気象モニターを作ろう（基礎編）

ー誰にでもできるプロトタイピングー

第４回　フルカラー色LED(テープ LED)を使う

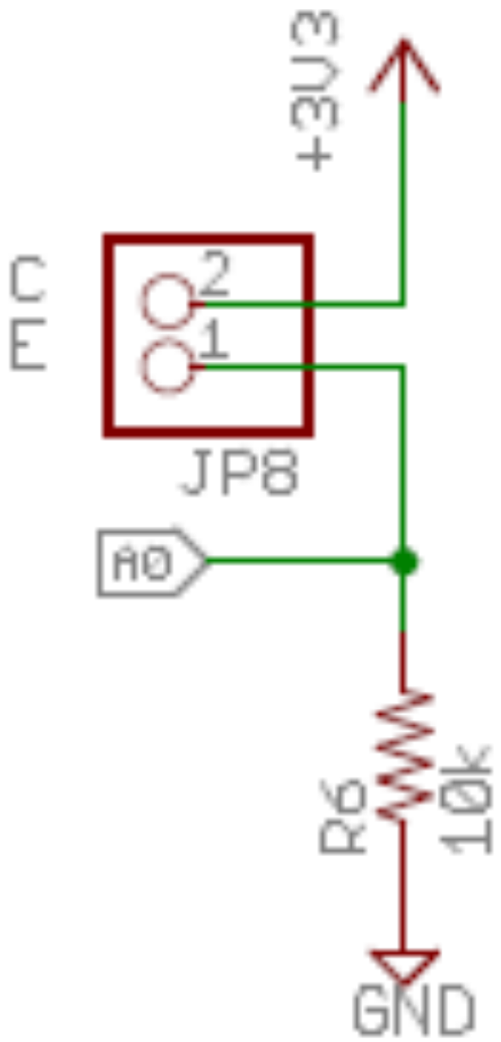
http://cms.db.tokushima-u.ac.jp/DAV/person/S10723/気象モニターを作ろう/

川上　博

2015/06/11

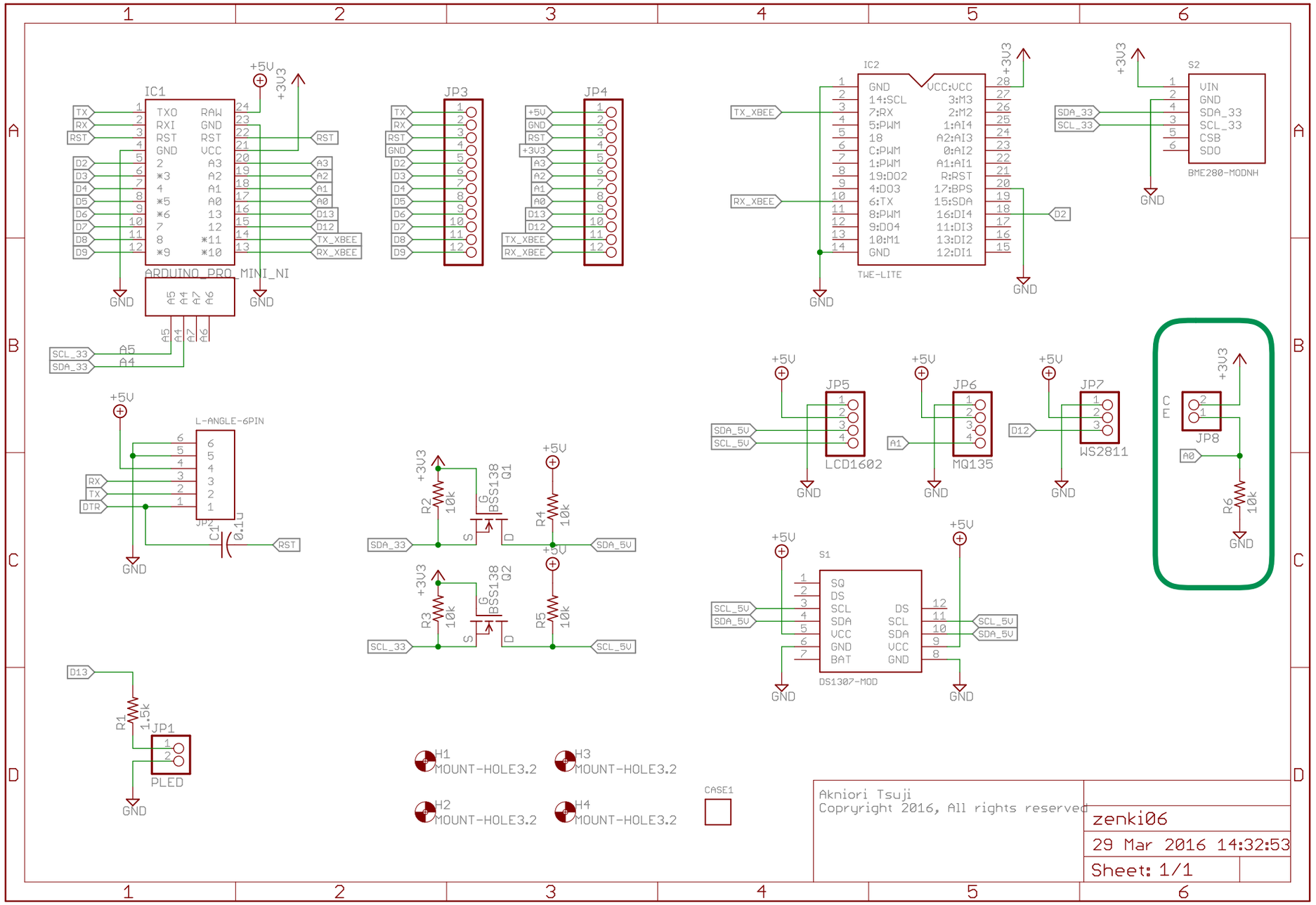
前回の質問：アナログ入力ピンA0

analogRead(0)とanalogRead(1)で値が異なる理由



A0には，基板上に既に左図のように10kの抵抗がGNDに接続されています．したがってブレッド・ボードで同じ回路を組むと，5kオームの抵抗となって出力電圧が小さくなります．

KOUKAIKOUZA2016/Schematic/JJ4-zenki-sch.pdf

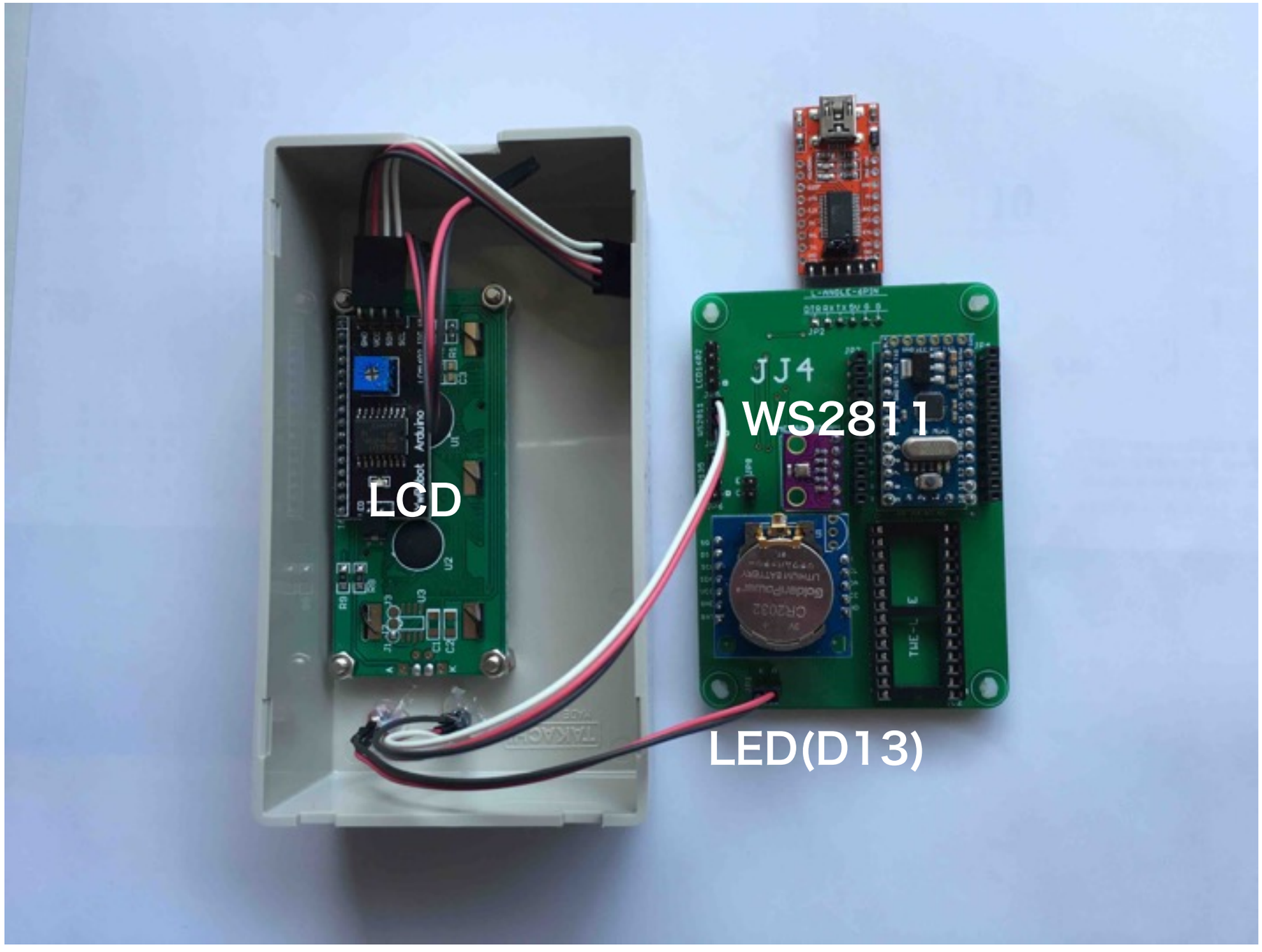


今日のテーマ

IC(WS2811)内蔵LED・テープLEDを点灯する

ポート	接続先	型番	ライブラリ	注釈
D0	USB シリアル変換	FT232RL		予約
D1	USB シリアル変換	FT232RL		予約
D2-D9	ー	ー	ー	空き
D10	ZigBee 無線(親機)*1	TWE-LITE	SoftwareSerial	19200bps
D11	ZigBee 無線(親機)*1	TWE-LITE	SoftwareSerial	19200bps
D12	フルカラーLED (WS2811)	PL9823-F5	FastLED	
D13	LED(赤)5mm	OSDR5113A	digitalWrite	
A0	照度センサ	NJL7502L	analogRead	
A1	温度センサ（アナログ）	LM61CIZ	analogRead	ブレッドボード
A2-A3	ー	ー	ー	空き
A4(SDA) A5(SCL)	液晶ディスプレイ 16x2	LCD1602	LiquidCrystal_I2C	I2C(5V)
A4(SDA) A5(SCL)	温度・湿度・大気圧センサ（デジタル）	BME280	BME280	I2C(3.3V)
A4(SDA) A5(SCL)	リアルタイムクロック	DS1307	RTCLib	I2C(5V)

JJ4 での接続

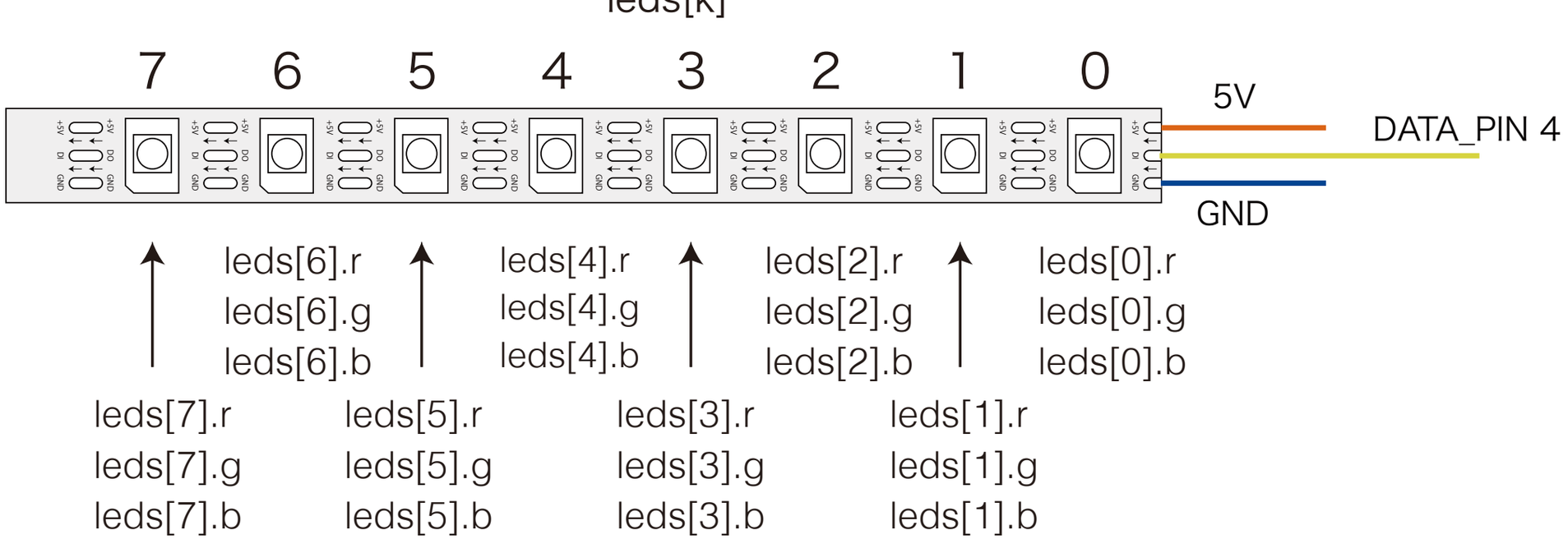


テープLEDの名前と番号付け

```
#include “FastLED.h”
```
テープLEDのヘッダーファイル

```
#define NUM_LEDS 1
#define DATA_PIN 12
```
テープLEDの数：1個
データ入力のピン番号

```
CRGB leds[NUM_LEDS];
```
テープLEDの名前 leds の配列




```
int a[5];
```
配列の例：整数の変数 a[0], a[1], a[2], a[3], a[4] を定義する

配列：array

a0, a1, a2, a3, a4

$a_0,\ a_1,\ a_2,\ a_3,\ a_4$

$a[0],\ a[1],\ a[2],\ a[3],\ a[4]$

```
int a[5]; // 配列aを定義し，５個変数を取る
for (i=0; i < 5; i++) {
    a[i] = 2* i;
}
```

スケッチ（プログラム）の基本構造

```
// Example401A
// simple blinck exmaple
// H. Kawakami, June 2016

#include "FastLED.h"

#define NUM_LEDS 1    // const int NUM_LEDS=1;
#define DATA_PIN 12   // const int DATA_PIN=12;

CRGB leds[NUM_LEDS];
```

定数，変数の定義（大域）

```
void setup() {
  delay(2000);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
}
```

初期設定（一度だけ実行）

```
void loop() {
  leds[0].setRGB(255,68,221);
  FastLED.show(); delay(1000);
  leds[0].setRGB(0,0,0);
  FastLED.show(); delay(1000);
}
```

実行させる仕事（繰り返し実行）

IC(WS2811)内蔵LED・テープLEDを点灯する

leds[0].setRGB(255,68,221);

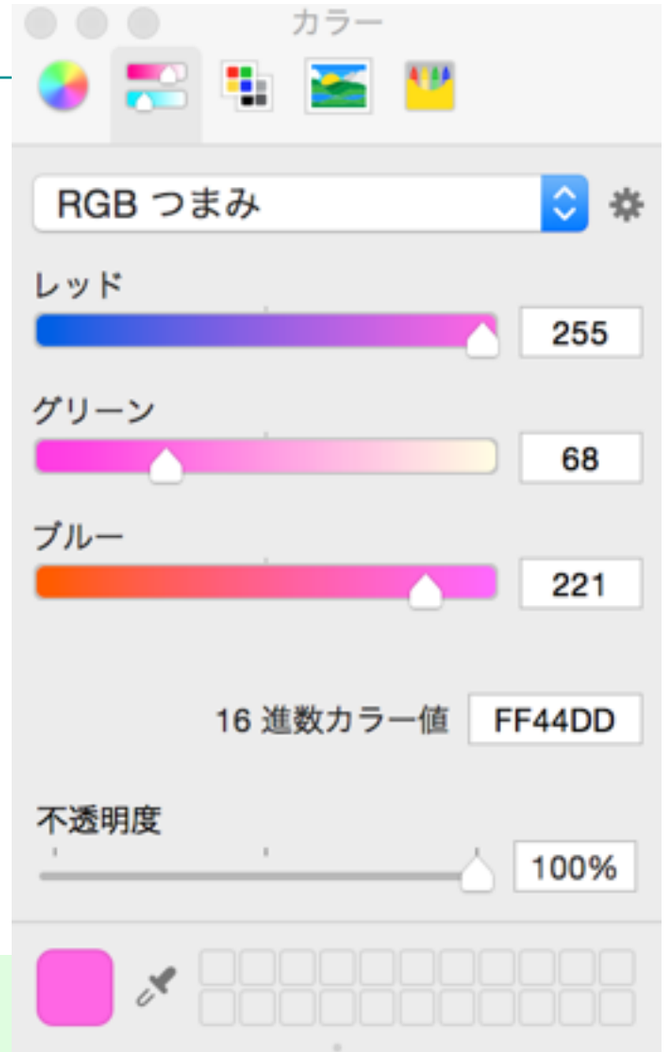
G: green 0 - 255

R: red 0 - 255

B: blue 0 - 255

2進数８桁をbyteという（16進数2桁，０～255の数）

RGB色指定の方法は，数種類ある




```
// Example 401B  simple blink example
// different kind of color setting
// The predefined colors list is found at:
// https://github.com/FastLED/FastLED/wiki/Pixel-reference
// H. Kawakami, June 2016

#include "FastLED.h"

#define NUM_LEDS 1
#define DATA_PIN 12
CRGB leds[NUM_LEDS];

void setup() {
  delay(2000);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
}

void loop() {
//    leds[0].setRGB(255, 68, 221);
//    leds[0].r=255; leds[0].g=68; leds[0].b=221;
//    leds[0]=CRGB::HotPink;  //HotPink:FF69B4
//    leds[0]=0xFF44DD;
    fill_solid(&(leds[0]), 1,  CRGB(255, 68, 221));
    FastLED.show(); delay(1000);

    leds[0].setRGB(0, 0, 0);
    FastLED.show(); delay(1000);
}
```

11

２進数，16進数，そして勿論10進数

たとえば，10進数の２３

23 = 2x10^1 + 3x10^0 = 23(dec)
= 1x16^1 + 7x16^0 = 17(hex)
= 1x2^4 + 1x2^2 + 1x2^1 + 1x2^0
= 10111(binary)

$$a_m 16^m + a_{m-1} 16^{m-1} + \cdots + a_2 16^2 + a_1 16^1 + a_0 16^0$$

$$a_m 10^m + a_{m-1} 10^{m-1} + \cdots + a_2 10^2 + a_1 10^1 + a_0 10^0$$

$$a_m 2^m + a_{m-1} 2^{m-1} + \cdots + a_2 2^2 + a_1 2^1 + a_0 2^0$$

0～15の数の表記法

16進数	2進数	10進数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

16進数	2進数	10進数
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

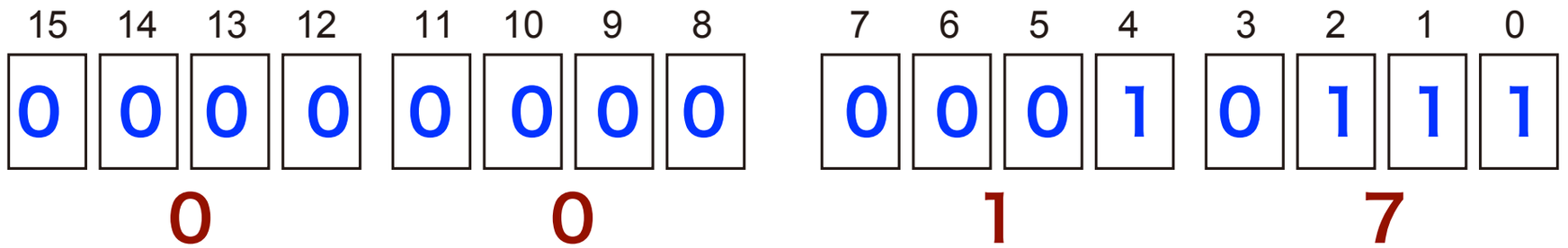
16進数：0xFF44DD

FF=15*16+15=240+15=255, 44=4*16+4=68,
DD=13*16+13=208+13=221

マイコンの中での数の記憶

- マイコンの中では，変数は2進数として蓄えられている

int x=23;



- 2進数は，0と1のみで表現できるが，桁が大きくなる

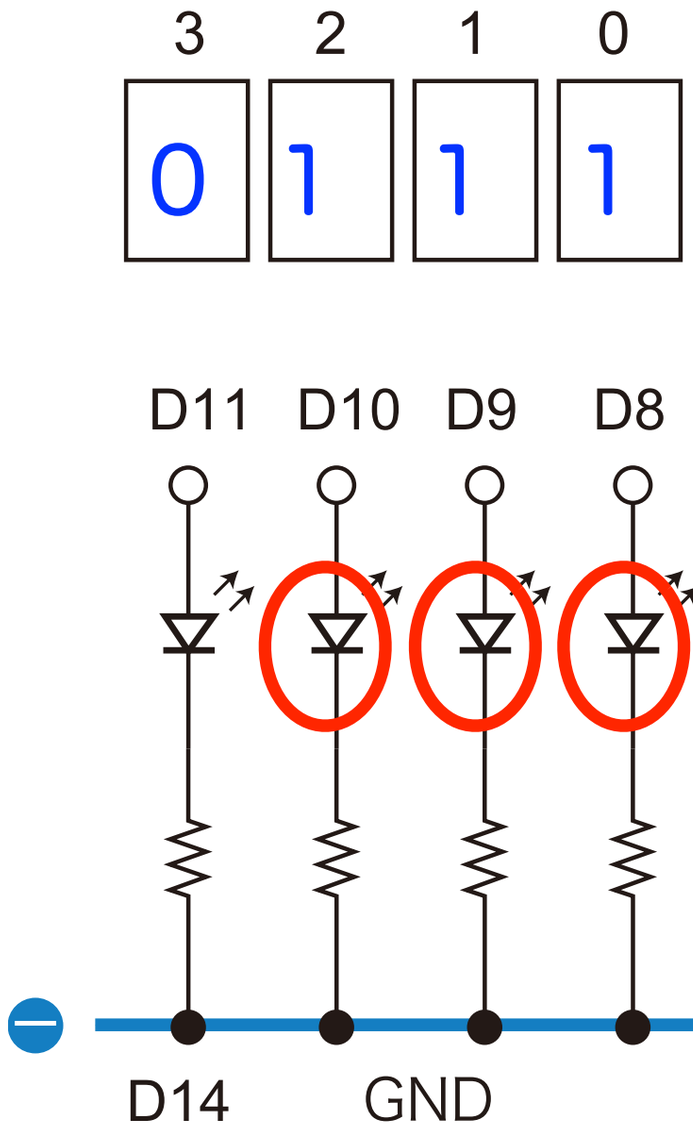
- 2進数１桁をbitという（binary digitからの造語?）
- 2進数４桁をnibbleという（16進数１桁，hexa decimal）
- 2進数８桁をbyteという（16進数2桁，０～255の数）

- hexa decimal : 0x0017

nibble display

x = 7 = 0x2^3 + 1x2^2 + 1x2^1 + 1x2^0

int x=7;



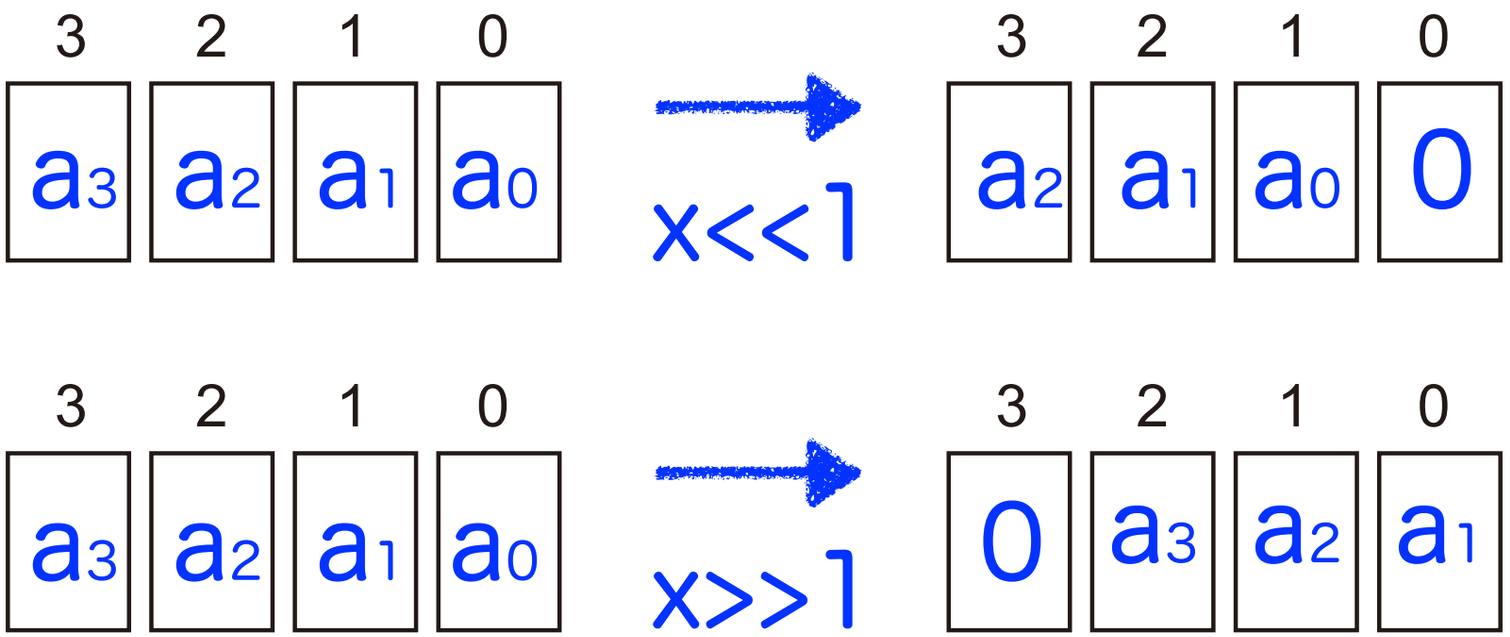
`bitRead(x, i)`　xのビットiの値を読む

2進数を2倍する，あるいは2で割る

$$x = a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0$$

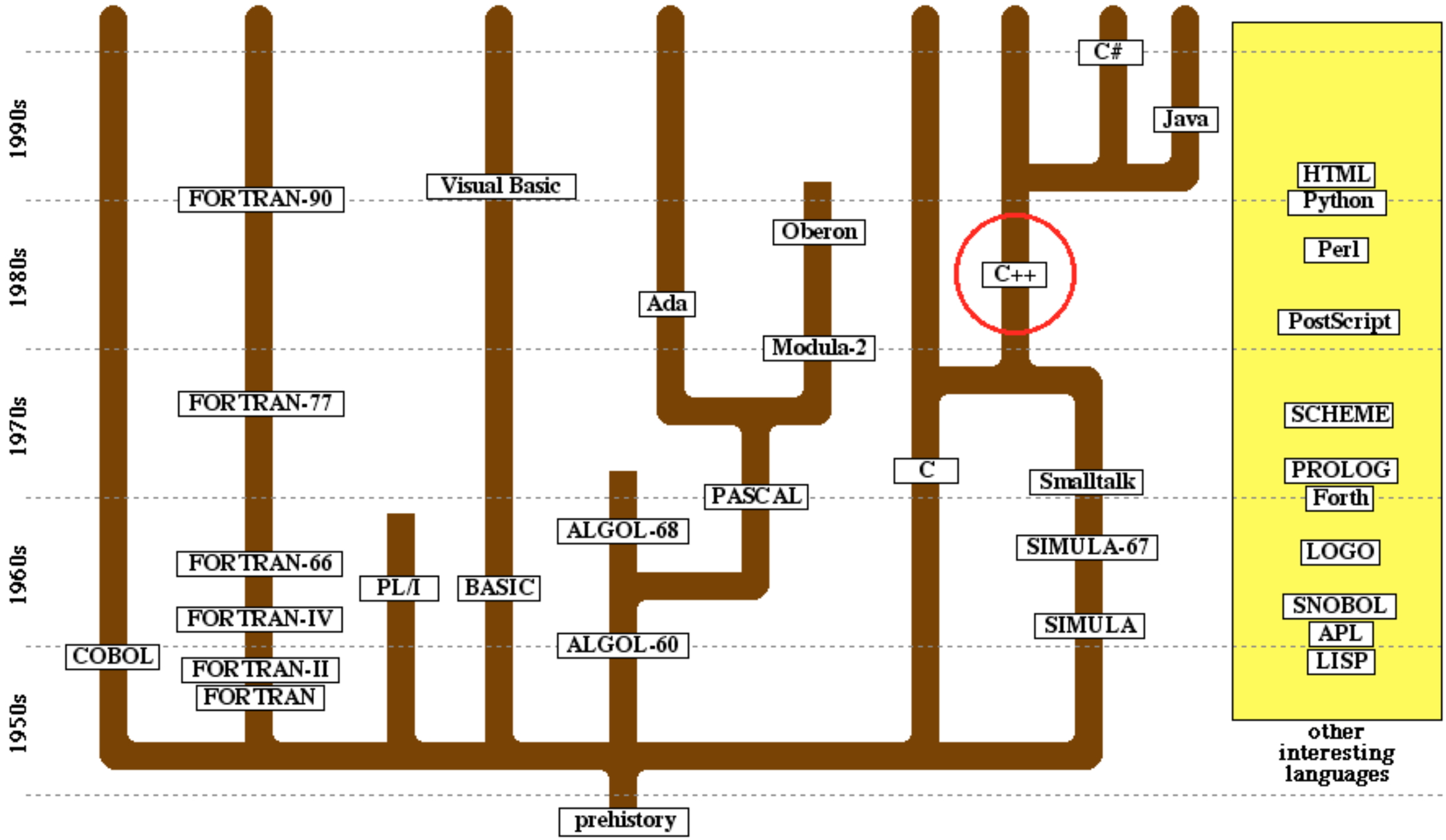
$$2x = a_3 2^4 + a_2 2^3 + a_1 2^2 + a_0 2^1$$

$$x/2 = a_3 2^2 + a_2 2^1 + a_1 2^0 + a_0 2^{-1}$$



主なプログラミング言語の発展木

http://www-cs-faculty.stanford.edu/~eroberts/books/ArtAndScienceOfJava/
から引用



RGB色指定の補正：calibration

```
// Example402A
// CRGB calibration
// H. Kawakami, June 2016

#include "FastLED.h"

#define NUM_LEDS 1
#define DATA_PIN 12

CRGB leds[NUM_LEDS];

void setup() {
    delay(2000);
    FastLED.addLeds<WS2811, DATA_PIN, GRB>(leds, NUM_LEDS);
}

void loop(){
   leds[0].setRGB(255,0,0); // red
   FastLED.show(); delay(1000);
   leds[0].setRGB(0,255,0); // green
   FastLED.show(); delay(1000);
   leds[0].setRGB(0,0,255); // blue
   FastLED.show(); delay(1000);
   leds[0].setRGB(255,255,255); //white
   FastLED.show(); delay(1000);
}
```

ここを変える：GRBなど

この順序に点灯するか？

Example402Aと同じ
switch 文に注意
（場合分けが多い場合に便利）

- 変数 leds の配列を宣言（１個）: leds[0]
- 添え字変数は，0から始まる

```
// Example402B
// another CRGB calibration
// H. Kawakami, June 2016

#include "FastLED.h"

#define NUM_LEDS 1
#define PIN 12

CRGB leds[NUM_LEDS];
int br=25;

void setup(){
  delay(2000);
  FastLED.addLeds<WS2811, PIN, RGB>(leds, NUM_LEDS);
}

void loop() {
  for(int i = 0; i < 4; i++) {
      memset(leds, 0, NUM_LEDS * 3);
      switch(i) {
        case 0: leds[0].setRGB(br, 0, 0); break;
        case 1: leds[0].setRGB(0, br, 0); break;
        case 2: leds[0].setRGB(0, 0, br); break;
        case 3: leds[0].setRGB(br, br, br); break;
        default: break;
      }
      FastLED.show();
      delay(1000);
  }
}
```

HSV(HSB)：色相・彩度・明度

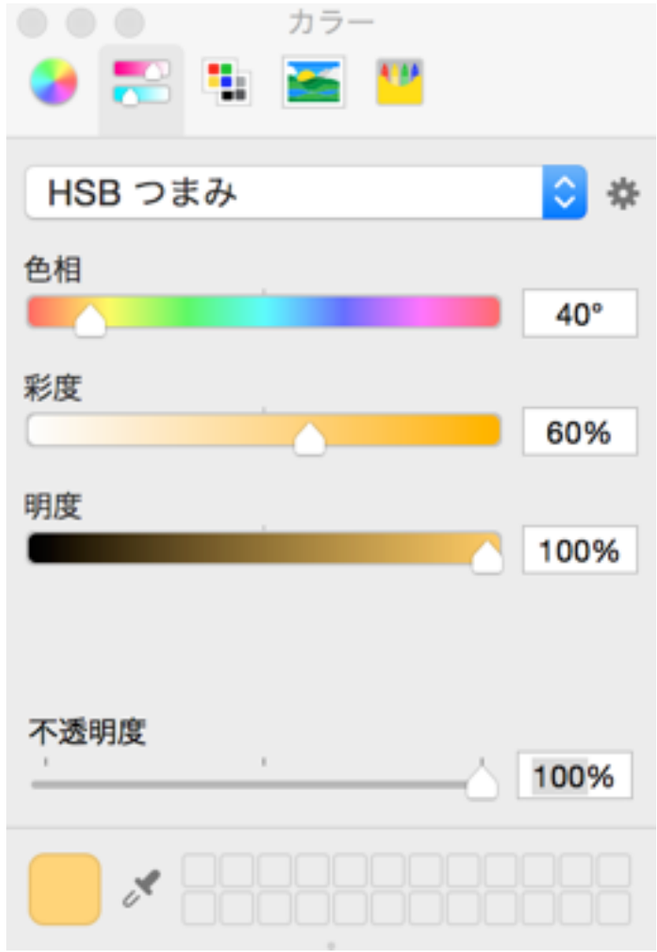
```
// Example 803A  simple blink by HSV color setting
// see https://github.com/FastLED/FastLED/wiki/Controlling-leds
// H. Kawakami, June 2016

#include <FastLED.h>

#define NUM_LEDS 1
#define DATA_PIN 12

CRGB leds[NUM_LEDS];

void setup() {
  delay(2000);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
}

void loop() {
   leds[0].setHSV(221, 51, 120);
//    leds[0]=CHSV(221, 51, 120);
//    leds[0].setHue(221);
//    fill_solid(&(leds[0]), 1,  CHSV(221, 51, 120));
    FastLED.show();
    delay(1000);
    leds[0].setHSV(0, 0, 0);
    FastLED.show();
    delay(1000);
}
```

FastLED

0 - 255

0 - 255

0 - 255



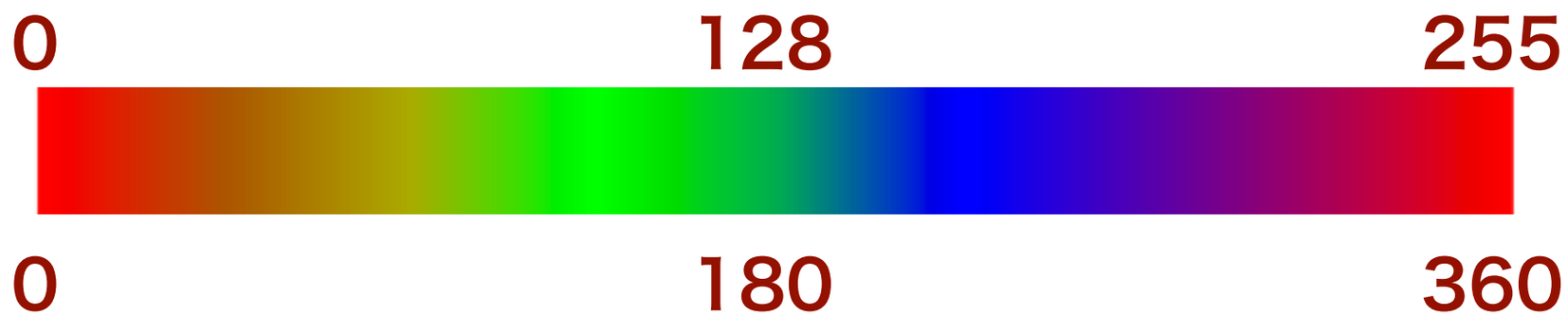
HSV : Hue, Saturation, Value

(HSB : Hue, Saturation, Brightness)

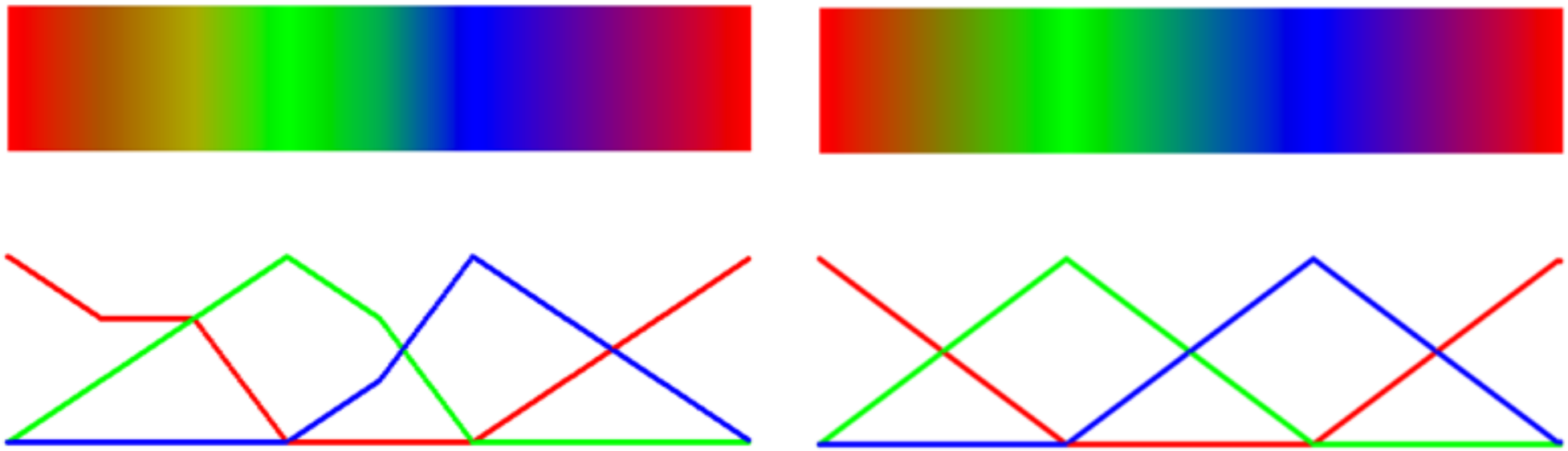
色相，彩度（鮮やかさ），明度（明るさ）

HSV(HSB)：色相・彩度・明度

```
// Example803B
// Color setting by Hue
// H. Kawakami, June 2016

#include “FastLED.h"

const int NUM_LEDS = 1;
const int DATA_PIN = 12;
CRGB leds[NUM_LEDS];

void setup() {
  delay(2000);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
}

void loop() {
  for (int i=0; i < 256; i++) {
    fill_solid(&(leds[0]), NUM_LEDS, CHSV(i, 150, 150));
    FastLED.show();
    delay(50);
  }
}
```



HSV(HSB)：rainbow vs spectrum



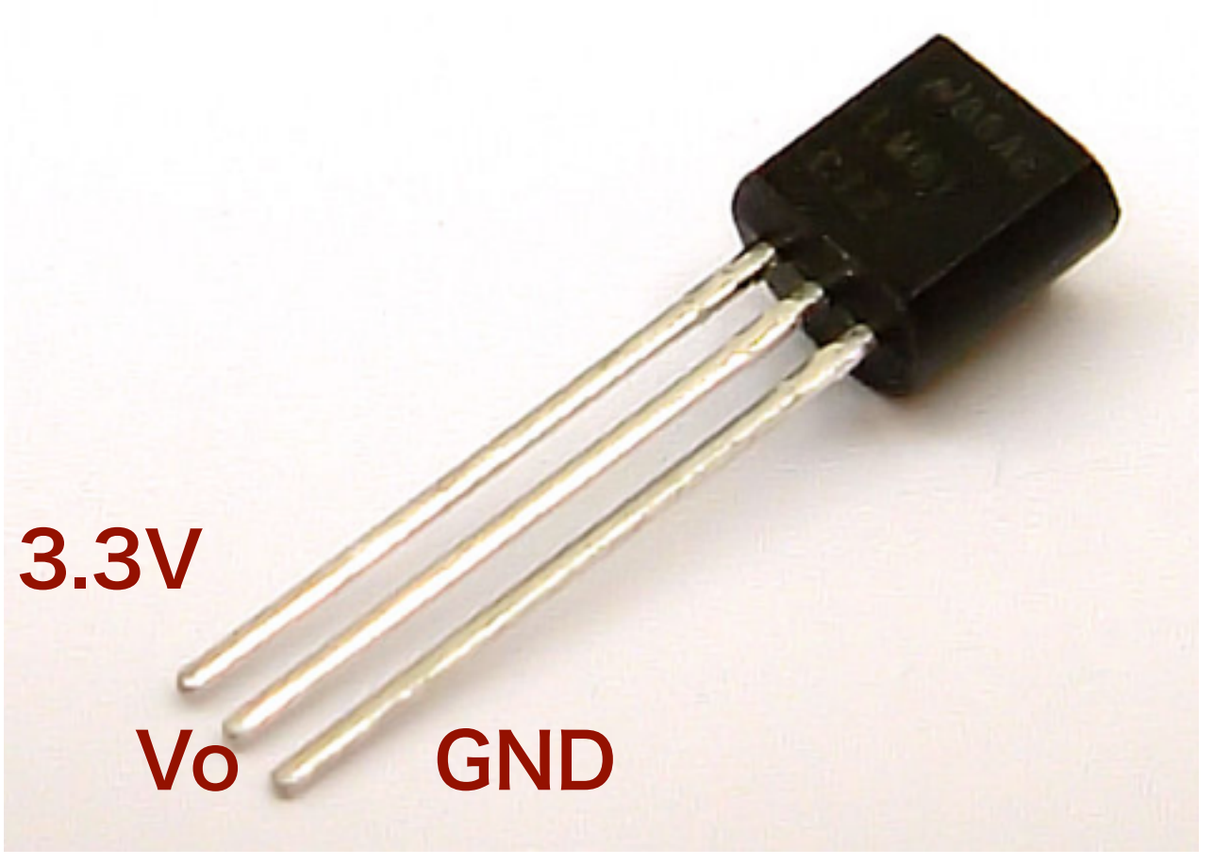
rainbow color system
(default color system)

spectrum color system

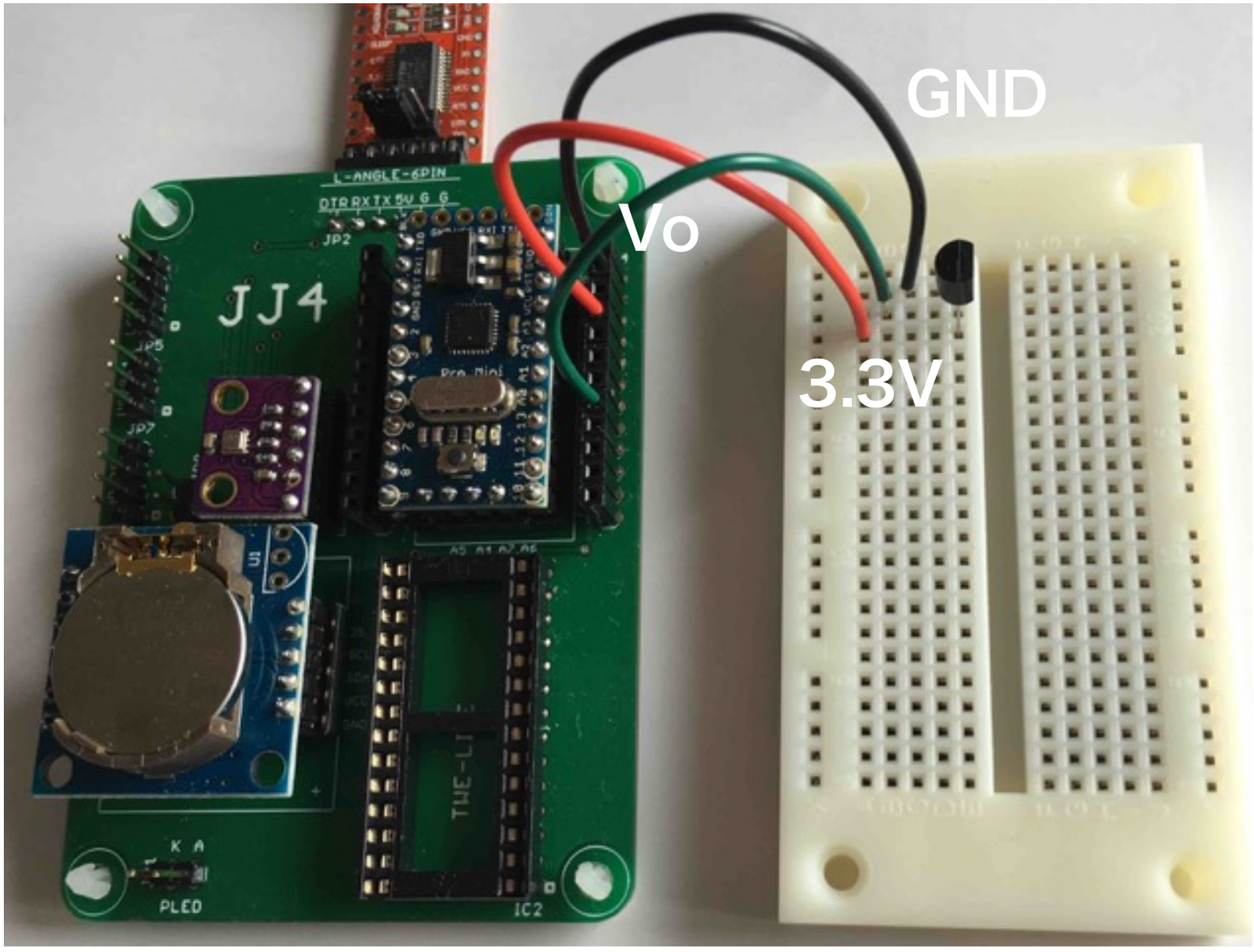
```
CHSV scolor;

scolor.hue = 221;
scolor.saturation = 51;
scolor.value = 120;
hsv2rgb_spectrum(scolor, leds[0]);
```

温度センサーを使ってLEDを調光する

温度センサー　LM61CIZ



Voをアナログ・ピンA0に接続


GND
Vo
3.3V
L-ANGLE-6PIN
DTR RX TX 5V G G
JP2
JJ4
JP5
JP7
Pro Mini
A5 A4 A7 A6
U1
K A
PLED
TWE-LI
IC2

Example 404A

```
/* Example404A
 * temperature indicator
 */

const int inPin=0;
float vo, temp;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int data=analogRead(inPin);
  Serial.println(data);
  vo=data*3.3/1024.0;
  temp=(1000.0*vo-600.0)/10.0;
  Serial.println(temp);
  delay(1000);
}
```



温度の範囲によって配色を変える

```
if(   ){
   文
}else{
   文
}
```

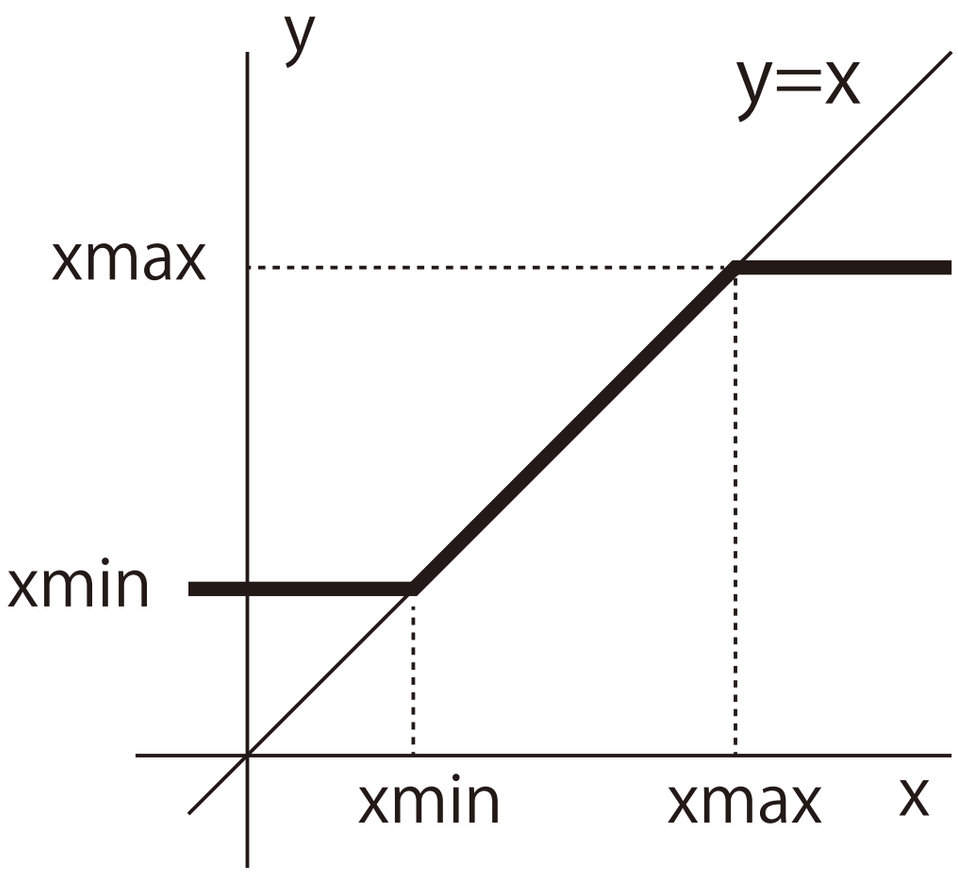
```
/* Example404B
 * temperature indicator
 */
#include <FastLED.h>
#define NUM_LEDS 1
#define DATA_PIN 12
CRGB leds[NUM_LEDS];
const int inPin=0;
float vo, temp, tmin=25.0, tmax=28.0;

void setup() {
  Serial.begin(9600);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
}
```

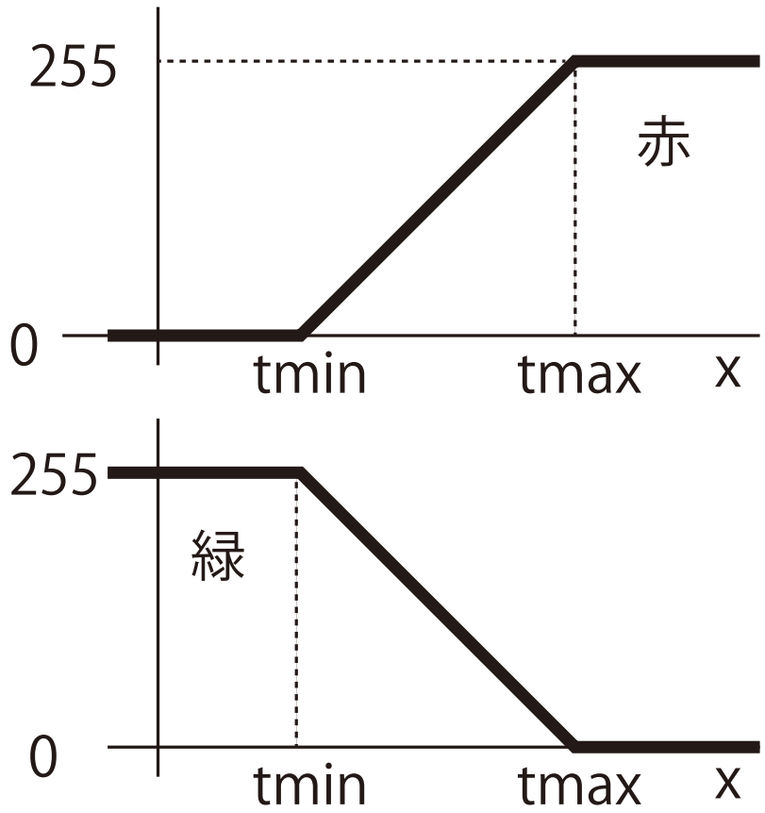
```
void loop() {
  int data=analogRead(inPin);
  Serial.println(data);
  vo=data*3.3/1024.0;
  temp=(1000.0*vo-600.0)/10.0;
  Serial.println(temp);
  if(temp < tmin){
    leds[0].setRGB(0, 255, 0);
  }else if(temp < tmax){
    leds[0].setRGB(255, 255, 0);
  } else{
    leds[0].setRGB(255, 0, 0);
  }
  FastLED.show();
  delay(1000);
}
```

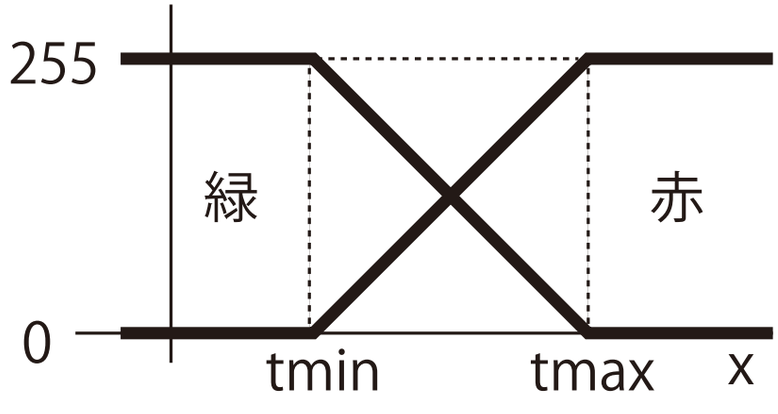
```
/* Example404C
 * temperature indicator
 */
#include <FastLED.h>

#define NUM_LEDS 1
#define DATA_PIN 12
CRGB leds[NUM_LEDS];
const int inPin=0;
float vo, temp, tmin=25.0, tmax=28.0, tm;

void setup() {
  Serial.begin(9600);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
  tm=tmax-tmin;
}

void loop() {
  int data=analogRead(inPin);
  Serial.println(data);
  vo=data*3.3/1024.0;
  temp=(1000.0*vo-600.0)/10.0;
  temp=constrain(temp, tmin, tmax);
  Serial.println(temp);
  leds[0].setRGB(255*(temp-tmin)/tm, 255*(tmax-temp)/tm, 0);
  FastLED.show();
  delay(1000);
}
```

contrain関数と配色のための関数

```
temp=constrain(temp, tmin, tmax);
leds[0].setRGB(255*(temp-tmin)/tm, 255*(tmax-temp)/tm, 0);
```







照度によって色相が変わるスケッチを書いてみよう

```
// Example405C: HSVTestWithSensor00
// Sensor = A0: Photo transistor

#include "FastLED.h"

#define NUM_LEDS 1
#define DATA_PIN 12
#define SENSOR_PIN 0

CRGB leds[NUM_LEDS];
int sensor;

void setup() {
  Serial.begin(9600);
  FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
}

void loop() {
  sensor=analogRead(SENSOR_PIN);
  Serial.println(sensor);
  sensor=map(sensor, 1023, 0, 0, 255);
  Serial.println(sensor);
  fill_solid(&(leds[0]), NUM_LEDS, CHSV(sensor, 255, 150));
  FastLED.show();
}
```