2016年6月25日(土)

気象モニターをつくろう
第6回 センサ情報のLEDによる可視化

2016年6月25日（土）10時～

徳島大学大学院 理工学研究部 総合技術センター
徳島大学 理工学部 情報光システムコース
担当： 辻　　明典

連絡先：
770-8506 徳島市南常三島町2-1
TEL/FAX： 088-656-7485
E-mail:：a-tsuji@is.tokushima-u.ac.jp

1 概要

本日の予定

1　概要

2　温度・湿度の計測

3　LEDとカラーマッピング

4　温度・湿度と指数

5　大気圧と高度

講座の概要

気象モニターをつくろう－誰でもできるプロトタイピング－

講師：川上 博（徳島大学名誉教授）
　　辻 明典（徳島大学大学院理工学研究部総合技術センター）

曜日・時間：土曜日 10時00分～11時30分

スケジュール：

① 5／21 概要，開発環境セットアップ
② 5／28 はじめてのスケッチ
③ 6／4 照度センサとシリアルモニタ
④ 6／11 フルカラーLEDを使う
⑤ 6／18 温湿度・大気圧センサとLCD
⑥ 6／25 センサ情報のLEDによる可視化

2 温度・湿度の計測

Example601A: 温度・湿度の計測（関数）

・温度・湿度を計測する関数を作成

```
#include "SparkFunBME280.h"

BME280 bme280;
float t, h;

void loop() {
  bme280Read();

  Serial.print(t, 1);
  Serial.write(0xDF);
  Serial.println("C ");
  Serial.print(h, 1);
  Serial.println("%");

  delay(1000); // 1秒毎に更新
}
```

■ 温度，湿度を測る関数

```
void bme280Read() {
  t = bme280.readTemperature();
  h =bme280.readHumidity();
}
```

Example601B: 温度・湿度の計測（シリアル通信）

・温度・湿度を通信する関数を作成

```
#include "SparkFunBME280.h"
#include "LiquidCrystal_I2C.h"

BME280 bme280;
float t, h;
LiquidCrystal_I2C lcd(0x27,16,2);

void loop() {
  bme280Read();
  bme280SerialPrint();
  lcd.setCursor(0,0);
  lcd.print(t, 1); lcd.write(0xDF);
  lcd.print("C ");
  lcd.setCursor(0,1);
  lcd.print(h, 1);
  lcd.print("%");
  delay(1000); // 1秒毎に更新
}
```

■ 温度，湿度を通信する関数

```
void bme280SerialPrint () {
  Serial.print(t, 1);
  Serial.write(0xDF);
  Serial.println("C ");
  Serial.print(h, 1);
  Serial.println("%");
}
```

Example601C: 温度・湿度の計測（シリアル通信，LCD）

・温度・湿度をLCDに表示する関数を作成

```
#include "SparkFunBME280.h"
#include "LiquidCrystal_I2C.h"

BME280 bme280;
float t, h;
LiquidCrystal_I2C lcd(0x27,16,2);

void loop() {
  bme280Read();
  bme280SerialPrint();
  bme280LcdPrint();
  delay(1000); // 1秒毎に更新
}
```

■ 温度，湿度をLCD表示する関数

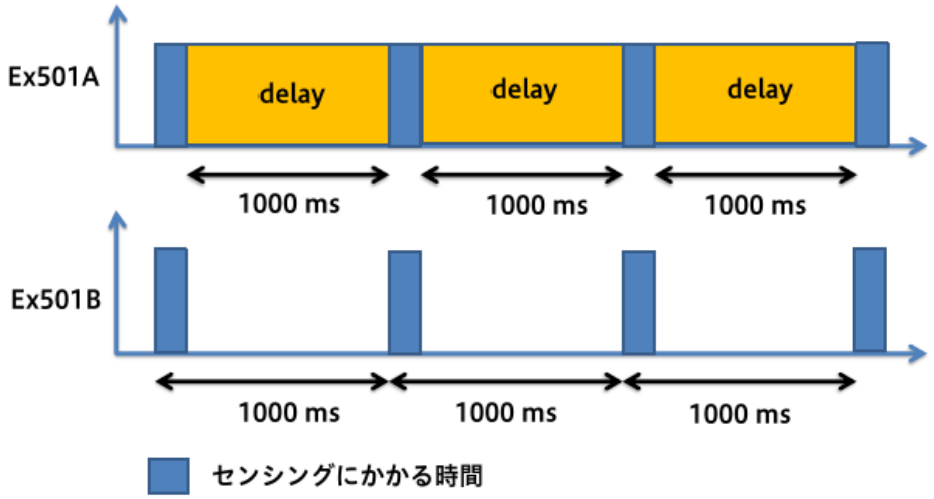
```
void bme280SerialPrint () {
  lcd.setCursor(0,0);
  lcd.print(t, 1); lcd.write(0xDF);
  lcd.print("C ");
  lcd.setCursor(0,1);
  lcd.print(h, 1);
  lcd.print("%");
}
```

Example601D: 温度・湿度の計測（1秒毎に実行）

・delayを使わず1秒毎に実行

```
#include "SparkFunBME280.h"
#include "LiquidCrystal_I2C.h"

BME280 bme280;
float t, h;
LiquidCrystal_I2C lcd(0x27,16,2);

long c_time=0, p_time=0;

void loop() {

  if (      -       > 1000  ) {

    bme280Read();
    bme280SerialPrint();
    bme280LcdPrint();
  }
}
```

Example501B参照



3 LEDとカラーマッピング

色と意味

信号機

青・・・・・進んでも良い
黄・・・・・止まれ
赤・・・・・止まれ
黄点滅・・・注意して進む
赤点滅・・・一時停止して進む

赤：波長が長く，遠くから視認しやすい

RGB表色系

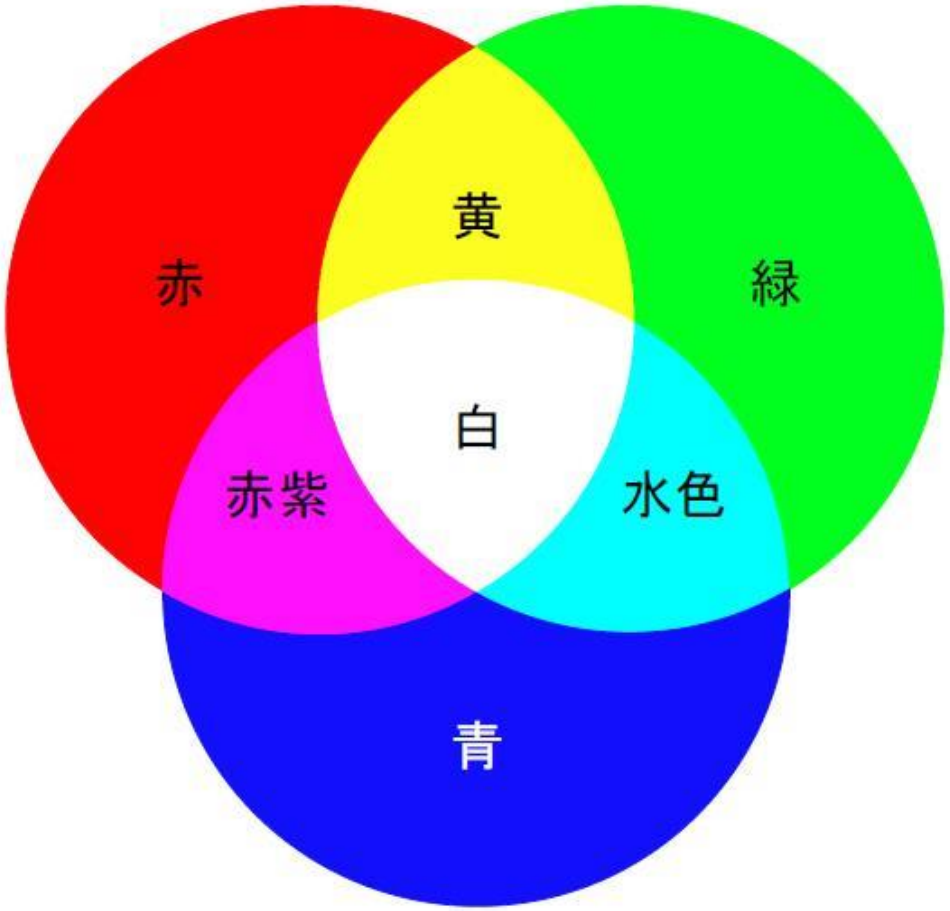
光の三原色

- 赤(Red)
- 緑(Green)
- 青(Blue)

の重ね合わせ（加法混色）

CRGB(R, G, B);

R値：0 – 255 赤の明るさ
G値：0 – 255 緑の明るさ
B値：0 – 255 青の明るさ



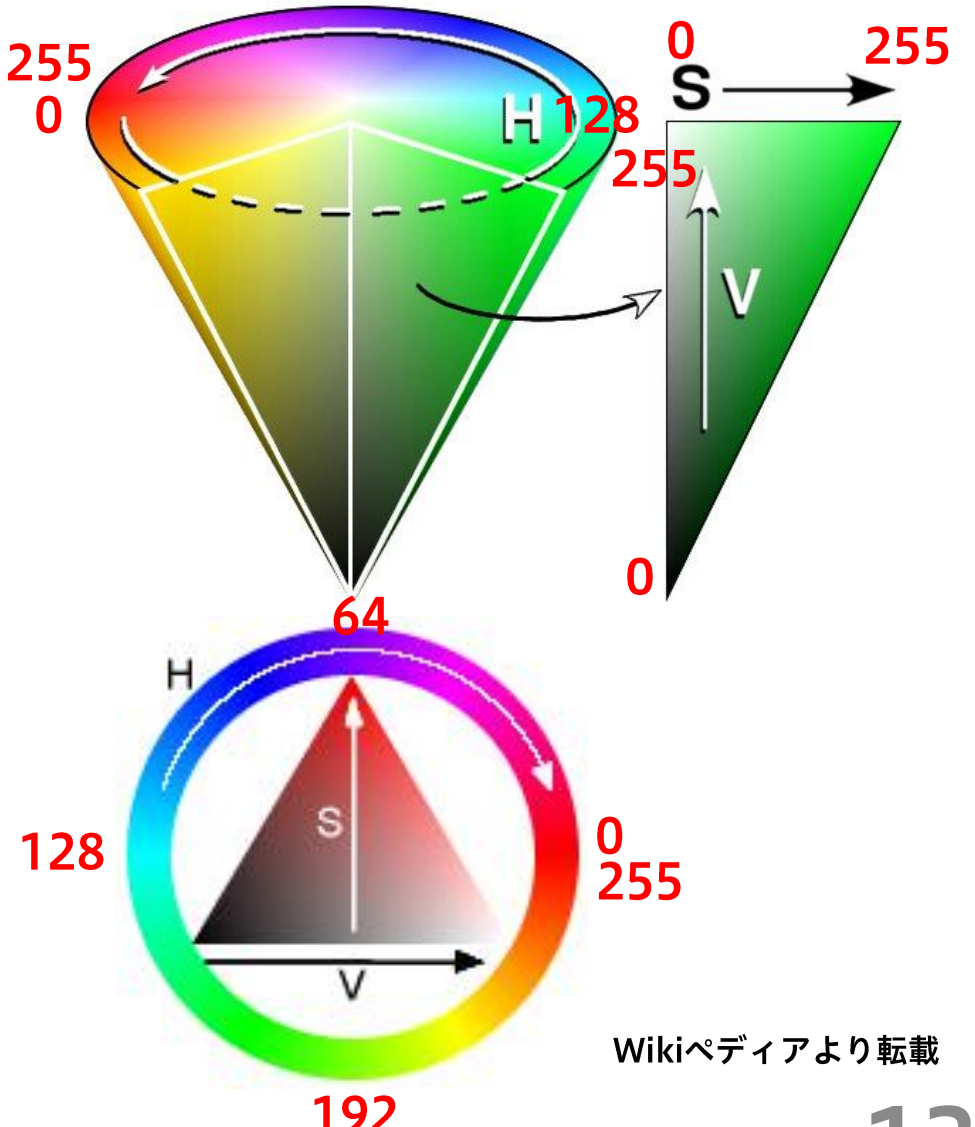
HSV表色系

HSV表色系

- 色相（H: Hue）
  色の種類
- 彩度（S: Saturation）
  色の鮮やかさ
- 明度（V: Value, Brightness）
  色の明るさ

CHSV（h, s, v）;

h値：0 – 255
s値：0 – 255
v値：0 - 255



Wikiペディアより転載

LEDの色相を変化

・温度のカラーマッピング例

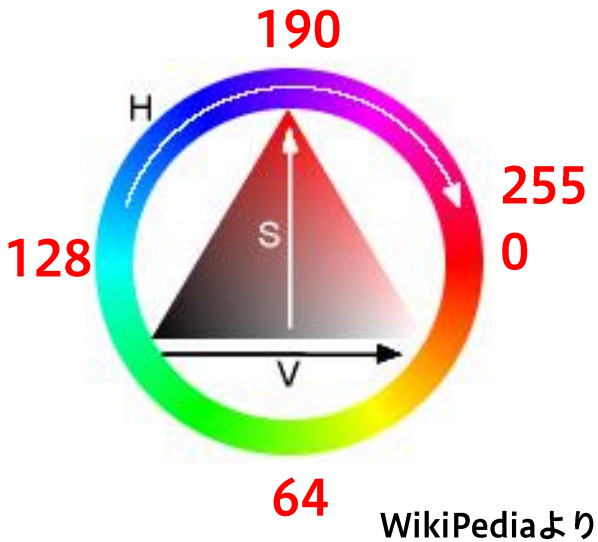
温度　40℃　　20℃　　0℃
色相　0　　64　　120

map(0, 40, 120, 0)

＜map関数＞

map(value, fromLow, fromHigh,
toLow, toHigh)

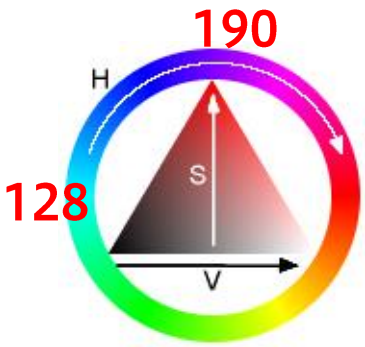
value　　：変換したい数値
fromLow　：現在の範囲の下限
fromHigh：現在の範囲の上限
toLow　　：変換後の範囲の下限
toHigh　　：変換後の範囲の上限



WikiPediaより転載

Example602A: 温度変化にあわせLEDの色変更

・温度に合わせLEDの色変更

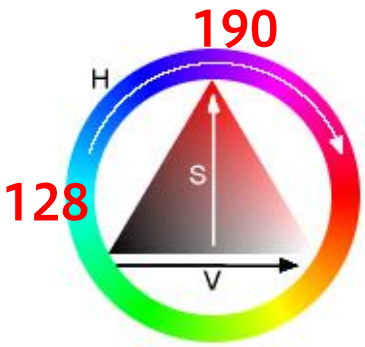


WikiPediaより転載

```
float t, h;
void loop() {
    ・・・
   bme280Read();
   bme280SerialPrint();
   bme280LcdPrint();
//    t = 0; // for test
   int hue = map((int)t, 0, 40, 120, 0); // 温度 0から42度を色相にマッピング
   leds[0] = CHSV(hue, 255, 255);
   FastLED.show();
    ・・・
}
```

Example602B: 湿度変化にあわせLEDの色変更

・湿度に合わせLEDの色変更



WikiPediaより転載

```
float t, h;
void loop() {
    ・・・
   bme280Read();
   bme280SerialPrint();
   bme280LcdPrint();
//    h = 0; // for test
   int hue = map((int)h,    ,    ,    ,    ); // 湿度    から    %を色相にマッピング
   leds[0] = CHSV(hue, 255, 255);
   FastLED.show();
    ・・・
}
```

4 温度・湿度と指数

体感温度

- **体感温度**
  - 人間の肌で感じる温度の感覚を定量的に表した温度
- **ヒートインデックス**
  - 気温と湿度から計算した体感温度
  - 高温のストレスを表す指標（℃）
  - NOAA's National Weather Service（アメリカ国立気象局）で採用

状態	温度	症状
注意	27-32℃	通常過激な運動をしない
極度な注意	32-41℃	脱水症状の危険
危険	41-54℃	差し迫った熱中症
非常に危険	54℃以上	生命の危険

Metrication of Template:HeatTable

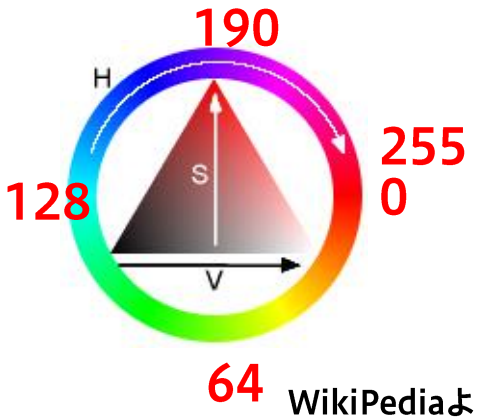
Relative Humidity (%) \ temperature (°C)	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
40	27	28	29	30	31	32	34	35	37	39	41	43	46	48	51	54	57
45	27	28	29	30	32	33	35	37	39	41	43	46	49	51	54	57	
50	27	28	30	31	33	34	36	38	41	43	46	49	52	55	58		
55	28	29	30	32	34	36	38	40	43	46	48	52	55	59			
60	28	29	31	33	35	37	40	42	45	48	51	55	59				
65	28	30	32	34	36	39	41	44	48	51	55	59					
70	29	31	33	35	38	40	43	47	50	54	58						
75	29	31	34	36	39	42	46	49	53	58							
80	30	32	35	38	41	44	48	52	57								
85	30	33	36	39	43	47	51	55									
90	31	34	37	41	45	49	54										
95	31	35	38	42	47	51	57										
100	32	36	40	44	49	54											

Caution
Extreme Caution
Danger
Extreme Danger

Wikipediaより転載

Example603A: 体感温度のカラーマッピング

```
void loop() {
   ・・・
   float hic = bme280HI();
   Serial.println(hic);
    int hue = map((int)hic,    ,    ,    ,    ); // 湿度    から    %を色相にマッピング
   leds[0] = CHSV(hue, 255, 255);
   FastLED.show();
    ・・・
}
```



WikiPediaより転載

■ ヒートインデクスを求める関数

```
float bme280HI() {
  float hi = -8.784695 +
    1.61139411 * t + 2.338549   * h +
    -0.14611605 * t * h +
    -0.01230809 * pow(t, 2) +
    -0.01642482 * pow(h, 2) +
     0.00221173 * pow(t, 2) * h +
     0.00072546 * t *pow(h, 2) +
    -0.00000358 * pow(t, 2) * pow(h, 2);
   return hi;
}
```

不快指数

■ 不快指数(DI：Discomfort Index)
– 人間が生活するうえで不快と感じる体感を気温と湿度で表した指数

$$DI = 0.81*T+0.01*H*(0.99*T-14.3)+46.3$$

T： 気温（℃），H：相対湿度（%RH）

不快指数（DI）	体感
～55	寒い
55～60	肌寒い
60～65	何も感じない
65～70	快い
70～75	暑くない
75～80	やや暑い
80～85	暑くて汗が出る
85～	暑くてたまらない

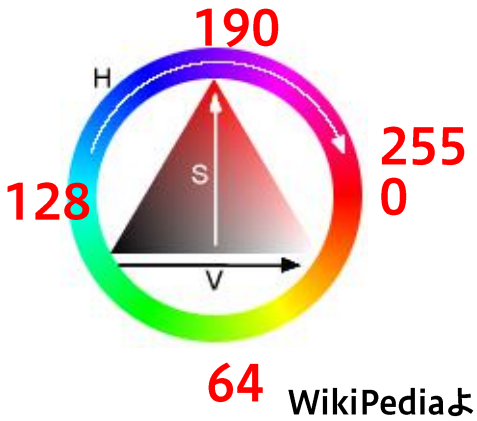
・地球の最高気温（WMO）
1913年7月10日
アメリカ合衆国カリフォルニア州
デスバレー
T=56.7℃

・日本の最高気温（気象庁）
2013年8月12日
高知県四万十市江川崎
T=41.0℃

Example603B: 不快指数のカラーマッピング

■ 不快指数のカラーマッピング

```
void loop() {
   ・・・
   float di = bme280DI();
   Serial.println(di);
    int hue = map((int)di,    ,    ,    ,    ); // 湿度    から    %を色相にマッピング
   leds[0] = CHSV(hue, 255, 255);
   FastLED.show();
    ・・・
}
```



WikiPediaより転載

■ 不快指数を求める関数

```
float bme280DI() {
  float di = ・・・
```

DI = 0.81*T+0.01*H*(0.99*T-14.3)+46.3
T： 気温（℃），H：相対湿度（%RH）

```
  return di;
}
```

5 大気圧と高度

大気圧と高度の関係

国際民間航空機関＊ICAO: International Civil Aviation Organizationによる
国際標準大気モデル（1952年）

$$Altitude(m) = 44330.77 \times \{1 - (P / 101325)^{0.190263}\}$$

Altitude：高度（Altitude）［m］
P：大気圧（Atmospheric Pressure）［Pa］

＜条件＞高度 0［m］
地上の気圧P_0： 1013.25［hPa］
地上の気温T_0： 15［℃］（288.15［K］）

＊ 国連の専門機関の一つ（本部：カナダ，モントリオール）

Example604A: 大気圧の計測

- 大気圧の計測

```
float t, h, p; // 温度，湿度，大気圧

void bme280Read() {
  t = bme280.readTempC();
  h = bme280.readFloatHumidity();
  p = bme280.readFloatPressure(); // 気圧計測(Pa)
}
```

- 単位の変換（Pa:パスカル → hPa:ヘクトパスカル）

```
Serial.print(p/100, 1);
Serial.println("hPa");

lcd.print(p/100, 1);
lcd.print("hPa");
```

Example604B: 大気圧から高度を求める関数

- 大気圧から高度を求める関数

```
float alt;
float bme280Altitude() {
   alt = ・・・
   Altitude(m) = 44330.77 x (1 - (P / 101325) 0.190263)
   return alt;
}
```

- X^y （xのy乗）

```
pow(X, Y);

pow( (p / 101325.0), 0.190263);
```