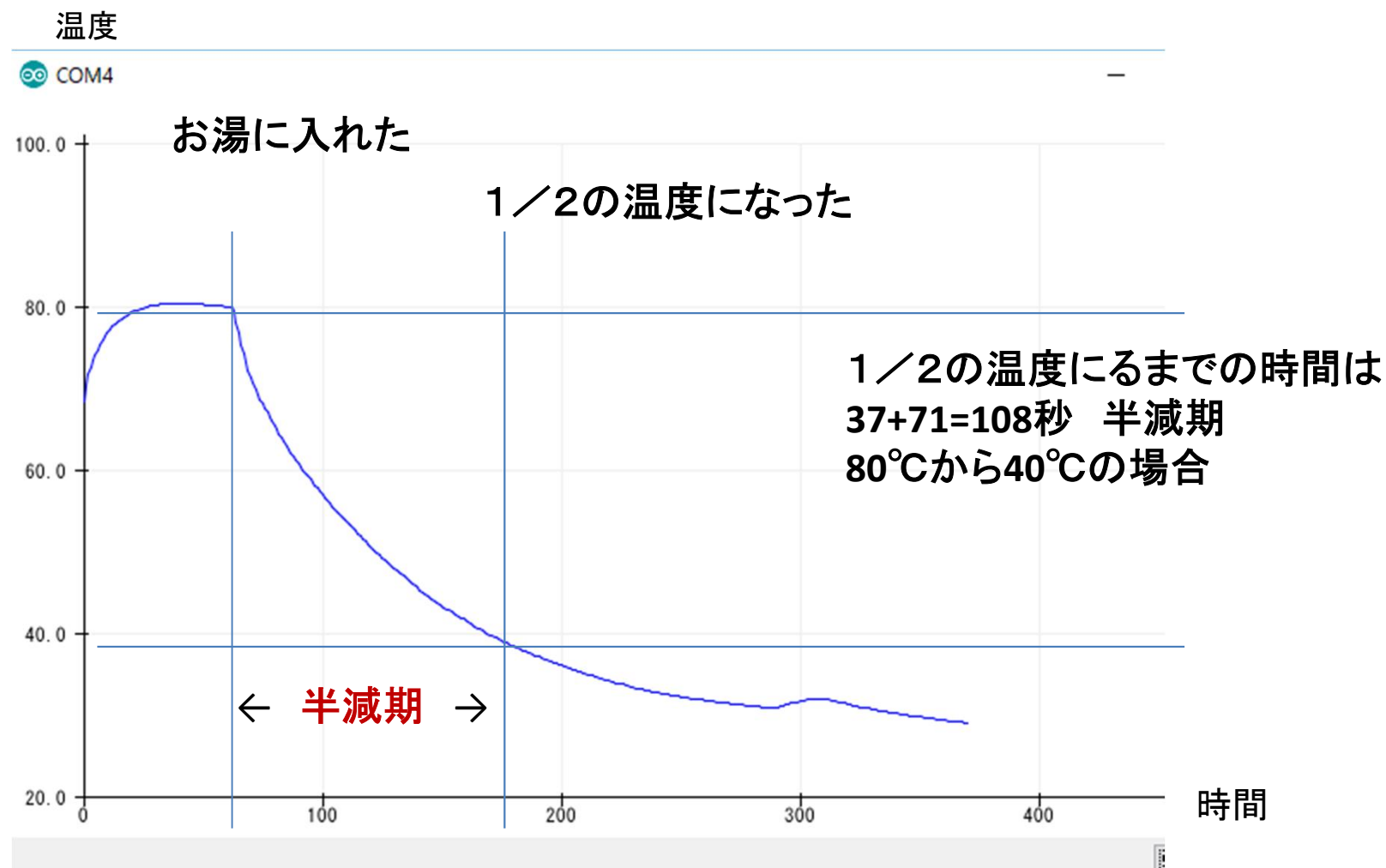
課題1) HTU21Dと比較してみよう。空気中の温度の比較

課題2) お湯をカップにいれて、温度計と温度センサ(DS18B20)(防水されている)をカップに入れる。シリアルプロッタで表示する。温度が一定になったら、温度計とセンサを取り出して、

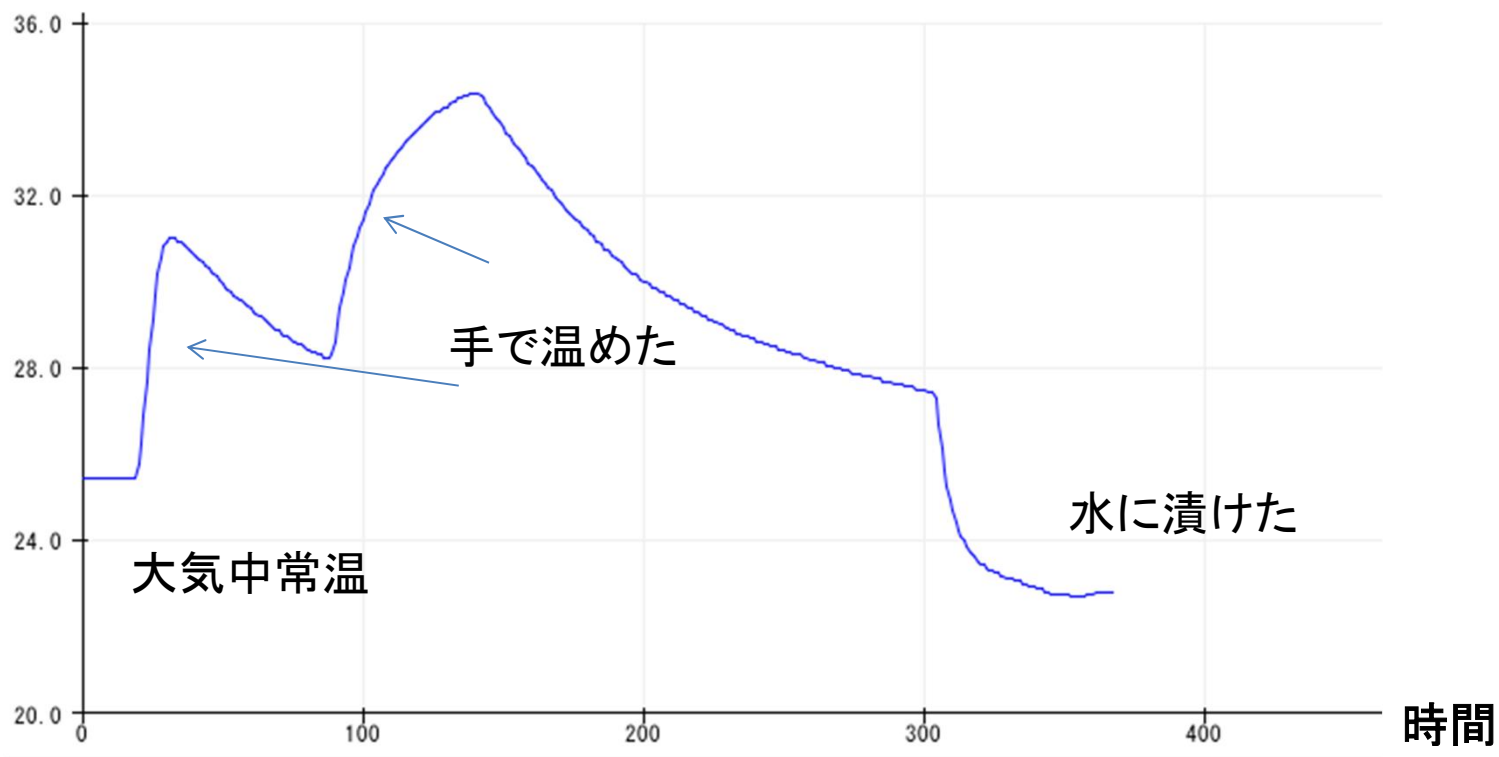
課題3) 温度の下がり具合をプロッタで測定する。どのようなグラフになるか？
また、温度が1/2になる時間はどの程度か？半減期と呼ばれる。測定してみよう。

参考)放射能は次第に減衰するので、半減期がある。「使用済み核燃料については、フィンランドの最終処分場では10~20万年安全に保管する必要がある」とのこと。もちろん、非常に短いものもあるし、非常に長いものもある。日本では最終処分地の選定を進めているが、ほとんど進んでいない。

容器に入れたお湯の温度の変化



60°C~30°Cの場合は、もっと長くなる。常温に近づくため。



温度の変化の例

<https://www.maximintegrated.com/jp/app-notes/index.mvp/id/148>

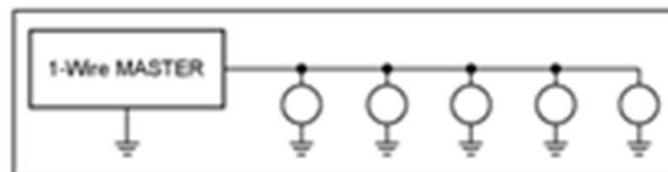
高信頼性長距離配線1-Wireネットワーク

1-Wireデバイスで使用可能なワイヤの種類およびトポロジには無数の組合せが存在します。このアプリケーションノートでは、1-Wireネットワークに関係する最も一般的で標準的なアプリケーションについてのみ説明

1-Wireネットワークのトポロジ

1-Wireネットワークは、多くの場合まったく「自由形式」の構造ですが、通常は1-Wireスレーブの分布および相互接続ワイヤの構成に基づいて、いくつかの一般化された分類に該当します。

1. リニア型トポロジ。1-Wireバスは、マスターを起点として最も遠いスレーブデバイスまで延びる単一のペアです。他のスレーブは、目立った(3mを超える)支線あるいは「スタブ」なしで1-Wireバスに接続されます。



上記は One-Wire 接続の簡単な場合の接続例である。

1本の線に複数のデバイスを接続する方法は幾つかある。電気的な制限もあるので、参考資料を探してください。