

LEDを光らせよう

第8回 テープLEDを使ってみよう

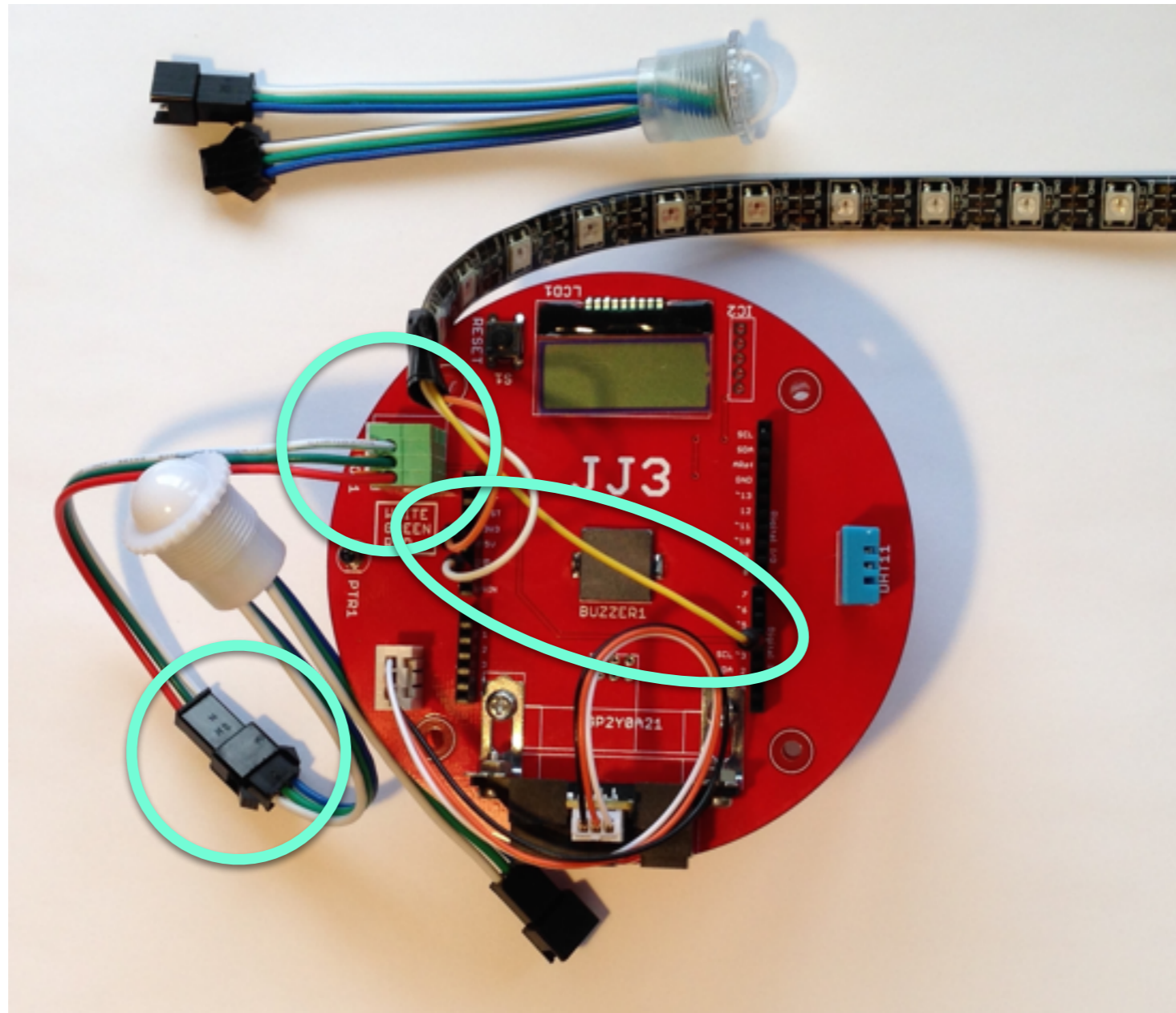
<http://cms.db.tokushima-u.ac.jp/DAV/person/S10723/LEDを光らせよう/>

川上 博

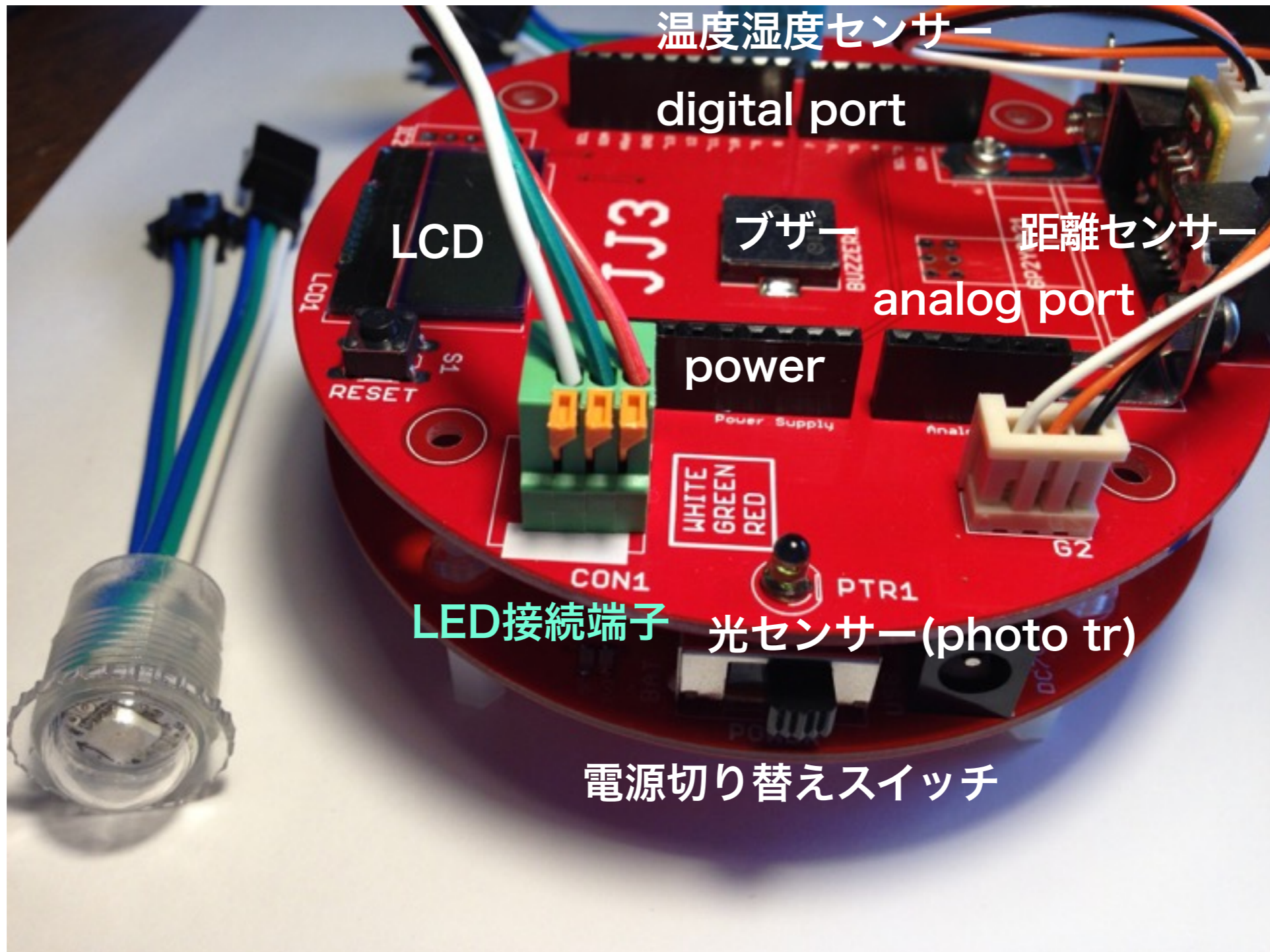
2015/10/10

今日のテーマ

IC(WS2811)内蔵LED・テープLEDを点灯する



JJ3 シールド



テープLEDの名前と番号付け

```
#include "FastLED.h"
```

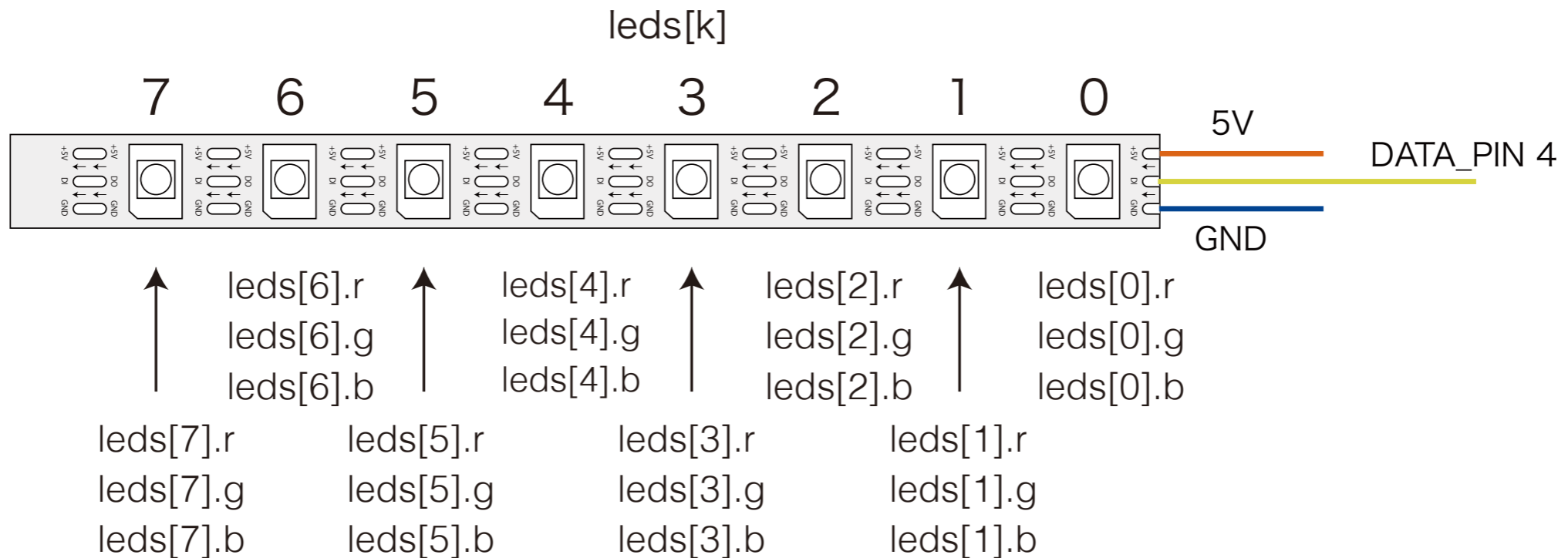
テープLEDのヘッダーファイル

```
#define NUM_LEDS 10
#define DATA_PIN 4
```

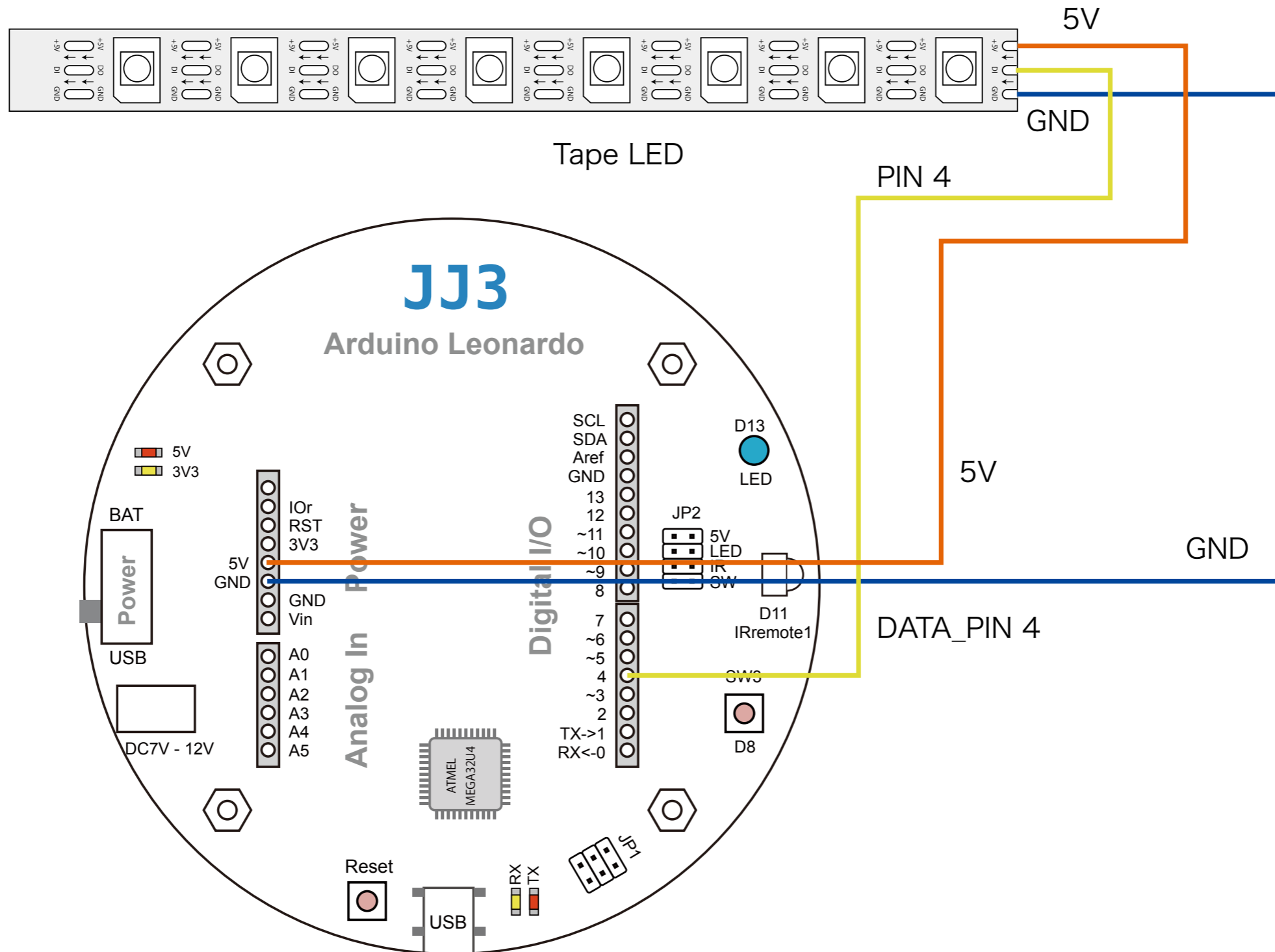
テープLEDの数：10個[0 - 9]
データ入力のピン番号

```
CRGB leds[NUM_LEDS];
```

テープLEDの名前 leds の配列



テープLEDとArduino間の配線



スケッチ (プログラム) の基本構造

```
// Example801A
// simple blinck exmaple
// H. Kawakami, October 2015
```

```
#include "FastLED.h"
```

```
#define NUM_LEDS 1
#define DATA_PIN 4
```

```
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() {
  for(int i=0; i<NUM_LEDS; i++){
    leds[i].setRGB(255,68,221);
    FastLED.show(); delay(1000);
    leds[i].setRGB(0,0,0);
    FastLED.show(); delay(1000);
  }
}
```

実行させる仕事 (繰り返し実行)

スケッチ（プログラム）の基本構造

```
const int led;  
int i, j;  
float x, y;
```

定数, 変数の定義 (大域)

```
void setup() {
```

初期設定 (一度だけ実行)

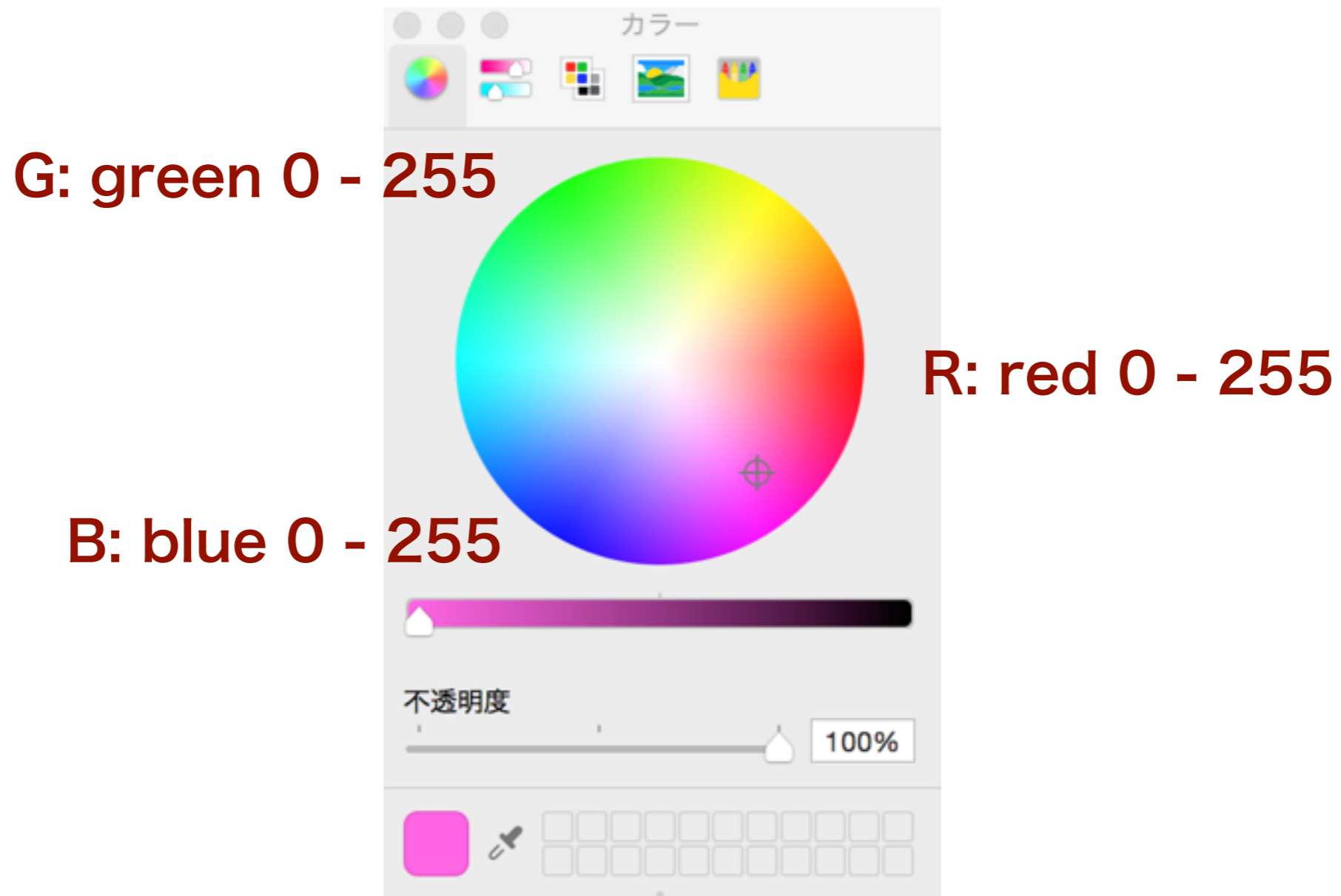
```
}
```

```
void loop() {
```

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



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 801B 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, October 2015
```

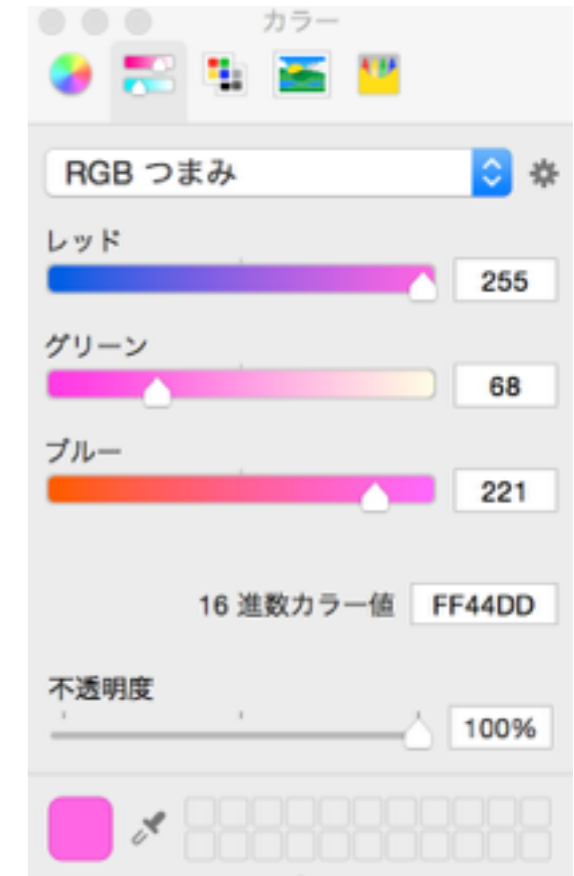
```
#include "FastLED.h"
```

```
#define NUM_LEDS 1
#define DATA_PIN 4
CRGB leds[NUM_LEDS];
```

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

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

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



0~15の数の表記法

16進数	2進数	10進数		16進数	2進数	10進数
0	0000	0		8	1000	8
1	0001	1		9	1001	9
2	0010	2		A	1010	10
3	0011	3		B	1011	11
4	0100	4		C	1100	12
5	0101	5		D	1101	13
6	0110	6		E	1110	14
7	0111	7		F	1111	15

16進数 : 0xFF44DD

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

RGB色指定の補正 : calibration

```
// Example802A
// CRGB calibration
// H. Kawakami, October 2015
```

```
#include "FastLED.h"
```

```
#define NUM_LEDS 1
#define DATA_PIN 4
```

```
CRGB leds[NUM_LEDS];
```

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

RGB色指定の補正 : calibration

```
// Example802C
// calibration of two LED strings on pins 4 and 6
// H. Kawakami, October 2015

#include "FastLED.h"

#define NUM_LEDS 10
#define DATA_PIN1 4
#define DATA_PIN2 6

CRGB leds1[NUM_LEDS]; CRGB leds2[NUM_LEDS];
int br=25;

void setup(){
  delay(2000);
  FastLED.addLeds<WS2811, DATA_PIN1, RGB>(leds1, NUM_LEDS);
  FastLED.addLeds<WS2811, DATA_PIN2, GRB>(leds2, NUM_LEDS);
}

void loop() {
  for(int i = 0; i < 3; i++) {
    memset(leds1, 0, NUM_LEDS * 3); memset(leds2, 0, NUM_LEDS * 3);
    for(int j=0; j < NUM_LEDS; j++){
      switch(i) {
        case 0:
          leds1[j].setRGB(br, 0, 0); leds2[j].setRGB(br, 0, 0); break;
        case 1:
          leds1[j].setRGB(0, br, 0); leds2[j].setRGB(0, br, 0); break;
        case 2:
          leds1[j].setRGB(0, 0, br); leds2[j].setRGB(0, 0, br); break;
        default: break;
      }
      FastLED.show();
      delay(1000);
    }
  }
}
```

LED毎にすべて2組用意する

GRBと順序を変える

テープLEDの点灯位置を変化させる

```
// Example803B
// see "cylon" in Examples of FastLED
//

#include "FastLED.h"

#define NUM_LEDS 10
#define DATA_PIN 4

CRGB leds[NUM_LEDS];

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

void loop() {
  for(int i = 0; i < NUM_LEDS; i++) {
    leds[i] = CRGB::Blue;
    FastLED.show();
    leds[i] = CRGB::Black;
    delay(30);
  }
  for(int i = NUM_LEDS-1; i >= 0; i--) {
    leds[i] = CRGB::Green;
    FastLED.show();
    leds[i] = CRGB::Black;
    delay(30);
  }
}
```

点灯位置が行ったり来たりする

HSV(HSB) : 色相・彩度・明度

```
// Example 804A simple blink by HSV color setting
// see https://github.com/FastLED/FastLED/wiki/Controlling-leds
// H. Kawakami, October 2015

#include <FastLED.h>

#define NUM_LEDS 1
#define DATA_PIN 4

CRGB leds[NUM_LEDS];

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

void loop() {
  for(int i=0; i<NUM_LEDS; i++){
    leds[i].setHSV(221, 51, 120);
    // leds[i]=CHSV(221, 51, 120);
    // leds[i].setHue(221);
    // fill_solid(&(leds[i]), 1, CHSV(221, 51, 120));
    FastLED.show();
    delay(1000);

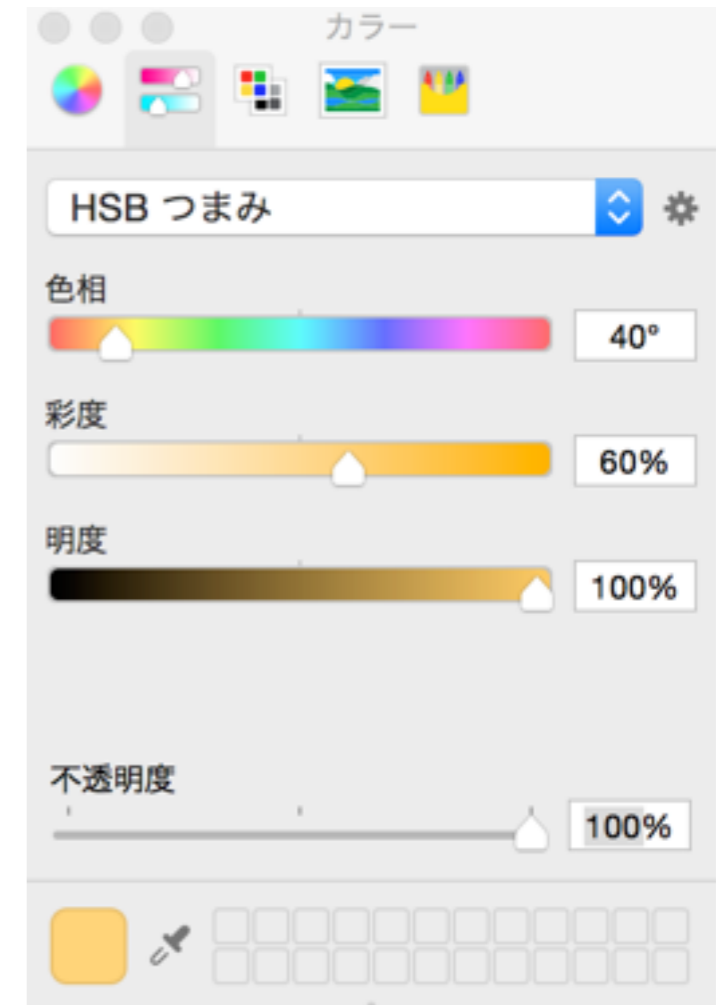
    leds[i].setHSV(0, 0, 0);
    FastLED.show();
    delay(1000);
  }
}
```

FastLED

0 - 255

0 - 255

0 - 255



HSV(HSB) : 色相・彩度・明度

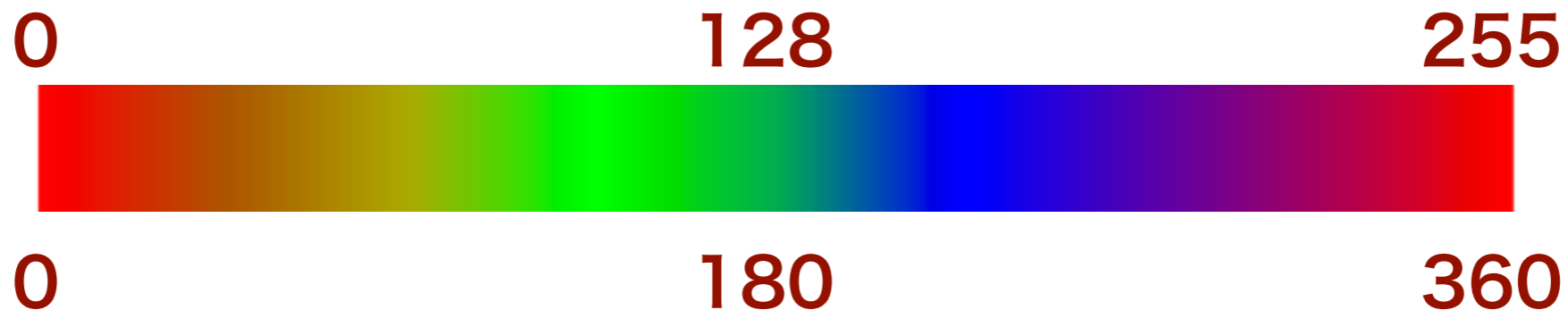
```
// Example804B
// Color setting by Hue
// H. Kawakami, October 2015

#include "FastLED.h"

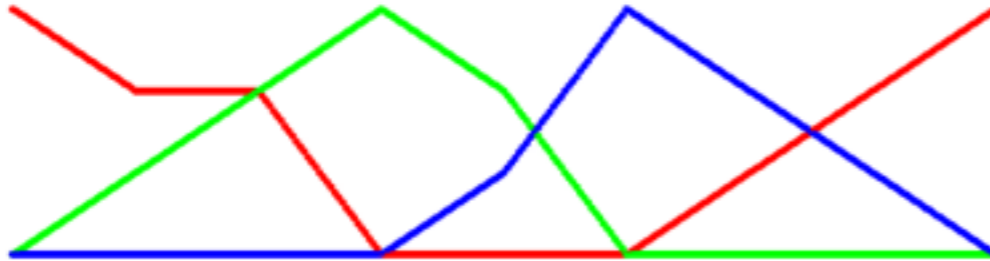
#define NUM_LEDS 10
const int DATA_PIN = 4;
CRGB leds[NUM_LEDS];

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

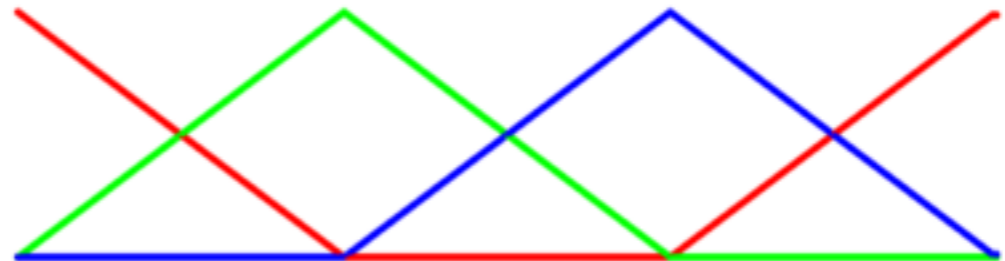
void loop() {
  for (int i=0; i < 256; i++) {
    fill_solid(&(leds[0]), NUM_LEDS, CHSV(i, 150, 150));
    FastLED.show();
    delay(50);
  }
}
```



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[i]);
```

光センサーを使ってLEDを調光する

フォトトランジスタ NJL7502L



C : コレクター

E : エミッター

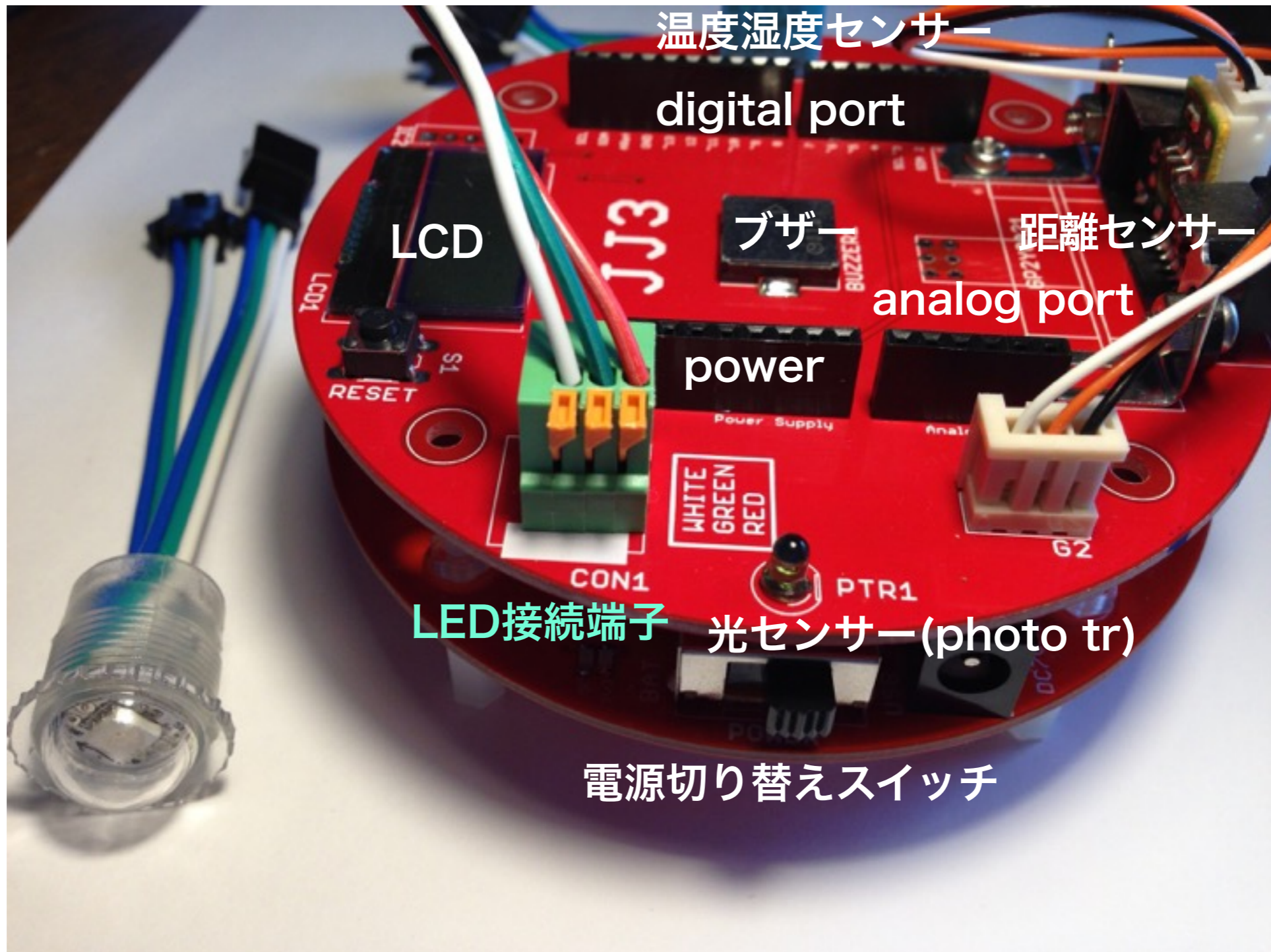
辻先生配布のDVD/資料

JJ3 ポート一覧

ポート	接続先	型番	ライブラリ
D0 (INT2)	人感センサ*1	SB-612A	digitalRead, INT2
D1	—	—	—
D2 (SDA)	液晶ディスプレイ 8x2	AQM0802	ST7032
D3 (SCL)	高精度温湿度センサ*1	AE-HDC1000	
D4	フルカラーLED (WS2811 互換)	WS2811LED	FastLED
D5, D6, D7	—		—
D8	タクトスイッチ	SW1	digitalRead INPUT_PULLUP
D9	温湿度センサ	DHT11	DHT
D10	圧電スピーカー	PKLCS1212	Tone
D11	赤外線受信センサ	GP1UXC4XQS	IRremote
D12	—		—
D13	青色LED		digitalWrite
A0	照度センサ	NJL7502L	analogRead
A1, A2	—		—
A3	赤外線距離センサ	GP2Y0A21YK	analogRead
A4, A5	—		—

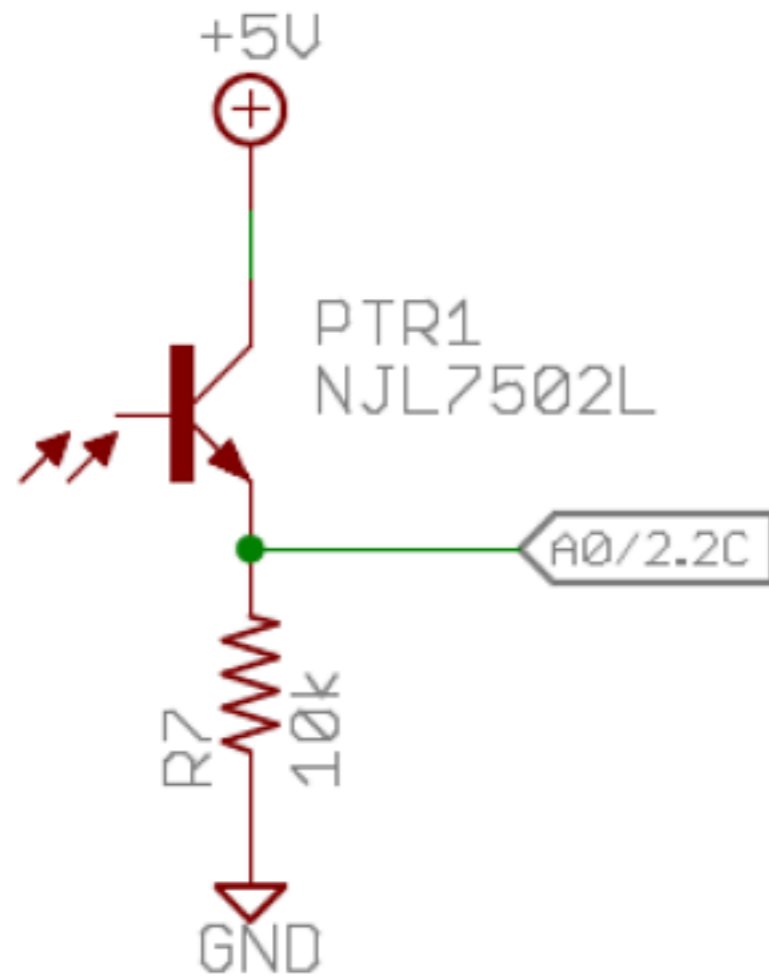
*1 オプション

JJ3 シールド

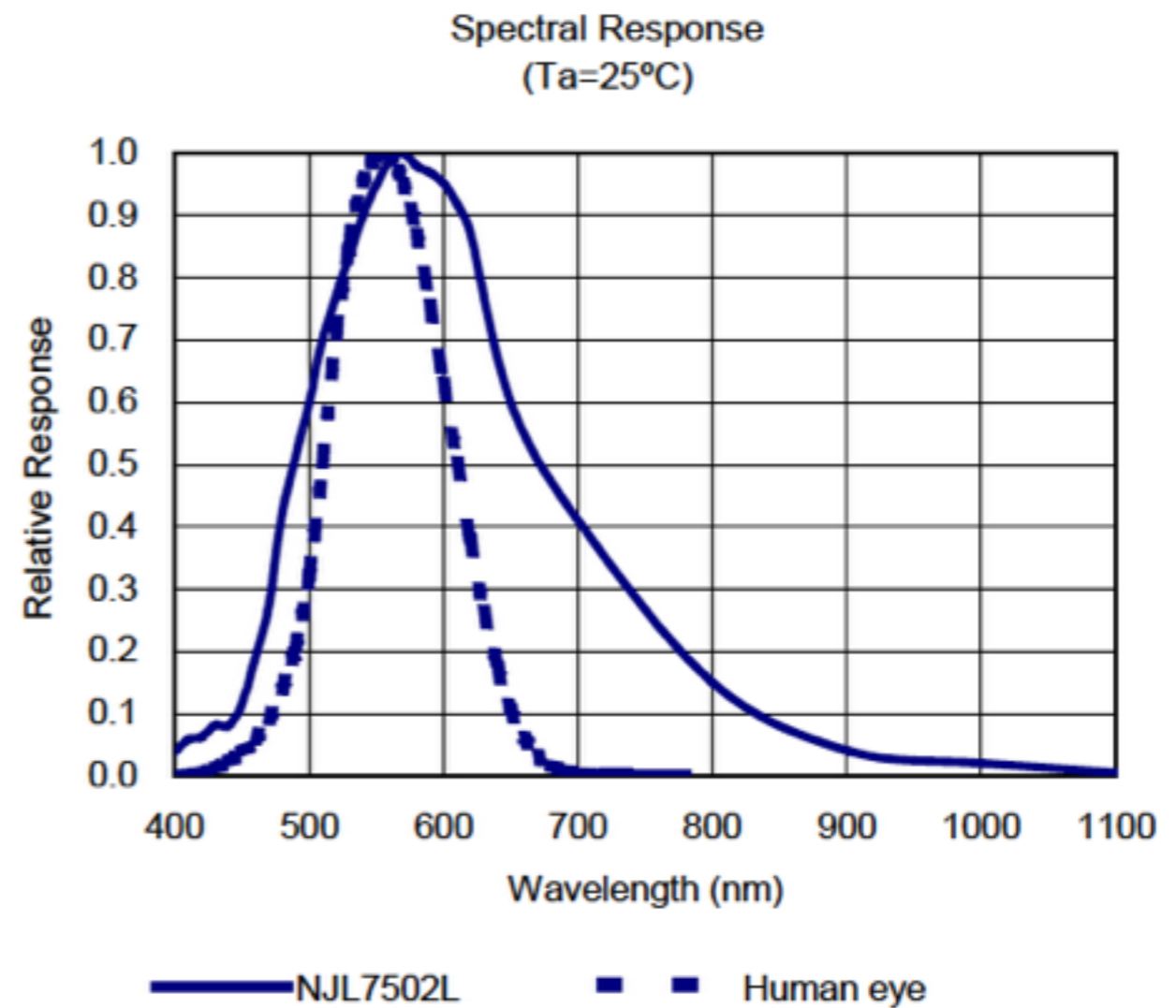


配線図とNJL7502Lの特性

辻先生配布のDVD/回路図



NJL7502Lの特性例

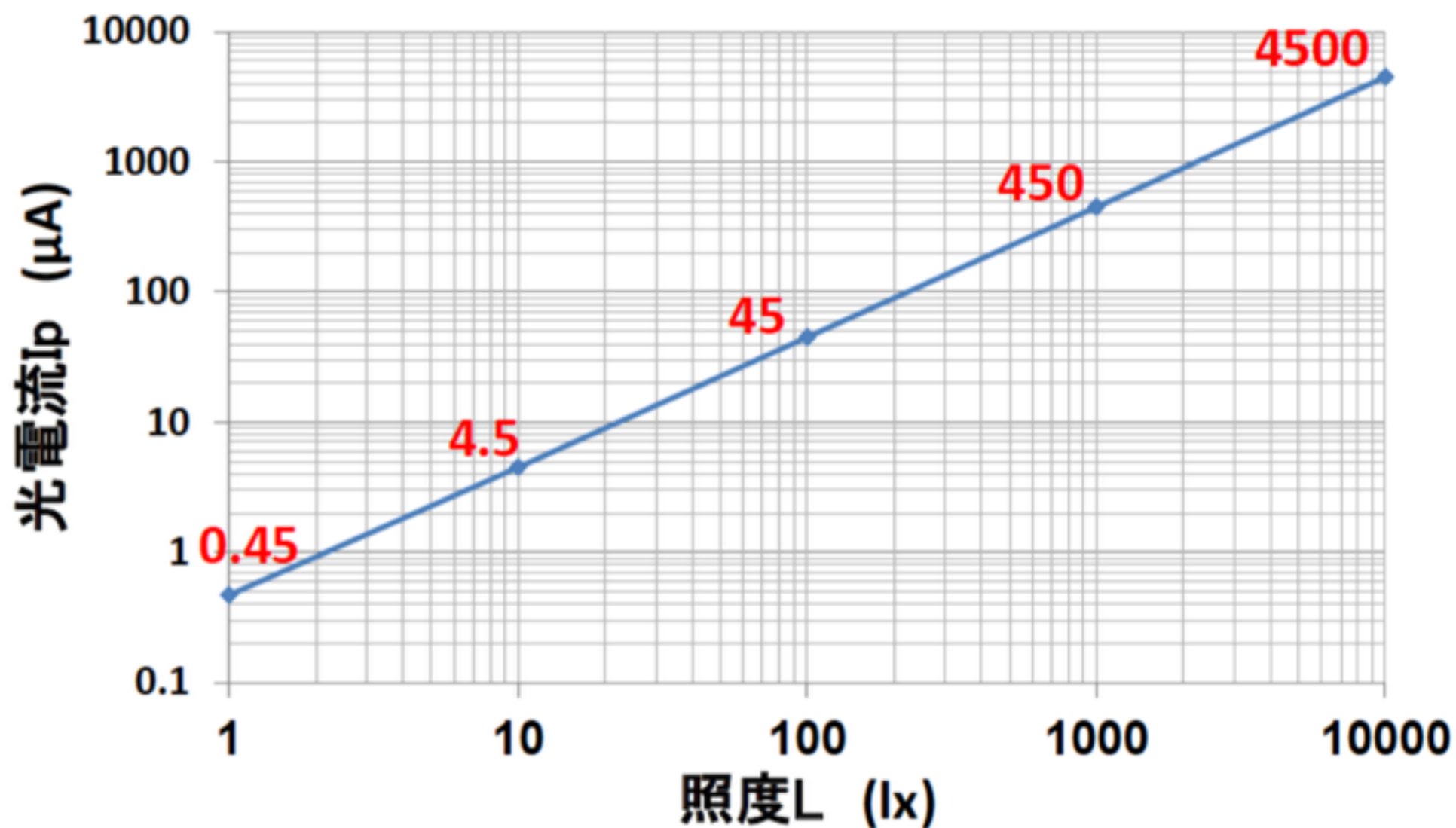


結果：A0の電圧が明るさに従って 0 - 1024 の数値で変化する

照度と光電流の関係

辻先生第6回のスライド p.24

$$\text{照度 } L = 2.22 \times I_p \text{ [lx]} = 222 \times V_p \text{ [lx]}$$



Example 804C

```
// Example804C: HSVTestWithSensor
// Led color test for HSV(Hue, Saturation, Value)
// TapeLED:
// red = 5V
// white = GND
// green = 4 : DATA_PIN
// Sensor = A0: Photo transistor
// H. Kawakami, October, 2015

#include "FastLED.h"

#define NUM_LEDS 10
#define DATA_PIN 4
#define SENSOR_PIN 0 ←

CRGB leds[NUM_LEDS];
int sensor=500; ←

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

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

$$y = \text{map}(x, 0, 1024, 0, 255)$$

