

2015年10月17日(土)

# LEDを光らせよう（応用編）

まわりを感じて光る小作品を作る  
液晶ディスプレイ，温度・湿度センサ

2015年10月17日（土）10時～

---

徳島大学大学院ソシオテクノサイエンス研究部

徳島大学工学部知能情報工学科

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](mailto:a-tsuji@is.tokushima-u.ac.jp)

# 本日の予定

- 1 講座の概要
  - 2 液晶ディスプレイ
  - 3 液晶ディスプレイとLED
  - 4 温度・湿度センサ
  - 5 温度・湿度センサとLED
- 付録：ライブラリの関数一覧

# 講座の概要

**講座名：**LEDを光らせよう（応用編）まわりを感じて光る小作品を作る

**講師：**川上博（徳島大学名誉教授）  
辻明典（徳島大学ソシオテクノサイエンス研究部  
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 10 / 3 講座概要，配布部品確認，基板説明，ライブラリの追加，動作確認
- ② 10 / 10 フルカラーLED（FastLEDの使い方），光センサ
- ③ 10 / 17 **液晶ディスプレイ，温度・湿度センサ**
- ④ 10 / 24 センサ（距離）
- ⑤ 11 / 7 リモコン（LEDとセンサの操作）
- ⑥ 11 / 14 作品作り，追加情報，ディスカッション

# まわりを感じて光る

## 何を感じる？

照度：明るい, 暗い  
温度：暑い, 寒い  
湿度：湿気ている, 乾燥している  
距離：どのくらい離れている

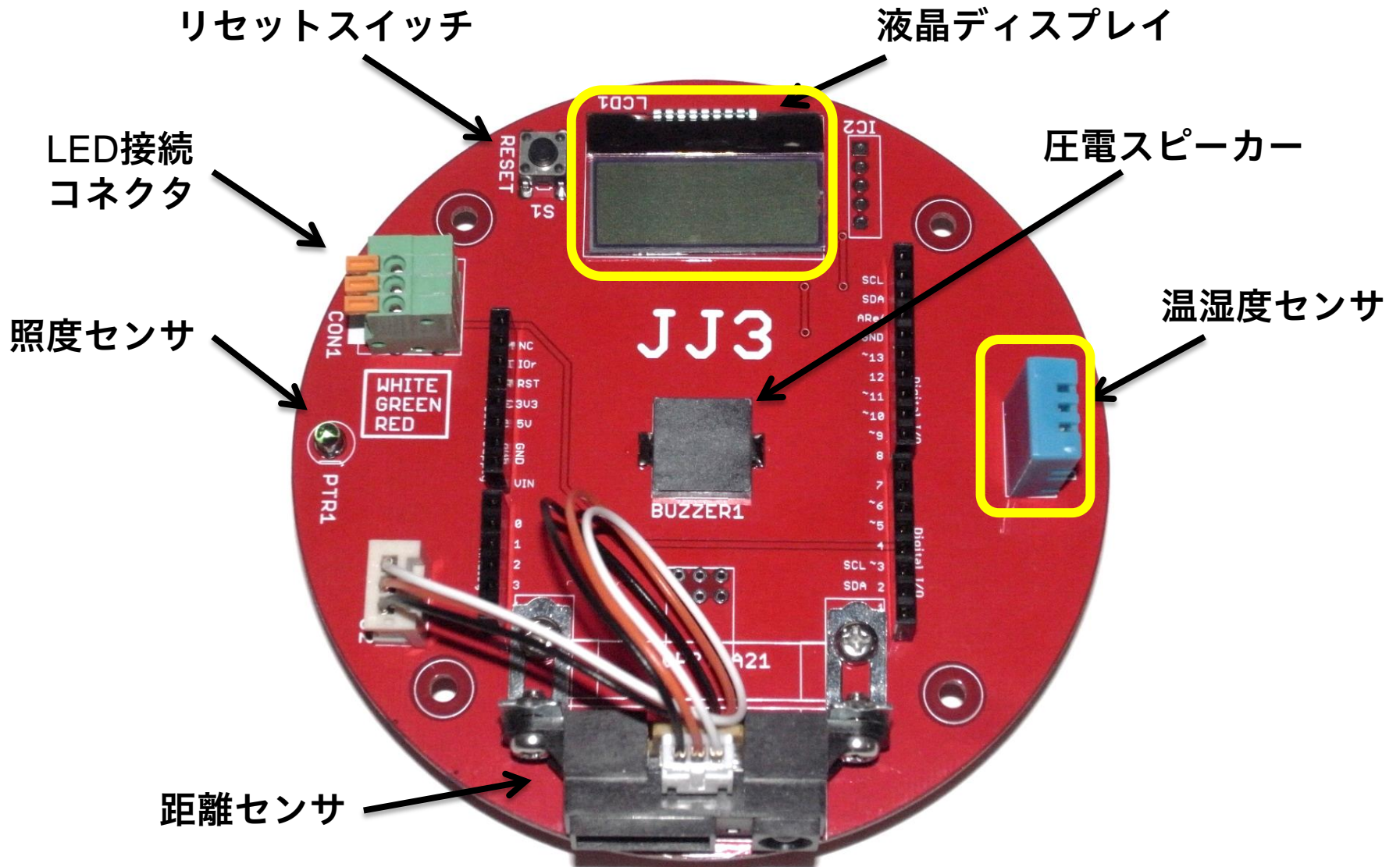
マイコンにセンサを接続して  
感覚機能を追加

LED：明暗, 色合い  
スピーカ：音量, 音色  
液晶ディスプレイ：表示

センサで感じた刺激を表現

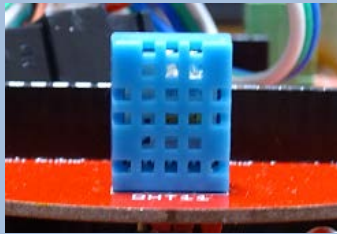
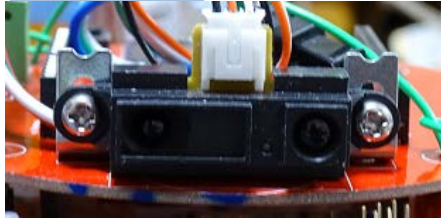


# JJ3シールド基板



# マイコンによる入出力処理

入力

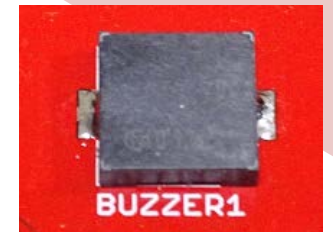
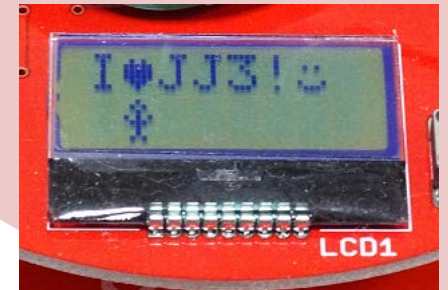
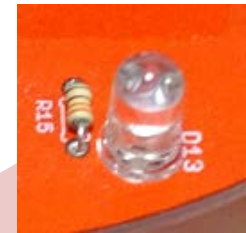


処理



マイコン

出力





# 液晶ディスプレイ， 温度・湿度センサの仕様

## ■ 液晶ディスプレイ (ACM0802)

モノクロ液晶 8文字 X 2行

シリアル接続

Arduino : SDA, SCLピン

=> **Arduino-ST7032ライブラリ**



液晶ディスプレイ

## ■ 温度・湿度センサ (DHT11)

測定範囲 : 温度 0~50°C 湿度 20~90%

測定誤差 : 温度  $\pm 2^{\circ}\text{C}$  湿度  $\pm 5\%$

精度 : 湿度 8 bit (分解能 : 1%)

温度 8 bit (分解能 : 1°C)

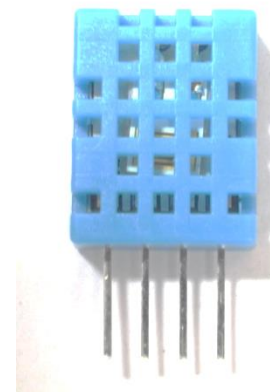
計測時間間隔 : 2秒

動作電圧 : 3.3V~5.5V

出力 : 単線双方向シリアル

Arduino : D9ピン

=> **Arduino-DHT-sensorライブラリ**



温度・湿度センサ

# 液晶ディスプレイ

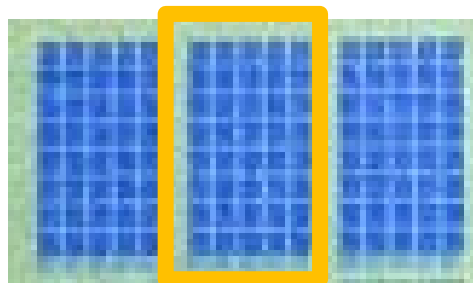
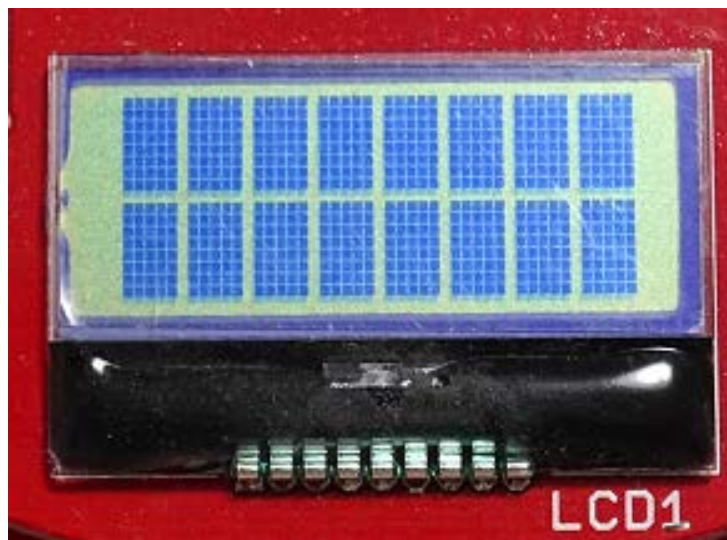
---



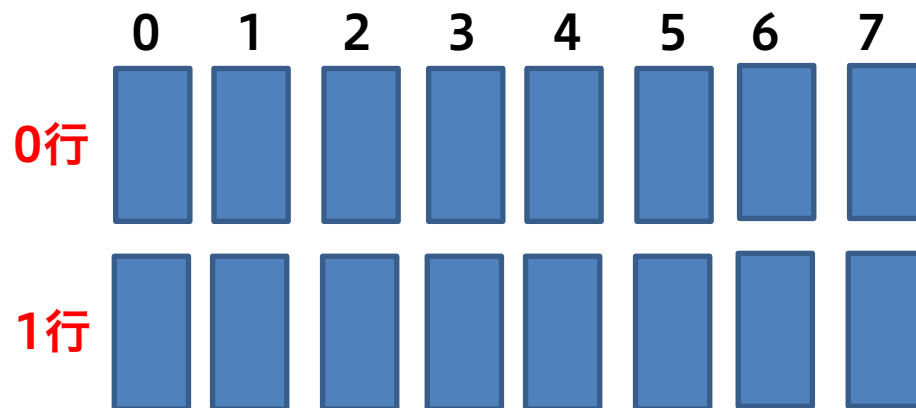
# 液晶ディスプレイの表示方法

## ■ 液晶ディスプレイ (LCD: Liquid Crystal Display)

8文字 X 2行



8ドット X 5ドット / 文字



(列, 行)

(0,0) (1,0) (2,0) . . . (6,0) (7,0)

(0,1) (1,1) (2,1) . . . (6,1) (7,1)

# Example 901 : 文字列の表示

```
#include <Wire.h>
#include <ST7032.h>
```

```
ST7032 lcd;
```

```
void setup() {
  lcd.begin(8, 2); // 8文字, 2行
  lcd.setContrast(30); // 液晶のコントラスト
```

```
  lcd.setCursor(0, 0); // 0列, 0行にセット
  lcd.print("JJ3!"); // 文字列を表示
```

```
}
```

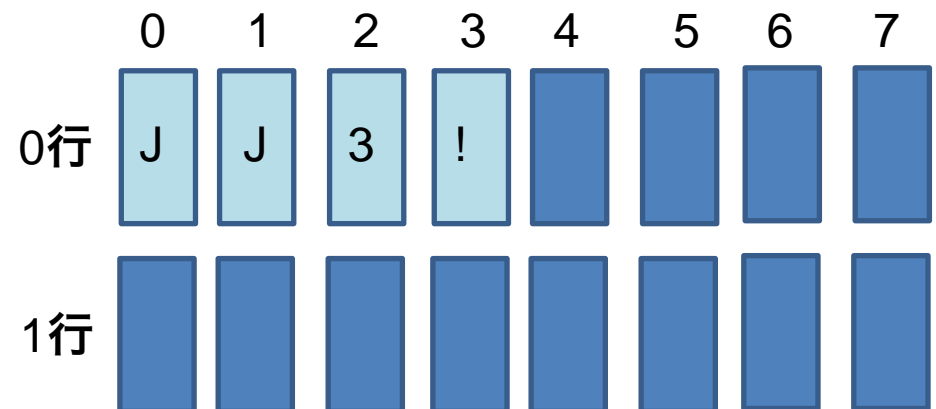
```
void loop() {}
```

液晶使用時に必要

液晶の指定の位置に文字列表示

指定の位置 : (0, 0)

文字列(4文字) : "JJ3!"

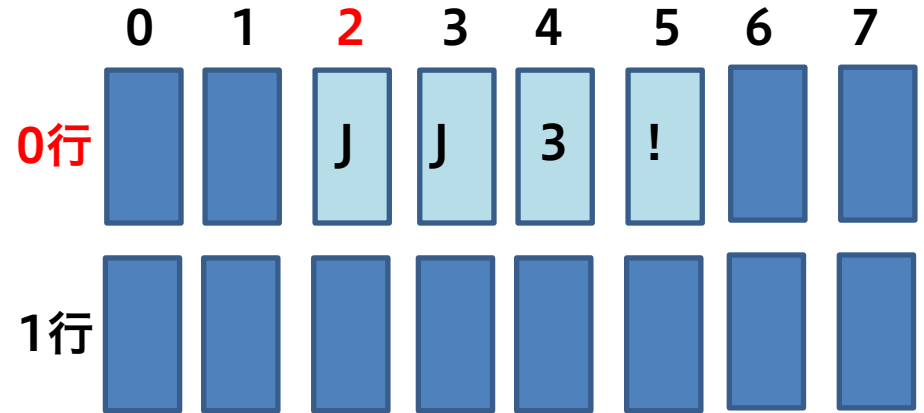
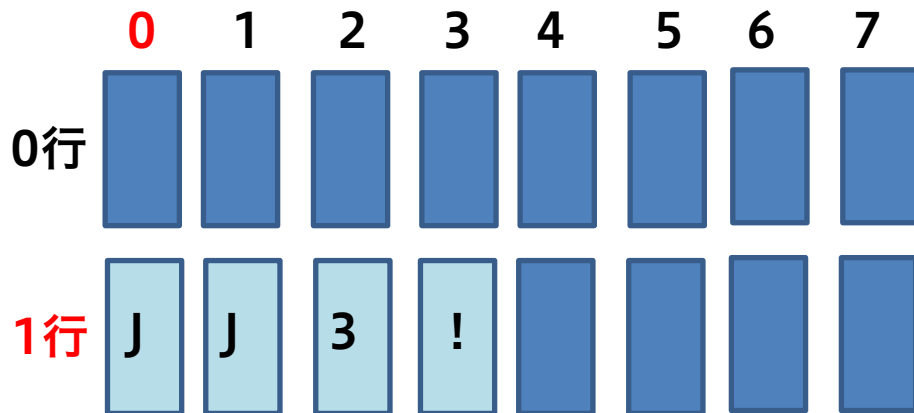


# Example 901 : 任意の位置に文字列の表示

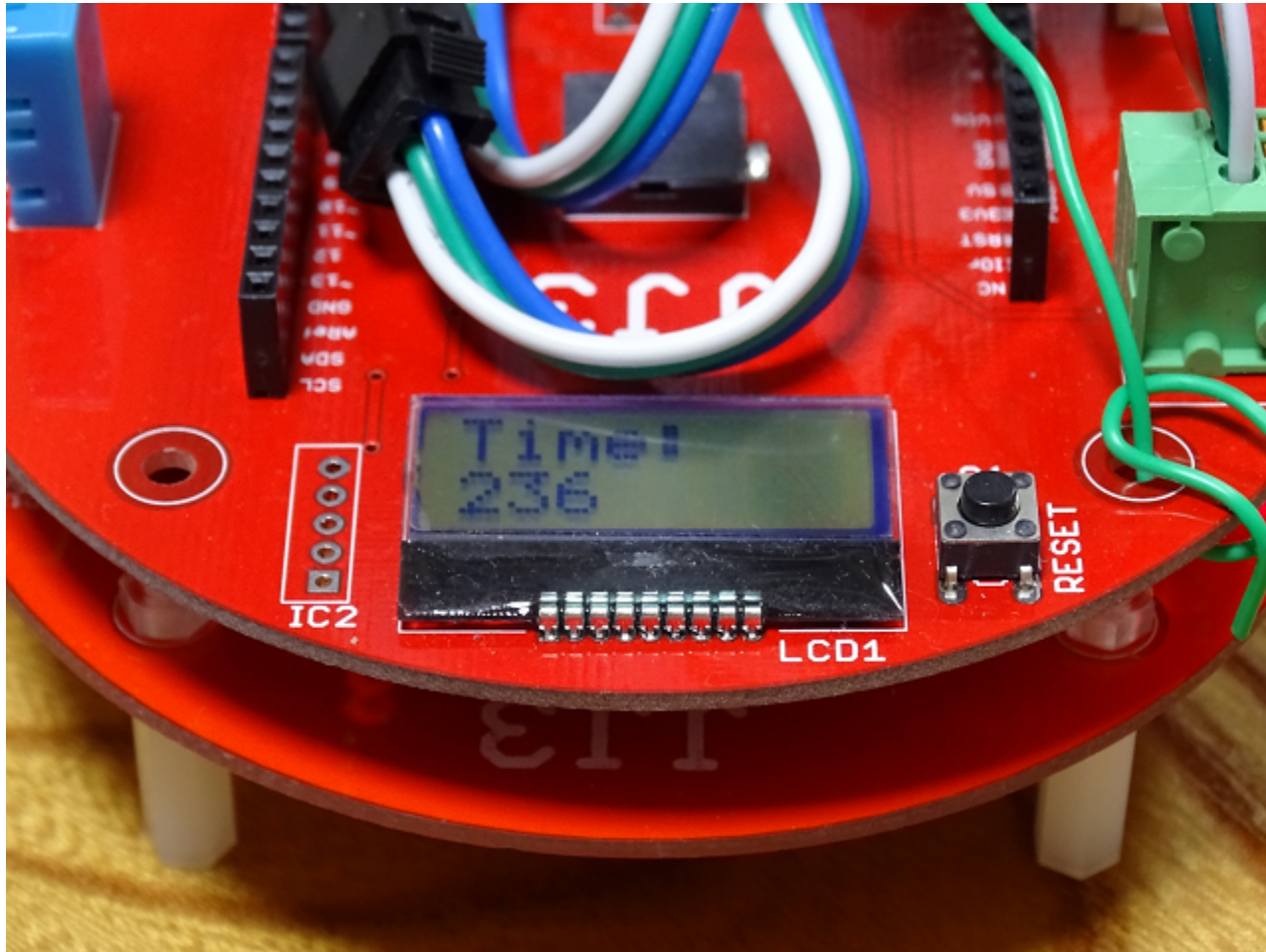
■ `lcd.setCursor(0, 1);`



■ `lcd.setCursor(2, 0);`



## Example 902 : 経過時間の表示



# Example 902 : 経過時間の表示

```
#include <Wire.h>
#include <ST7032.h>
```

```
ST7032 lcd;
```

```
void setup() {
```

```
  lcd.begin(8, 2); // 8文字, 2行
```

```
  lcd.setContrast(30); // 液晶のコントラスト
```

```
  lcd.setCursor(0, 0); // 0列, 0行にセット
```

```
  lcd.print("Time"); // 文字列を表示
```

```
}
```

```
void loop() {
```

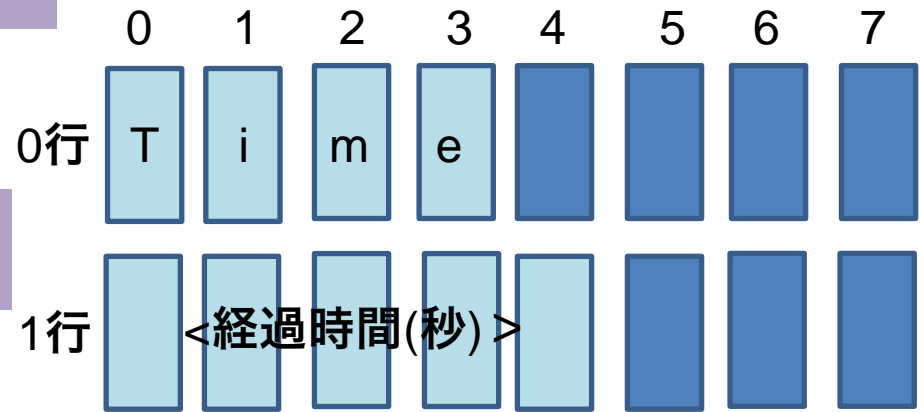
```
  lcd.setCursor(0, 1); // 0列, 1行にセット
```

```
  lcd.print(millis() / 1000); // 秒
```

```
}
```

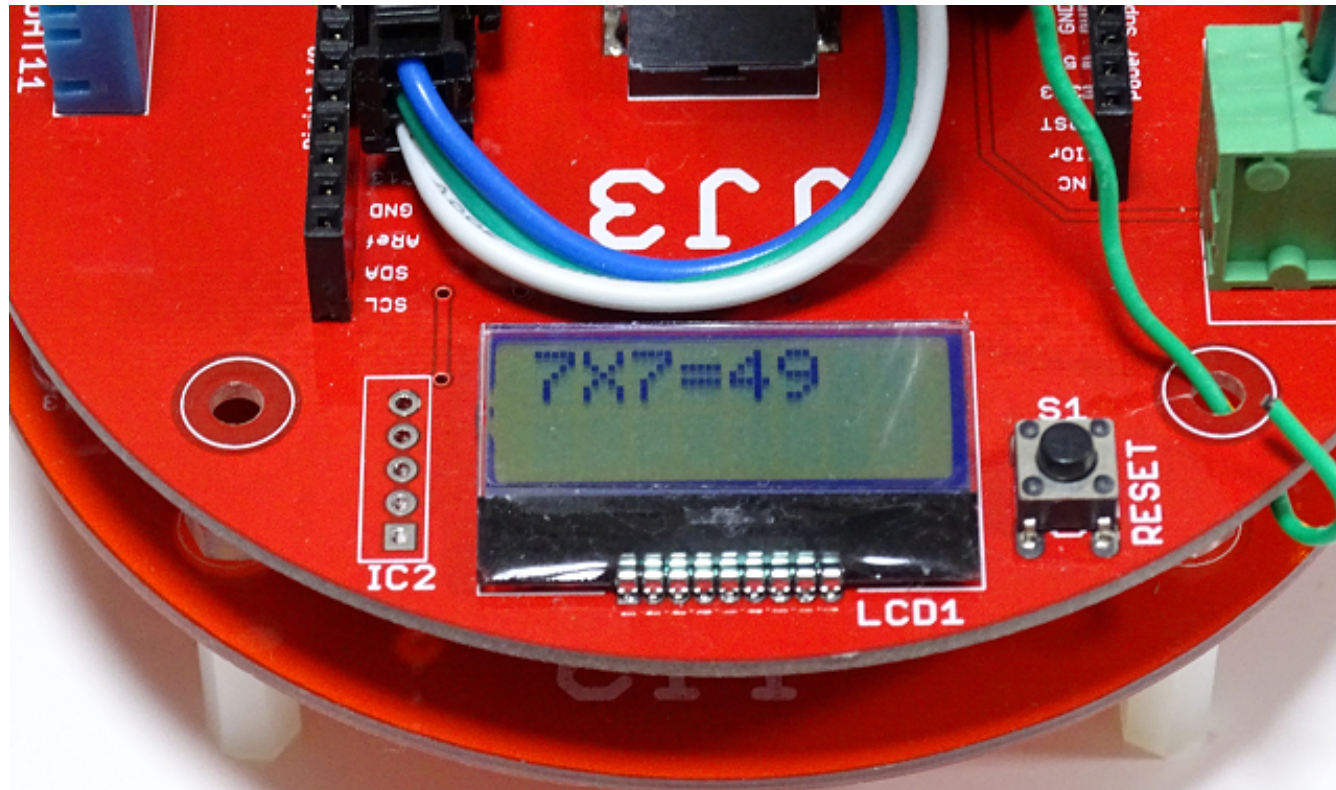
- millis(): 電源投入後の経過時間をミリ秒単位で計測

1 ミリ秒 = 1 / 1000 秒





# Example 903 : 九九の計算結果の表示



# Example 903 : 九九の計算結果の表示

## ■ 九九の計算結果を1秒毎に表示

```
void loop() {  
  for (int x=1; x<=9; x++) {  
    for (int y=1; y<=9; y++) {  
      int z = x * y; // 九九の計算  
      lcd.clear(); // 液晶の表示をクリア  
      lcd.setCursor(0, 0); // 0列, 0行にセット  
      // 計算結果を表示  
      lcd.print(x); lcd.print("X"); lcd.print(y);  
      lcd.print("="); lcd.print(z);  
      delay(1000);  
    }  
  }  
}
```

## ■ 九九の計算

$z = x * y$ ;  
x掛けるyの計算結果をzに入れる

## ■ for ループ

指定回数の繰り返し

x = 1 から x=9 まで 1ずつ増やす

y = 1 から y=9 まで 1ずつ増やす

## ■ 2重ループ

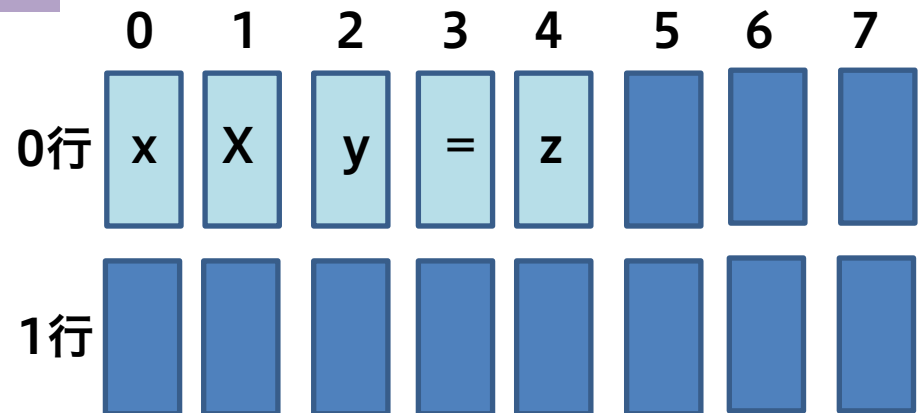
x=1, y=1,2,3,4,5,6,7,8,9

x=2, y=1,2,3,4,5,6,7,8,9

x=3, y=1,2,3,4,5,6,7,8,9

...

x=9, y=1,2,3,4,5,6,7,8,9

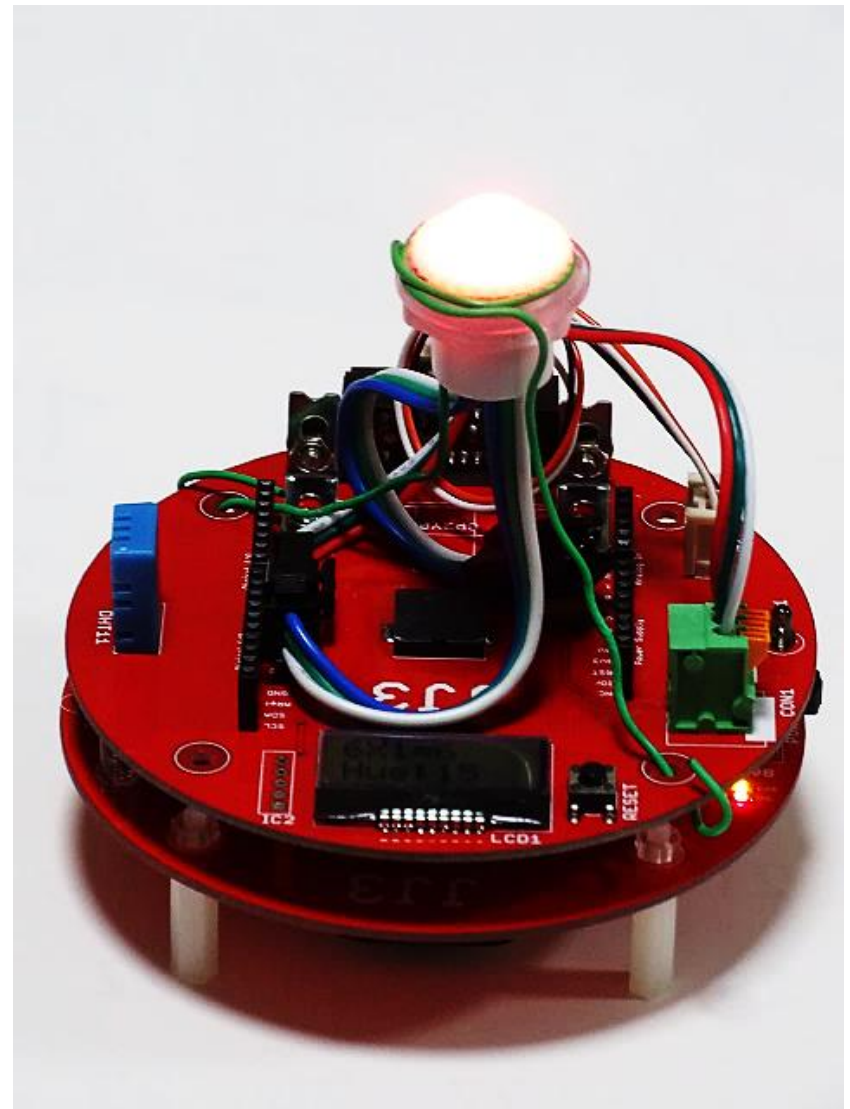
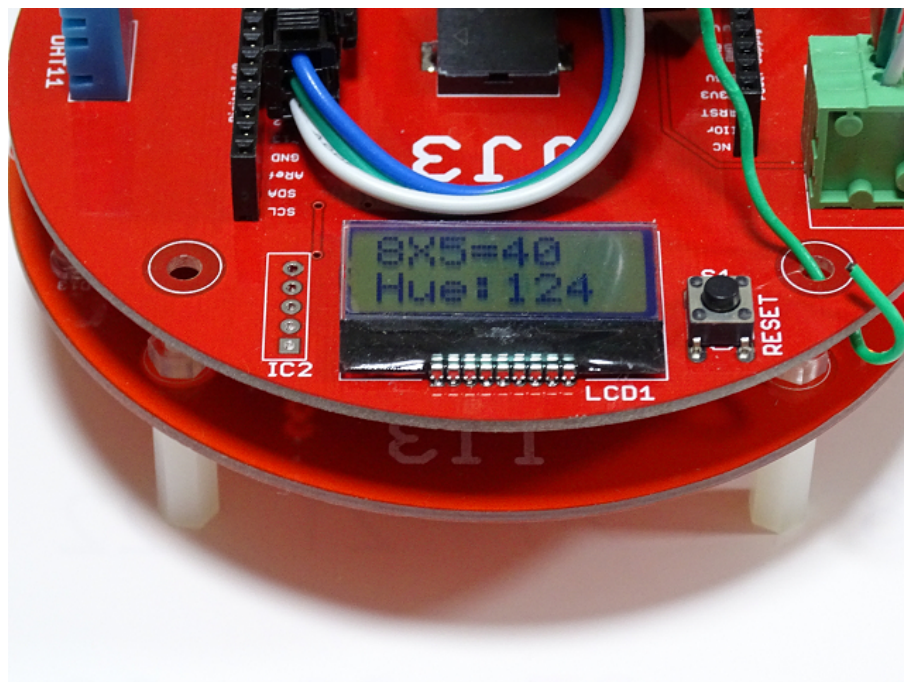




# 液晶ディスプレイとLED

---

# Example 904 : 九九の計算結果を色相に



# Example 904 : 九九の計算結果を色相に

## ■ 1秒毎に九九の計算結果を表示

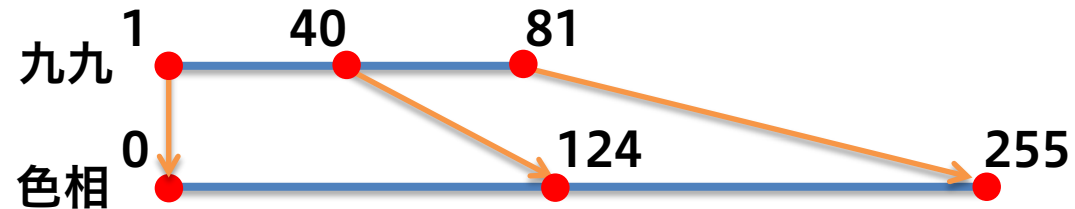
```
void loop() {  
  // 九九の計算  
  for (int x=1; x<=9; x++) {  
    for (int y=1; y<=9; y++) {  
      int z = x * y;  
      . . .  
      lcd.setCursor(0, 1);  
      lcd.print("Hue:");  
      int hue = map(z, 1, 81, 0, 255);  
      lcd.print(hue);  
  
      leds[0] = CHSV(hue, 255, 255);  
      FastLED.show();  
      delay(1000);  
    }  
  }  
}
```

## ■ map関数

九九の計算結果(z) : 1~81

色相(hue) : 0~255

に対応



	0	1	2	3	4	5	6	7
0行	x	X	y	=	z			
1行	H	u	e	:	<色相値>			

# 温度・湿度センサ

---

# 温度・湿度センサ

## ■ 温度・湿度センサ DHT11

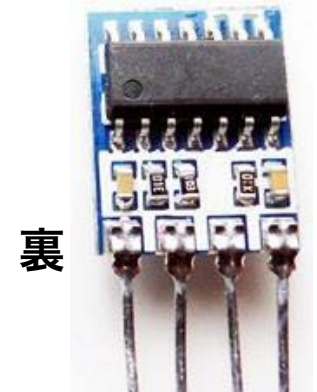
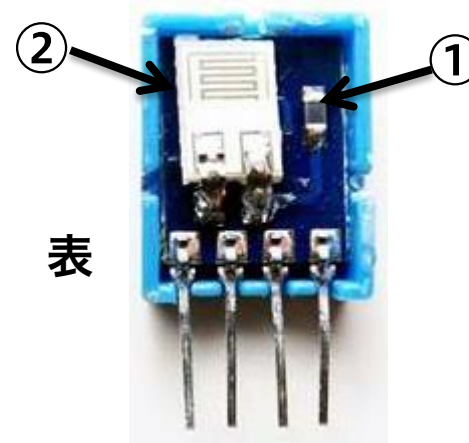
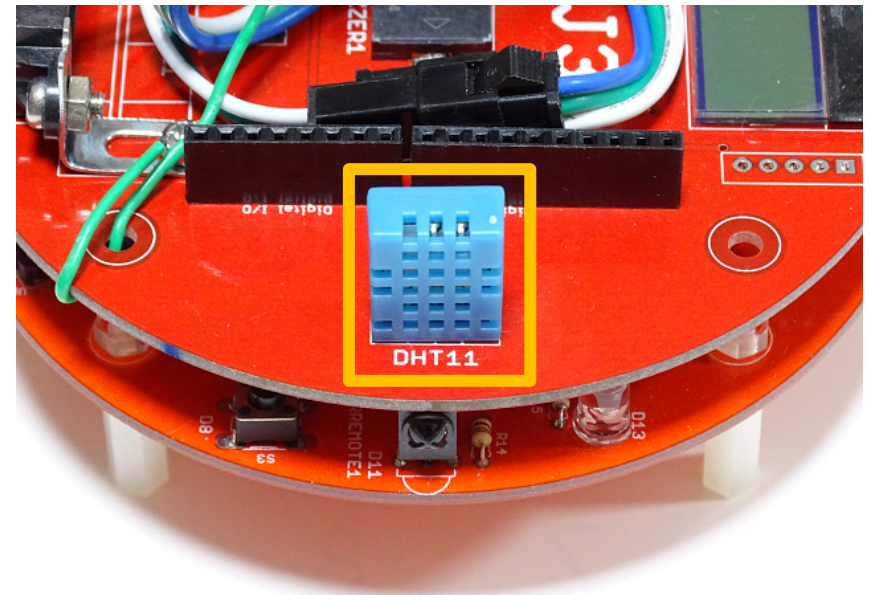
1つのパッケージに温度センサと湿度センサを内蔵

### ① 温度センサ：NTCサーミスタ

- 温度の変化を抵抗の変化で検出
- 金属酸化物（マンガン，ニッケル）にコバルト，鉄，銅などを混合して焼結したセラミックス
- 温度が高くなると抵抗値が下がる

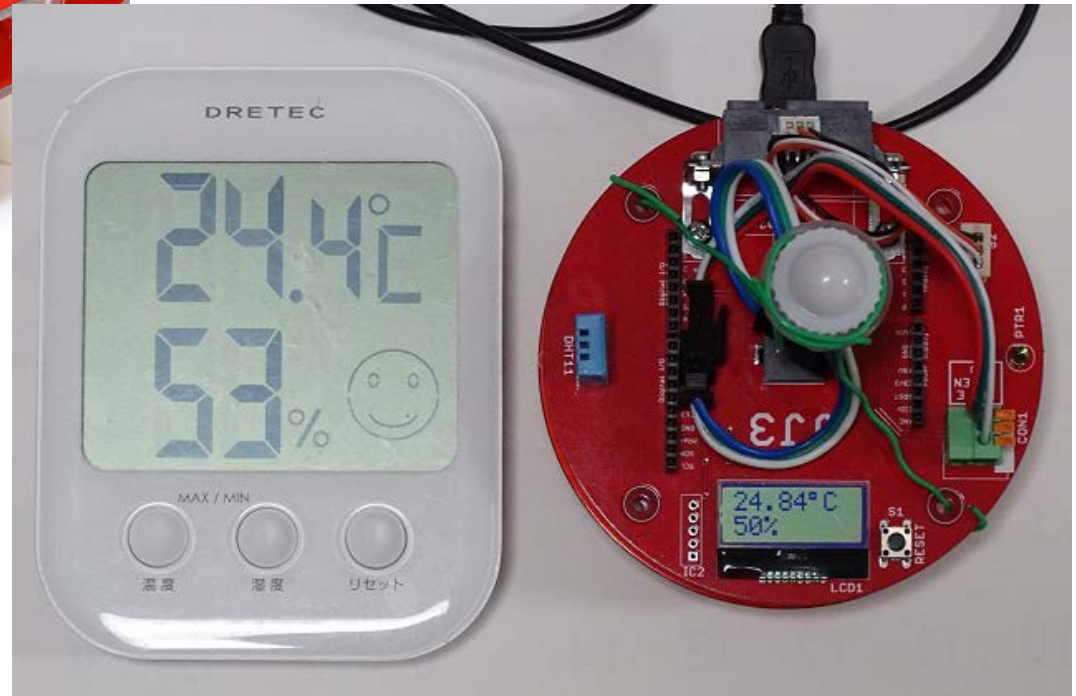
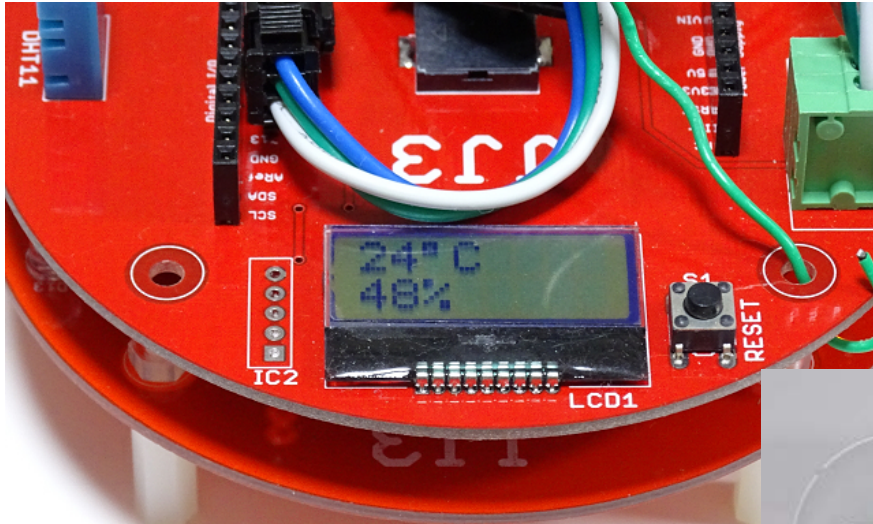
### ② 湿度センサ：抵抗式

- 湿度の変化を抵抗の変化で検出
- 電極にアンモニウム塩を含んだ感湿性高分子を塗布して，空気中の水蒸気に反応して抵抗が変化
- 湿度が高くなると抵抗値が下がる





# Example 905 : 温度・湿度の計測



# Example 905 : 温度・湿度の計測

```
#include <Wire.h>
#include <ST7032.h>
#include <DHT.h>
DHT dht(9, DHT11); // D9ピン, センサ型番
ST7032 lcd;
void setup() {
  lcd.begin(8, 2);
  lcd.setContrast(30);
  dht.begin(); // センサ計測開始
}
void loop() {
  int t = dht.readTemperature(); // 温度の計測
  int h = dht.readHumidity(); // 湿度の計測
  lcd.setCursor(0, 0); // 0列, 0行にセット
  lcd.print(t); lcd.print((char)0xDF);
  lcd.print("C");
  lcd.setCursor(0, 1); // 0列, 1行にセット
  lcd.print(h); lcd.print("%");
  delay(2000); // 2秒待つ
}
```

- `dht.readTemperature();`  
温度の計測関数
- `dht.readHumidity();`  
湿度の計測関数



# 温度・湿度センサとLED

---

# Example 906 : 体感温度の表示

## ■ 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 (°C)
- 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°C	通常過激な運動をしない
極度な注意	32-41°C	脱水症状の危険
危険	41-54°C	差し迫った熱中症
非常に危険	54°C以上	生命の危険

Metrication of Template:HeatTable

		temperature (°C)																
		27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
Relative Humidity (%)	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												



Wikipediaより転載

# 色の表現：HSV表色系

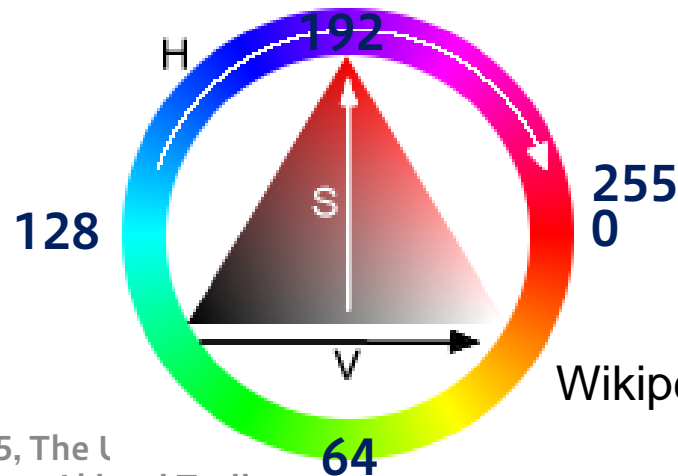
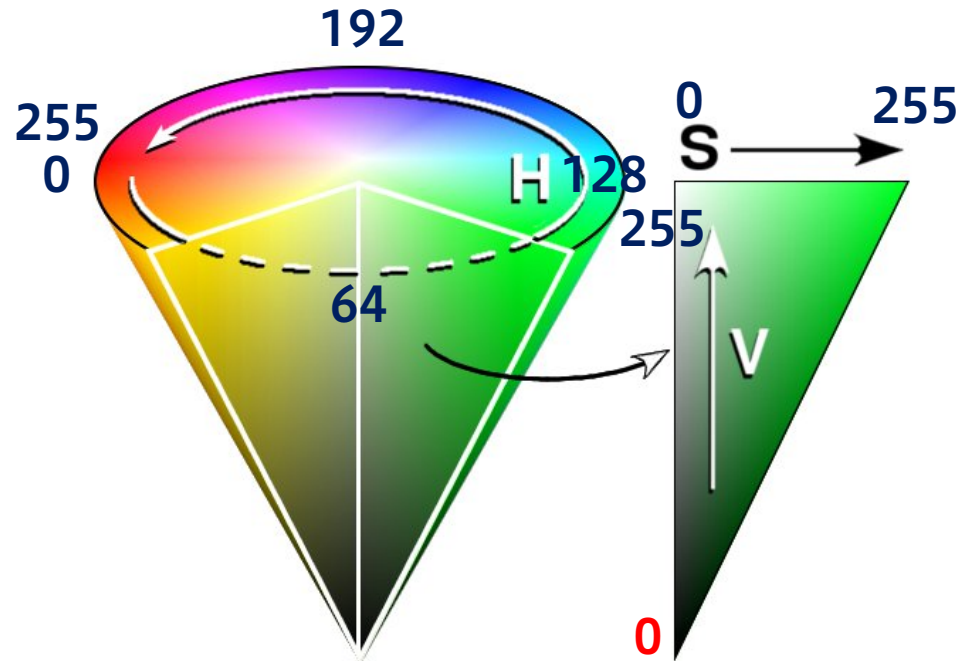
- 色相 (H: Hue)  
色の種類
- 彩度 (S: Saturation)  
色の鮮やかさ
- 明度 (V: Value, Brightness)  
色の明るさ

CHSV(h, s, v);

h値：0 - 255

s値：0 - 255

v値：0 - 255



Wikipediaより転載

# Example 906 : 体感温度の表示

```
long time; // 時間計測用変数
```

```
void setup() {  
  . . .
```

```
  time = millis(); // 時間を記録  
}
```

```
void loop() {  
  float hic;
```

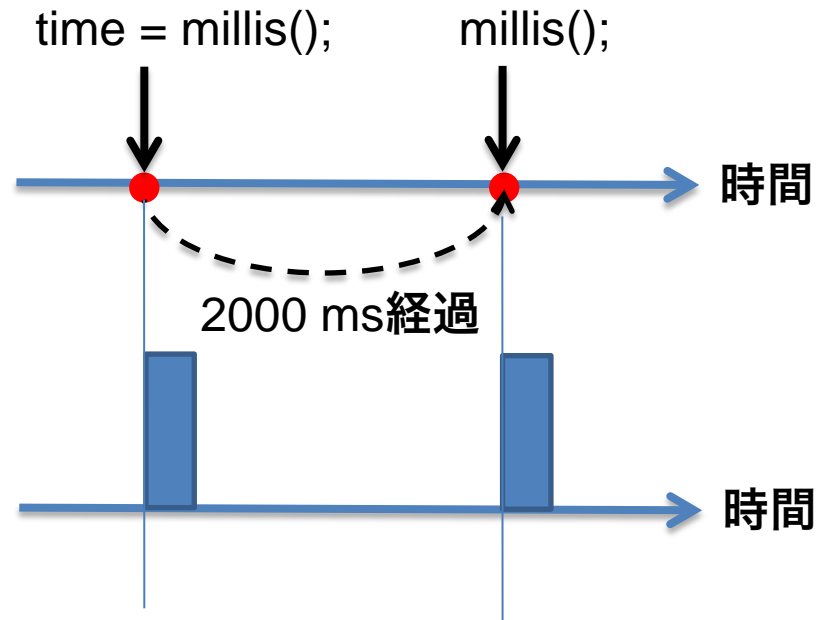
```
  if (millis() - time > 2000) {  
    float t = dht.readTemperature();  
    float h = dht.readHumidity();  
    hic = dht.computeHeatIndex(t, h, false);  
    time = millis(); // 時間を記録
```

```
    . . .
```

```
  }  
}
```

- `dht.computeHeatIndex(t, h, false);`  
ヒートインデックスの取得  
(温度tと湿度hから計算)

- `delay`関数を使わない計測方法  
`millis`関数を使い経過時間を調べる



# Example 906A : 不快指数の表示

## ■ 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 (°C) , H : 相対湿度 (%)

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°C
- 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°C

# 付録：ライブラリの関数一覧

---

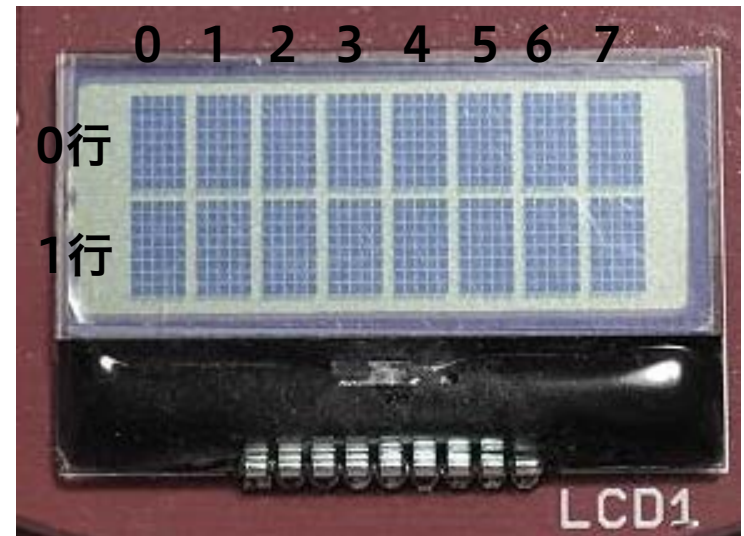
# 付録：液晶ディスプレイ (ACM0802) 用の関数

## ■ 関数一覧

```
void begin(uint8_t cols, uint8_t rows); // ST7032表示開始  
void setContrast(uint8_t cont); // コントラストの設定  
void setCursor(uint8_t col, uint8_t row); // カーソル位置の移動  
void clear(); // 表示のクリア  
void home(); // ホームに戻る  
void createChar(uint8_t location, uint8_t charmap[]); // 文字の作成  
virtual size_t write(uint8_t value); // 値 (文字) を書き込む
```

## ■ 文字の作成

```
byte heart[8] = {  
    0b00000,    lcd.createChar(num, heart);  
    0b01010,    lcd.write(num);  
    0b11111,  
    0b11111,  
    0b11111,  
    0b01110,  
    0b00100,  
    0b00000};
```





# 付録：温度・湿度センサ(DHT11)用の関数

## ■ 関数一覧

```
DHT(uint8_t pin, uint8_t type, uint8_t count=6); // ピン番号, センサ名
void begin(void); // 計測を開始する
float readTemperature(bool S=false); // 温度の計測結果を読み込む
float convertCtoF(float); // °CからFに変換
float convertFtoC(float); // Fから°Cに変換
float computeHeatIndex(temperature, percentHumidity, isFahrenheit=true);
// 温度と湿度からヒートインデックスに変換
float readHumidity(void); // 湿度の計測結果を読み込む
```

# 付録：FastLED用の関数

## ■ 関数一覧

```
void setBrightness(uint8_t scale); // 明るさの最大値(0-255)を設定する  
void show(); // LEDの表示を更新する  
void clear(); // LEDの表示を消す  
void showColor(const struct CRGB & color); // RGB色を表示する  
void setTemperature(const struct CRGB & temp); // 色温度を表示する
```

## ■ 色の指定方法

```
leds[].r = val;          leds[].setRGB(r, g, b);          <color>:  
leds[].g = val;          leds[] = CRGB(r, g, b);          FastLEDライブラリ  
leds[].b = val;                                     pixeltypes.hに定義  
  
leds[] = CRGB::<color>   leds[].setHSV(h, s, v);  
                          leds[] = CHSV(h, s, v);
```

```
fill_solid(&(leds[0]), NUM_LEDS, CHSV(h, s, v));  
fill_solid(&(leds[0]), NUM_LEDS, CRGB(r, g, b));
```

val, r, g, b, h, s, v: 0~255