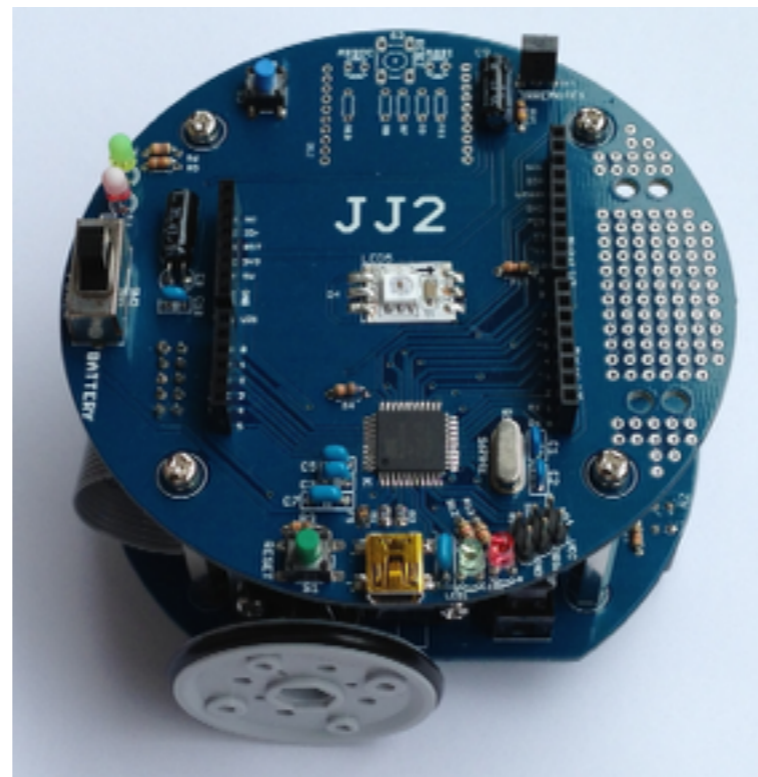


# ロボットをつくらう

## 第4回 センサーとLEDの調光



<http://cms.db.tokushima-u.ac.jp/DAV/person/S10723/>

川上 博

2014/08/02

# 今日のテーマ

---

ブレッドボード上のLEDの調光

今日の電子工作：暗くなったらLEDが点灯する

JJ2の上のLED5を点灯してみよう

JJ2の上のBuzzer1を鳴らしてみよう

# スケッチ（プログラム）：前回の復習

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(10);
  digitalWrite(led, LOW);
  delay(10);
}
```

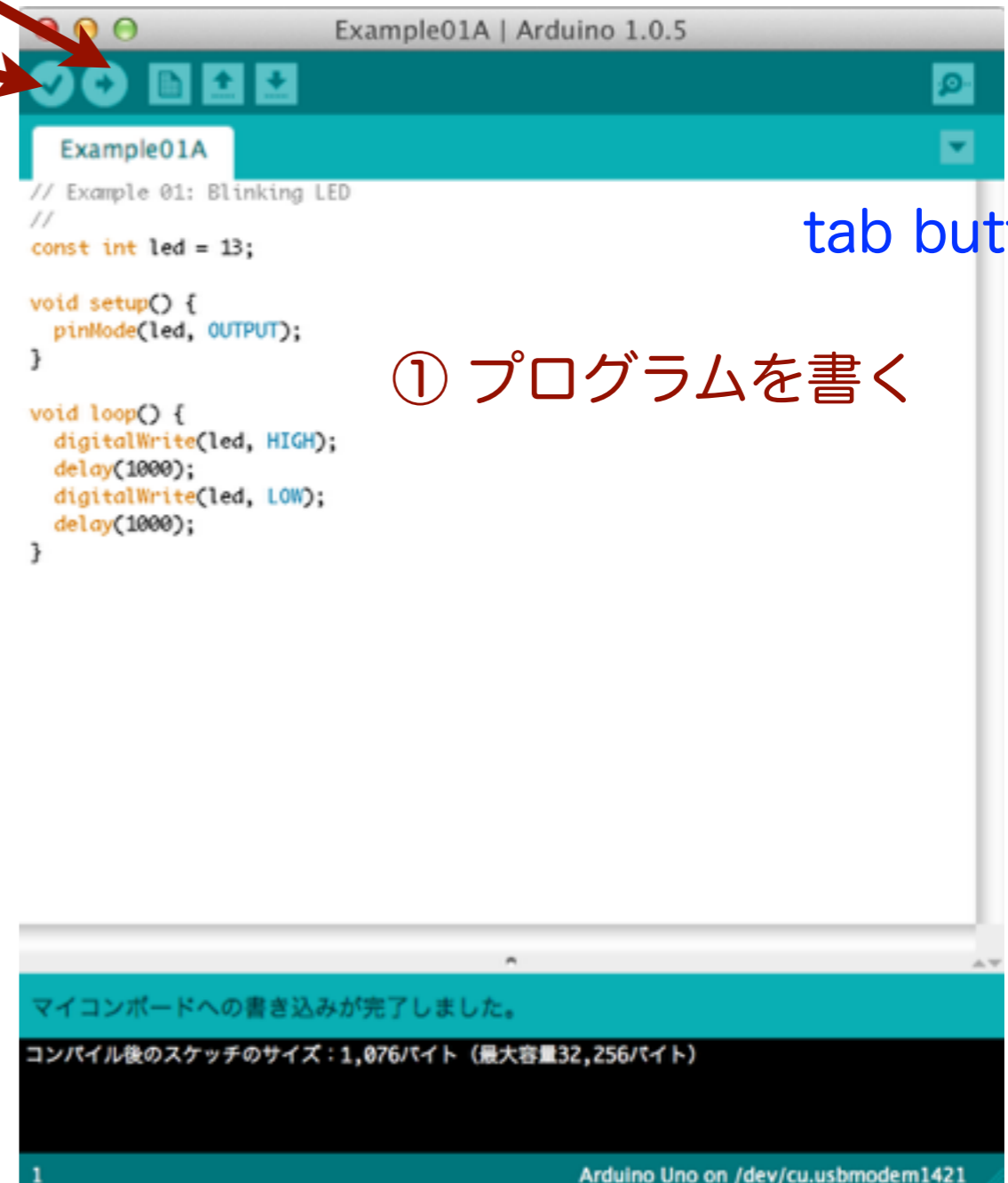
② Verify

③ Upload

ファイル操作

New Open Save

Serial monitor



tab button

① プログラムを書く

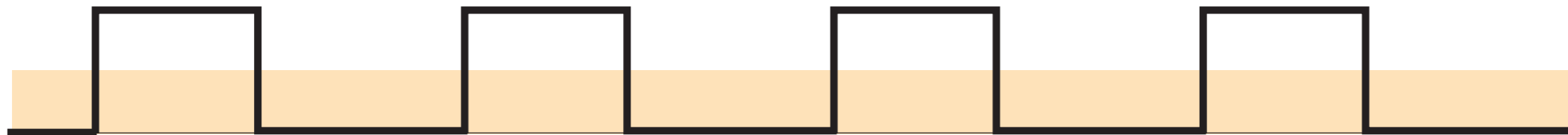
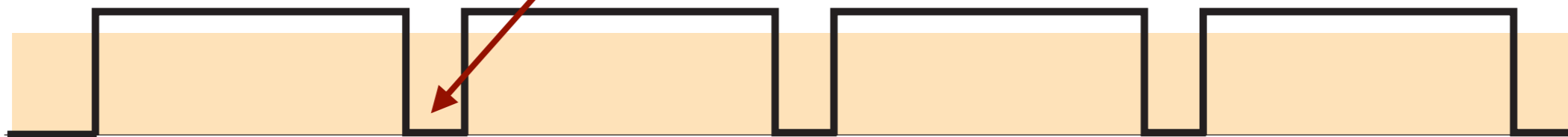
④ 加えて20となる数字

# LEDの明るさが変わっているようにみえる

```
digitalWrite(13, HIGH);
```

```
digitalWrite(13, LOW);
```

明るくみえる



暗くみえる



周期(20 ms)



時刻

# ブレッドボード上のLEDの調光

# Read と Write

```
digitalRead(10);
```

```
digitalWrite(12, HIGH);
```



## Read

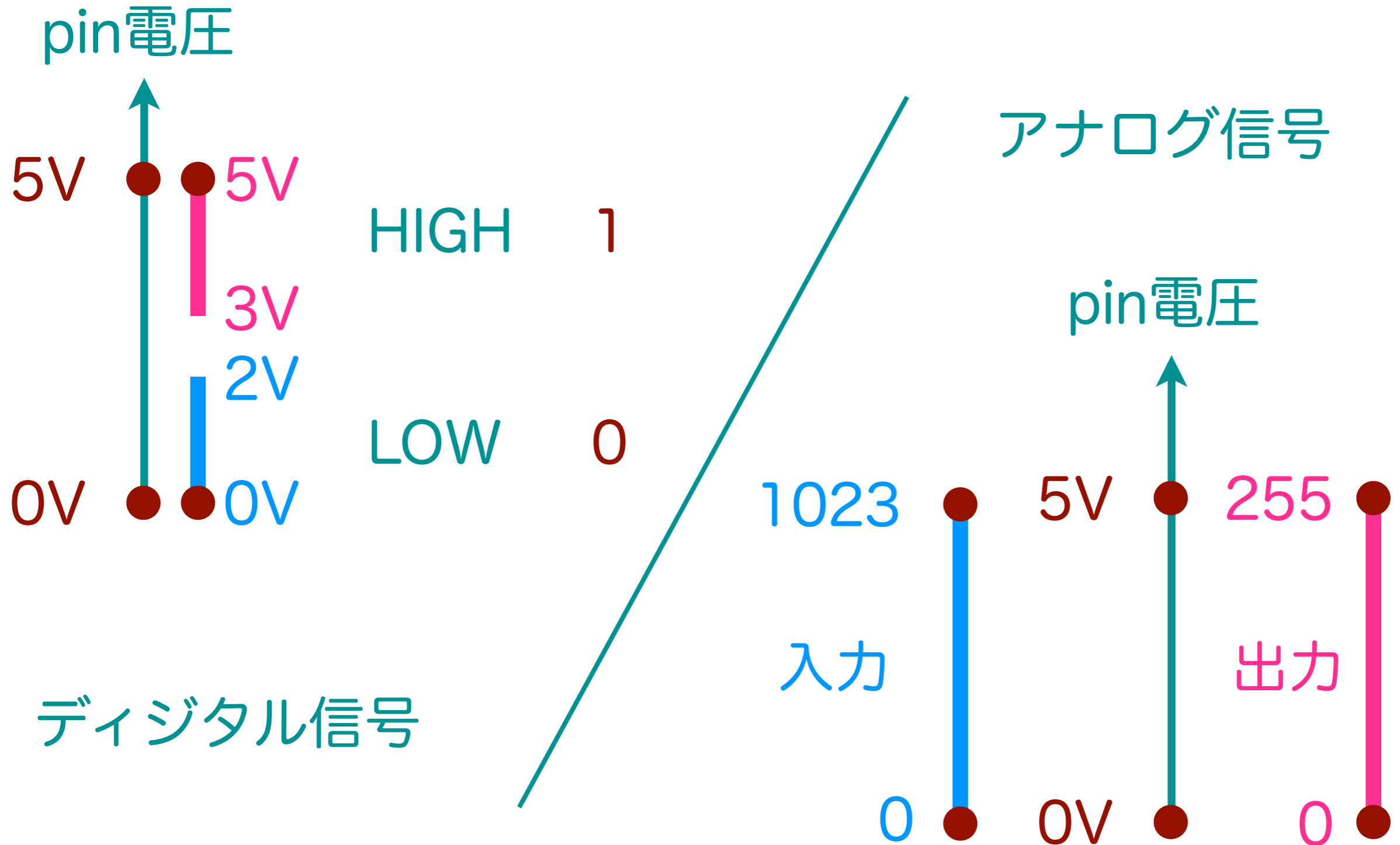
## Write

```
analogRead(5);
```

```
analogWrite(6, 255);
```

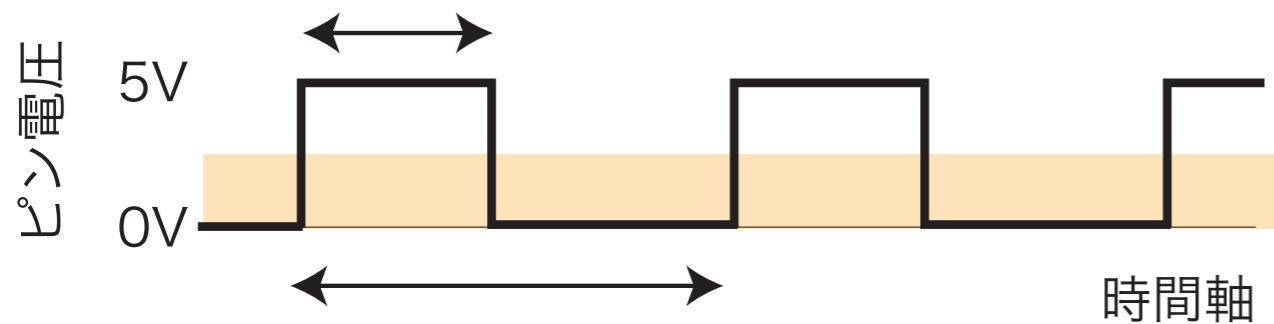


# Digital信号 と Analog信号



# analogWrite

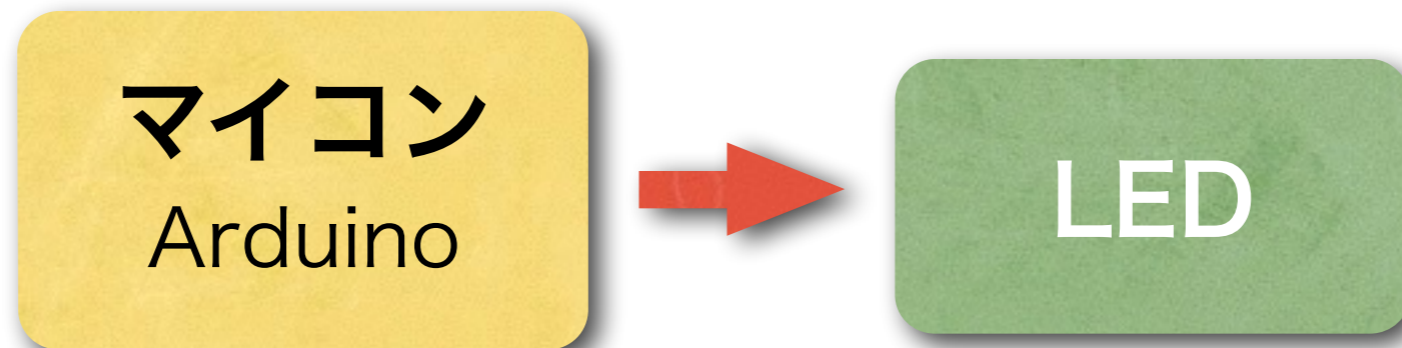
パルスの幅を調節する：PWM(pulse width modulation)



整数：0～255

ピン番号

```
analogWrite(6, 100);
```

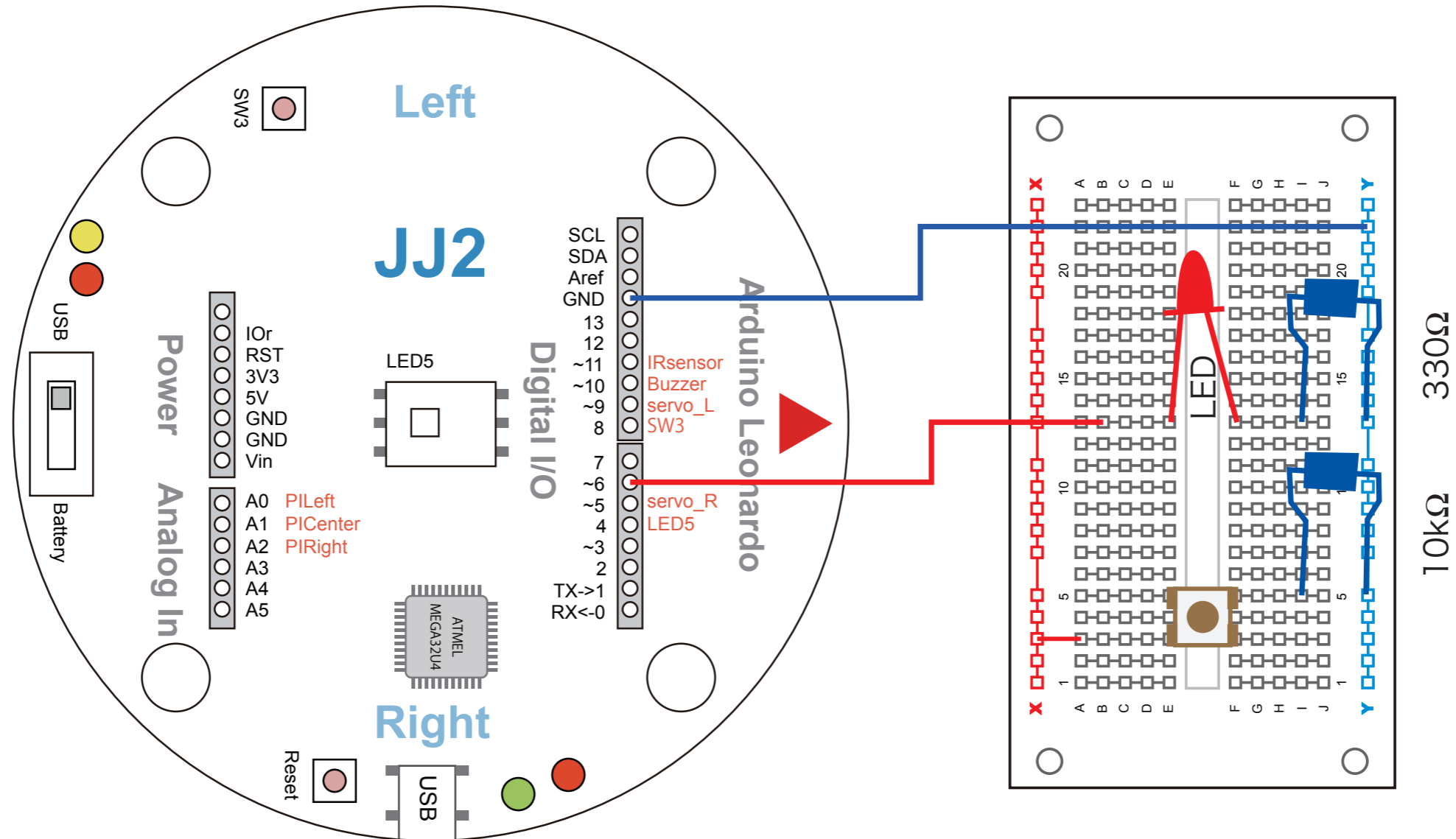


アナログ出力

ブレッドボード上のLEDを調光する

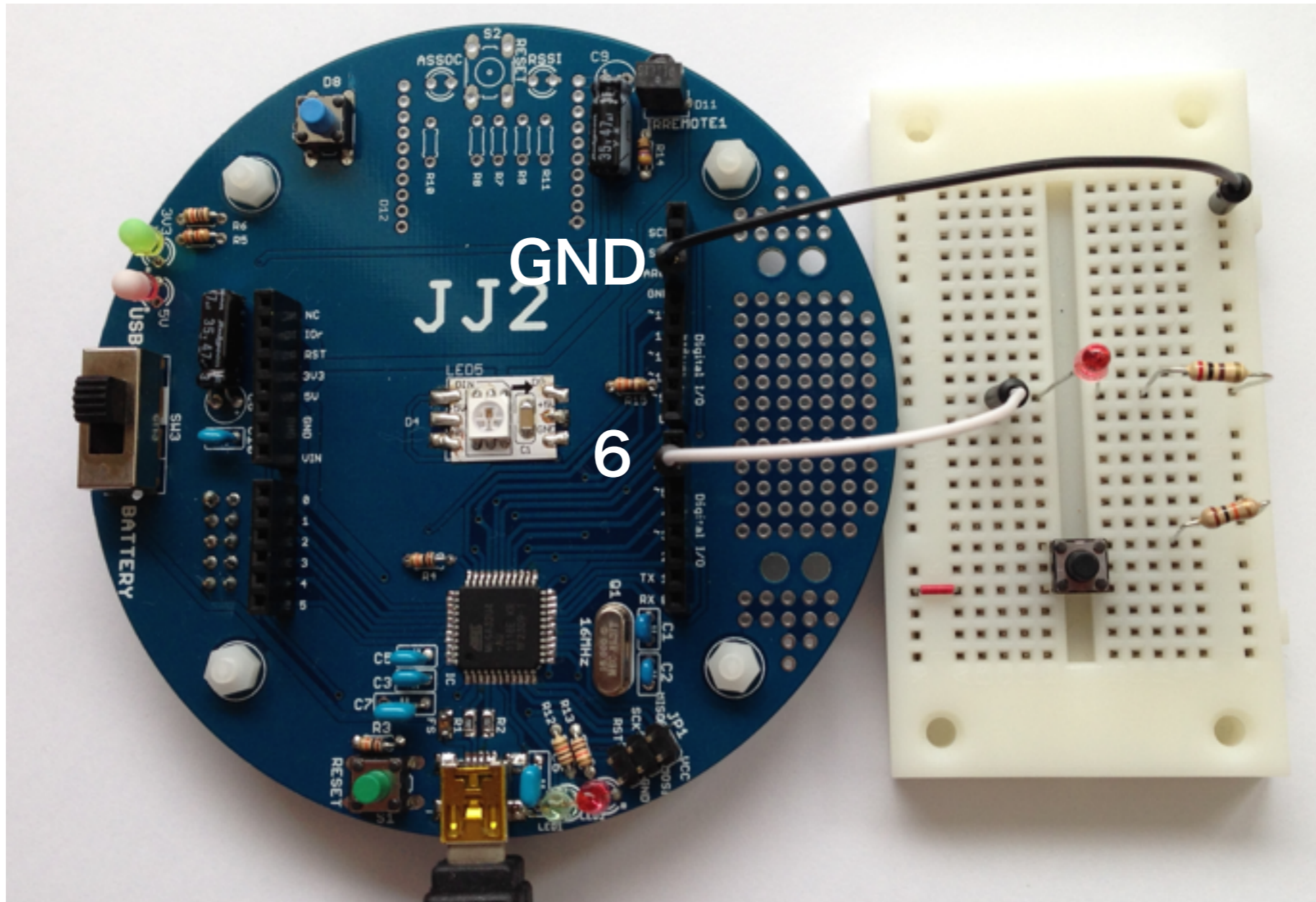


# LEDの調光：配線図



~の付いたピン：analogWriteの使用可能

# LEDの調光：配線図



```
/*  
Example 401A  
Fade in and out: analogWrite  
*/
```

```
const int led = 6;
```

```
void setup() {  
  pinMode(led, OUTPUT);  
}
```

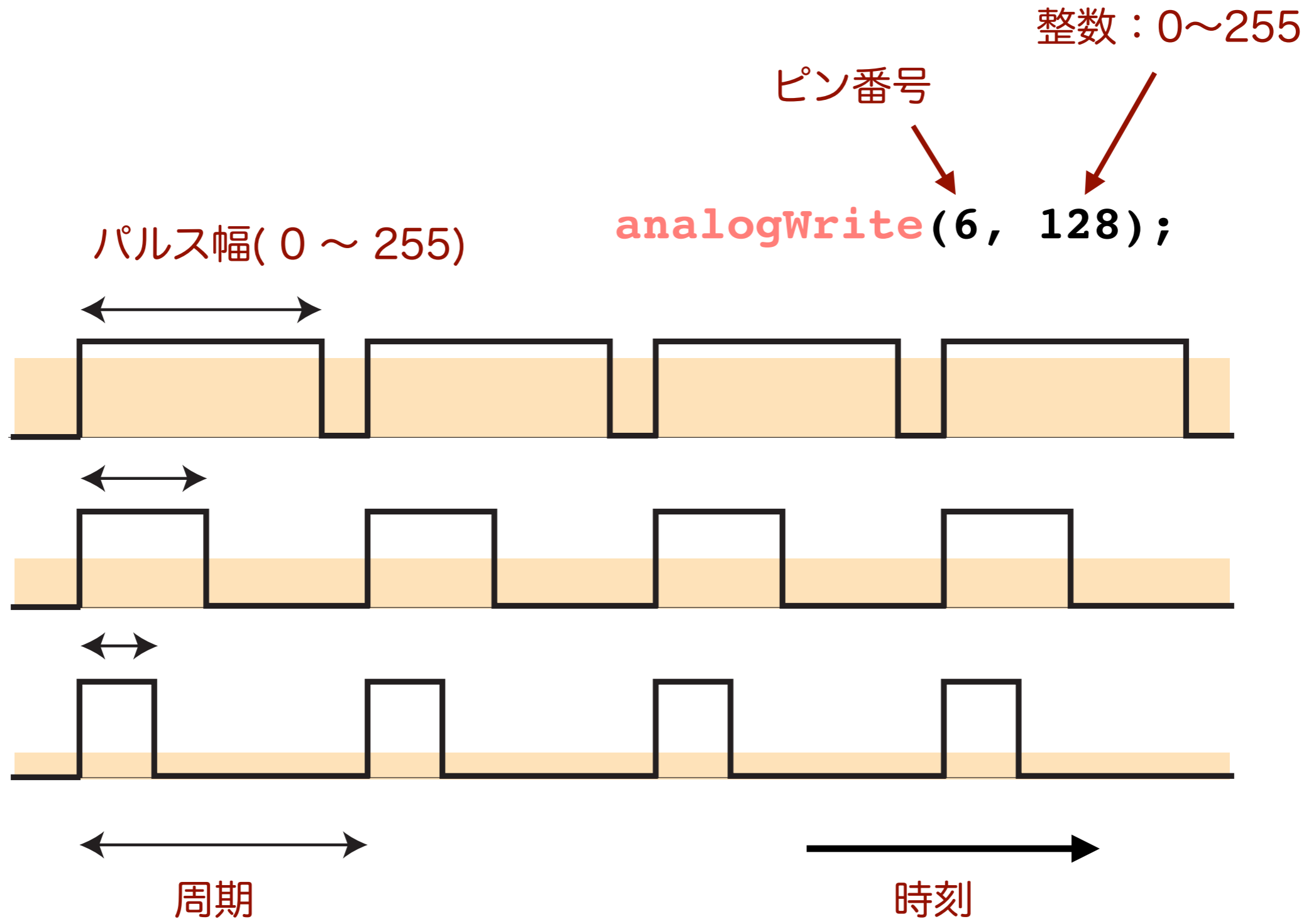
```
void loop() {  
  for(int i=0; i<255; i+=5){  
    analogWrite(led, i);  
    delay(50);  
  }  
  for(int i=255; i>0; i-=5){  
    analogWrite(led, i);  
    delay(50);  
  }  
}
```

(繰り返し文)

```
for(i=0; i<255; i+=5){  
  実行したい計算  
}
```



# PWM : pulse width modulation



# 繰り返し計算：for文とwhile文



## 繰返文：確定的

```
sum=0;
for(i=0; i<101; i++){
    sum+=i;
}
```



## 繰返文：不確定的

```
sum=0;
i=0;
while(i<101){
    sum+=i;
    i++;
}
```



## 代入文

```
sum = sum+i;

sum += i;
```



## 1ふやす

```
i = i+1;

i += 1;

i++;

++i;
```

## 今日の電子工作

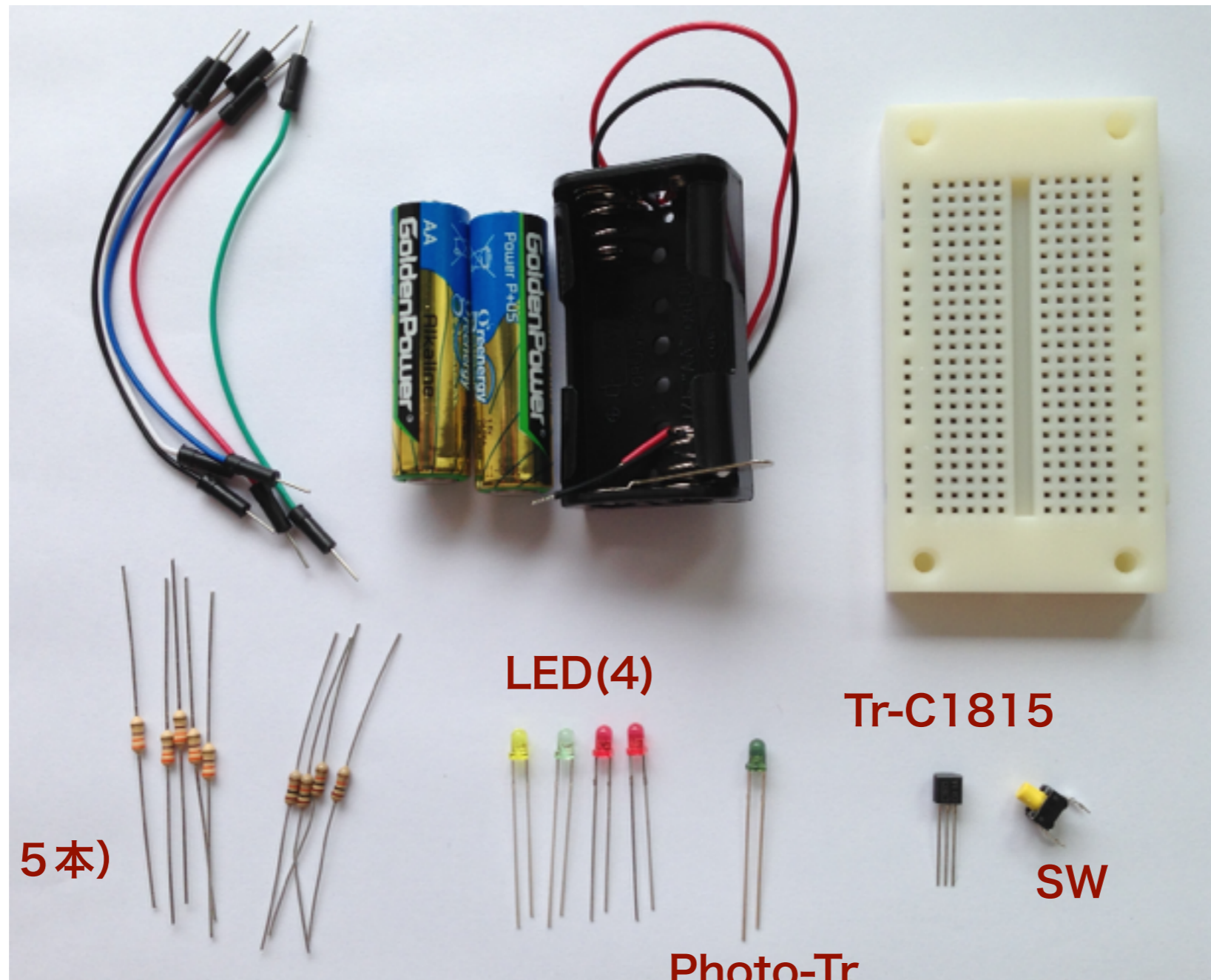
暗くなったらLEDが点灯する

## 配布した電子部品

ジャンパー線 (5本)

電池とホルダー

ブレッド・ボード



抵抗 (330 5本)

LED(4)

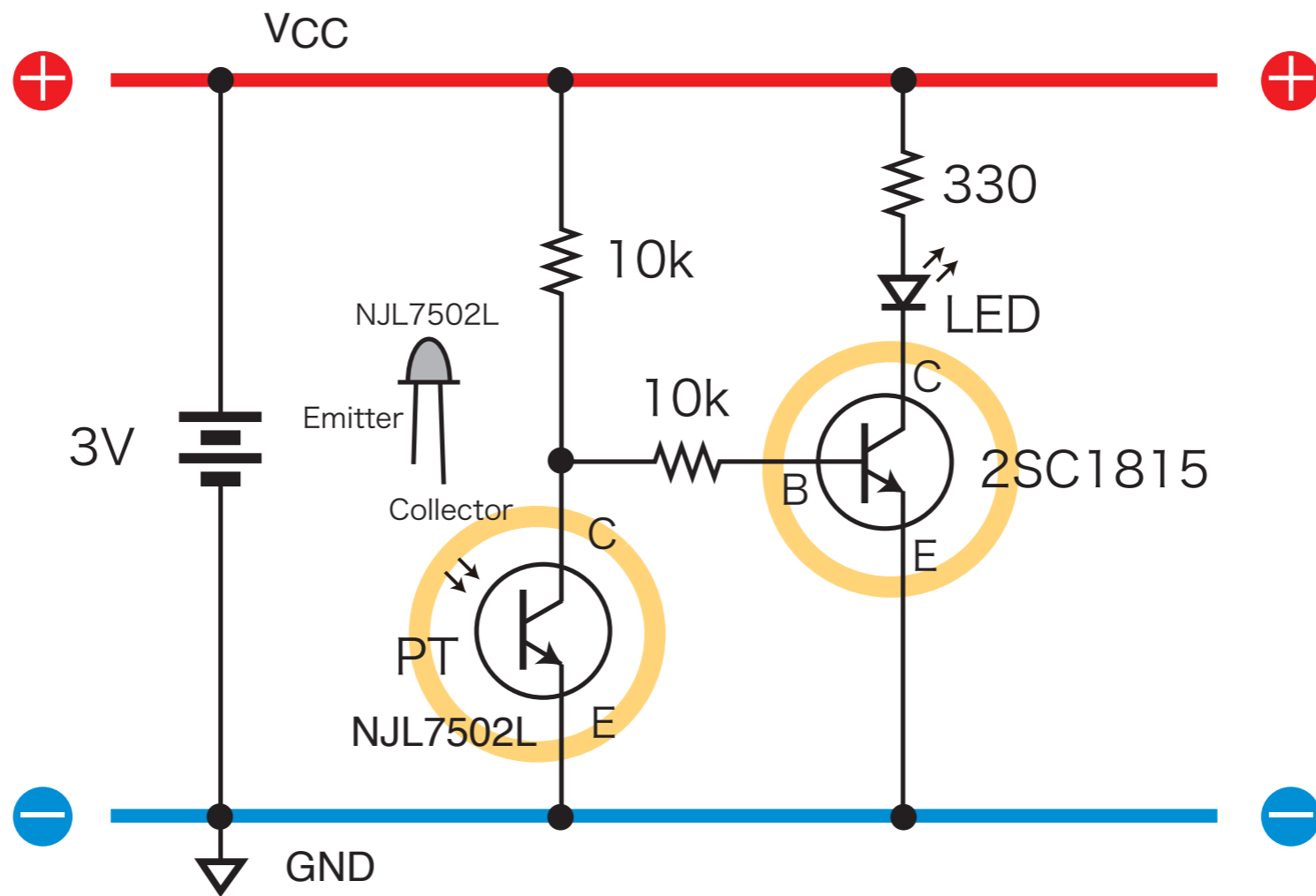
Tr-C1815

抵抗 (10k 4本)

Photo-Tr  
NJL7502L

SW

# 暗くなったら点灯する





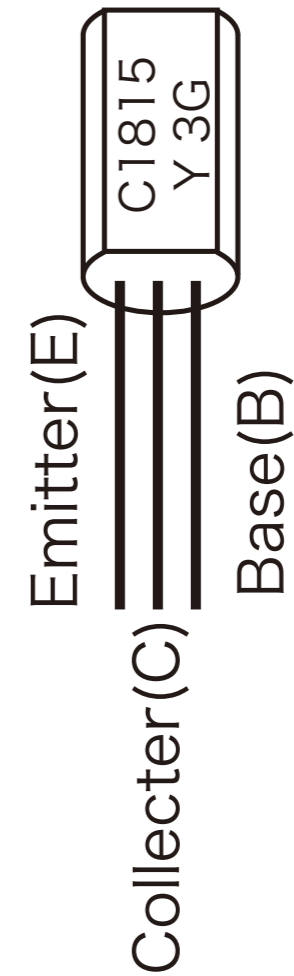
## 2つのトランジスタ

フォト・トランジスタ NJL7502L

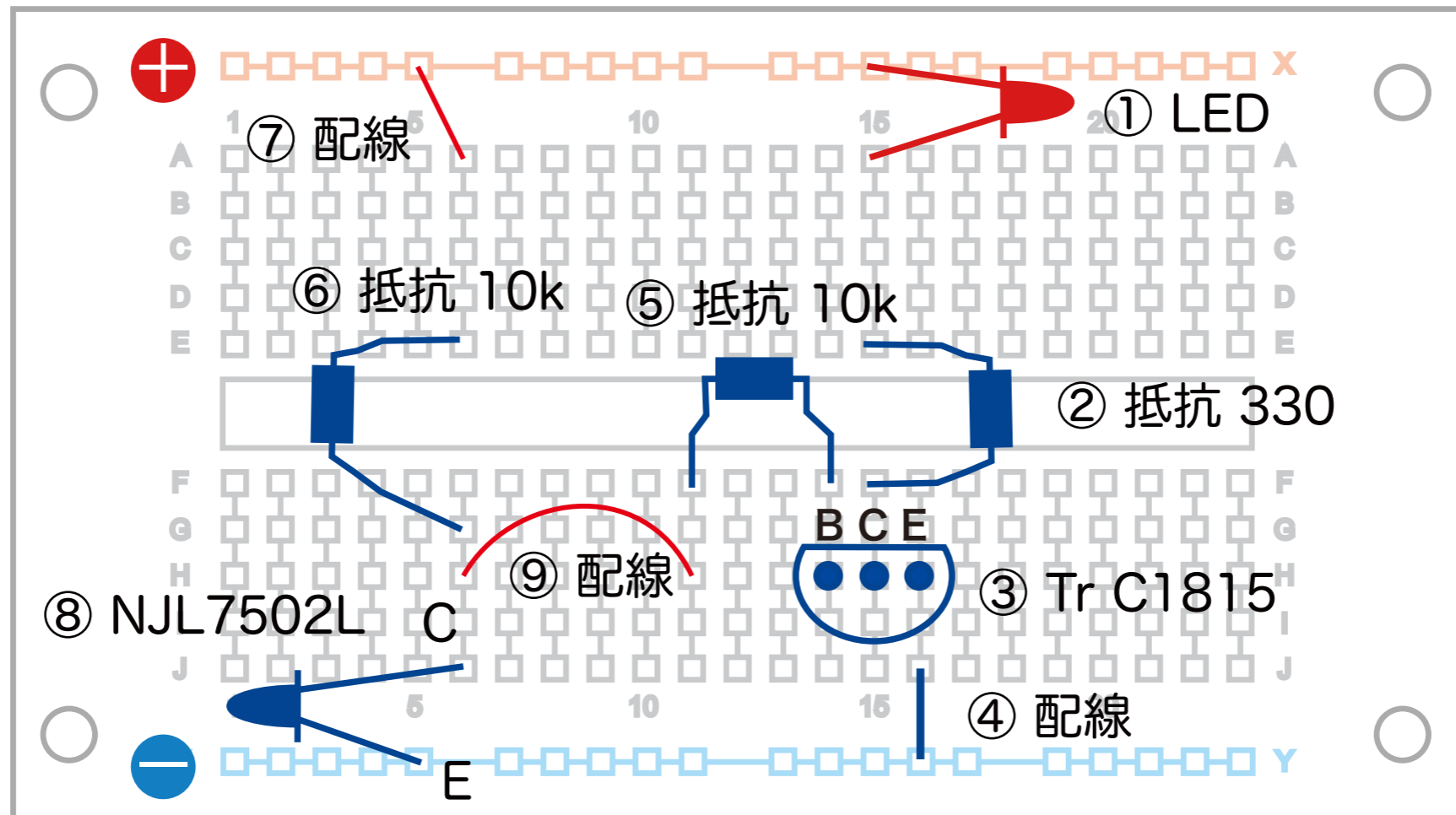
2SC1815

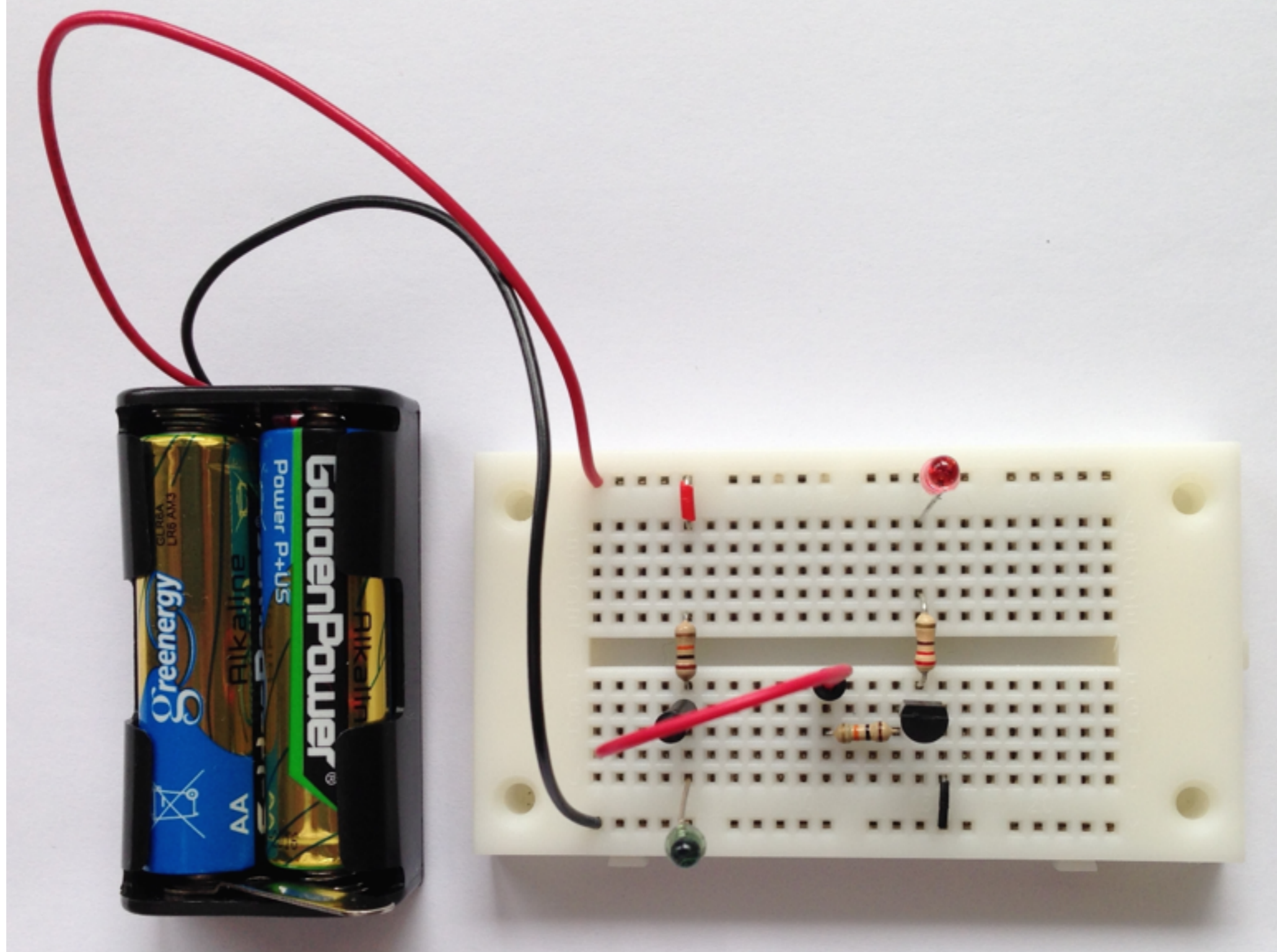
C : コレクター

E : エミッター



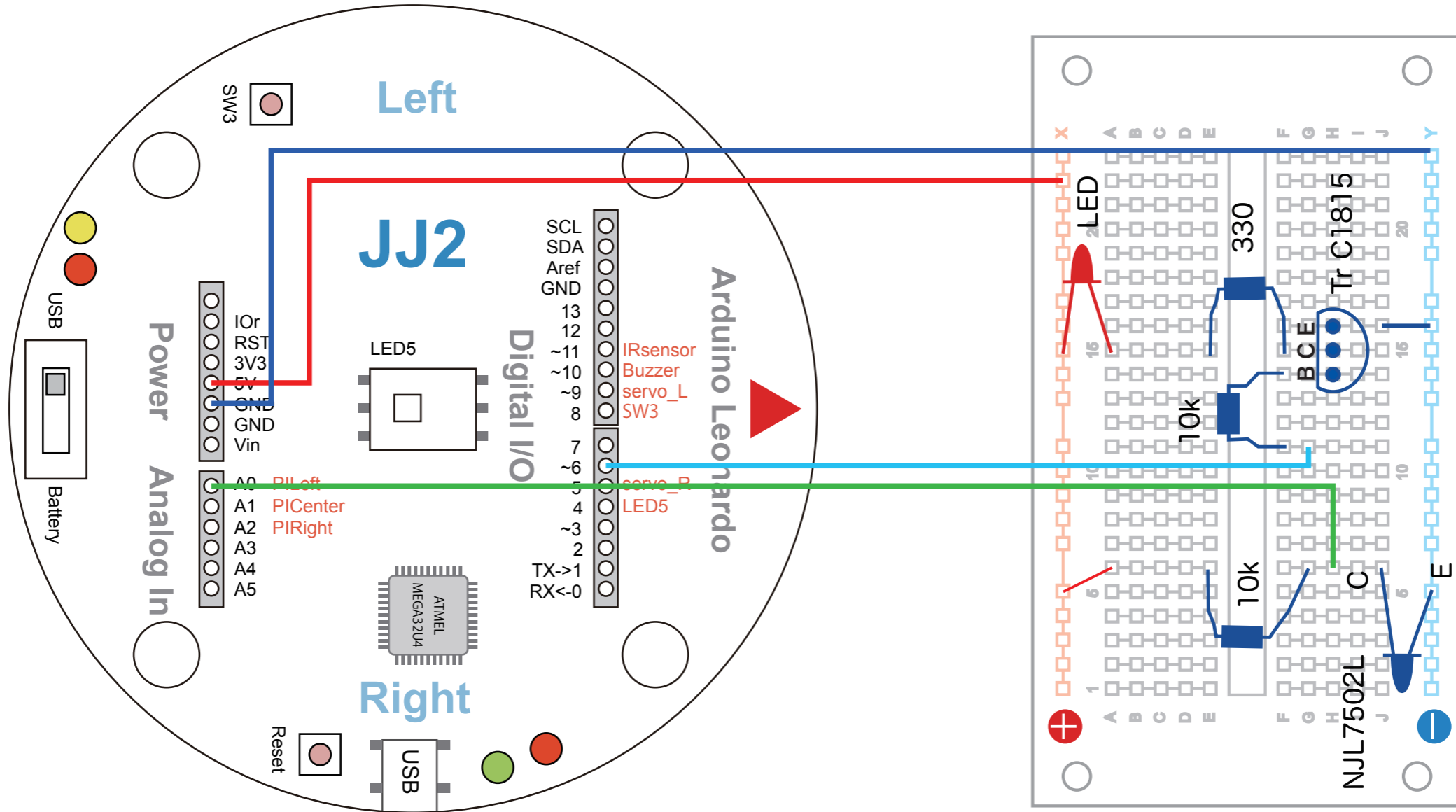
# 立体配線図

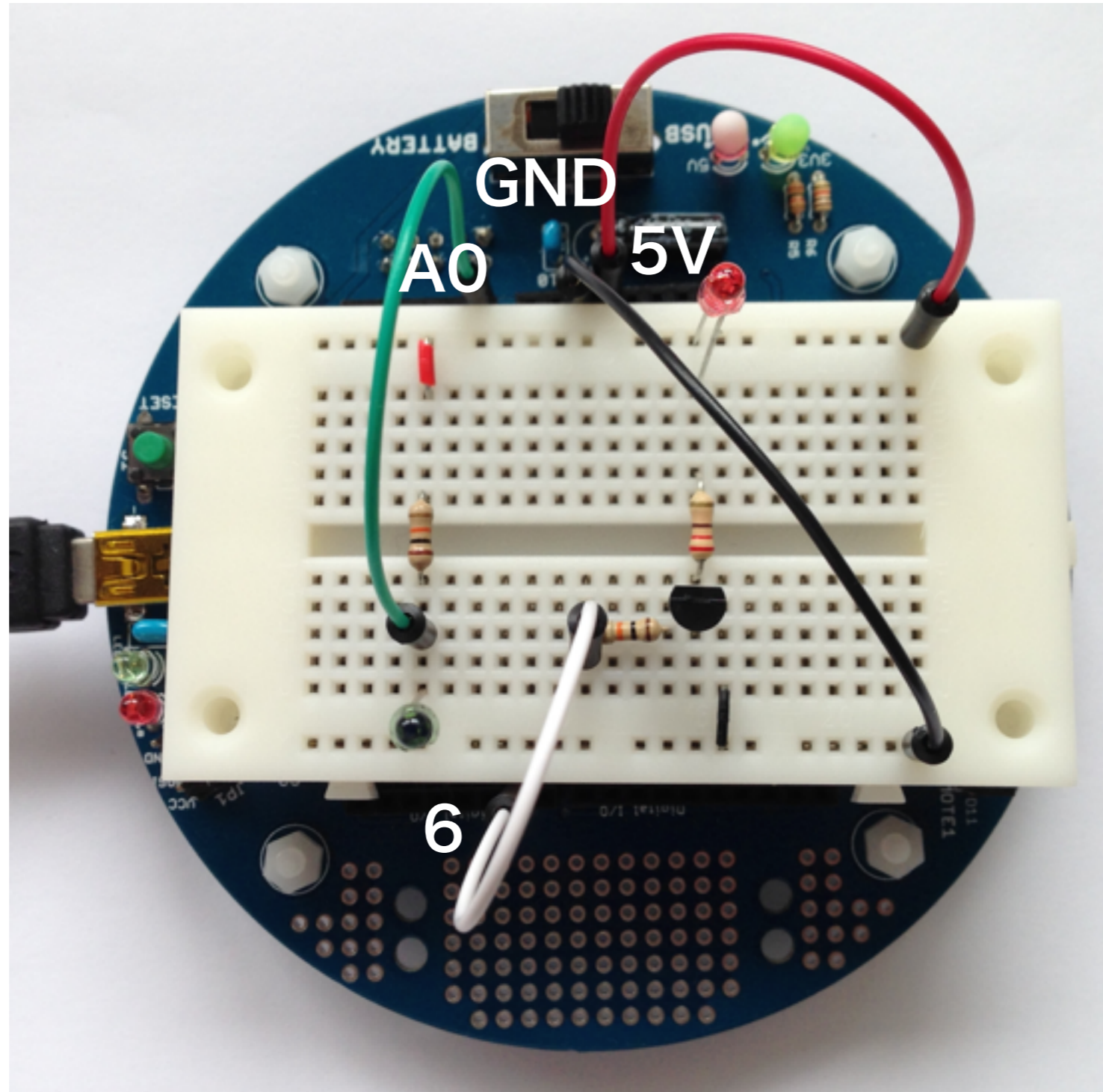




センサーを使ってLEDを点けたり消したりする

# 光センサー入力, アナログ出力 : 配線図





# Example 402A

---

```
/* Example 402A
   analog read and write
*/
const int led = 6;
const int photoTr=A0;
int val;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  val=analogRead(photoTr)/4;
  analogWrite(led, val);
  delay(50);
}
```

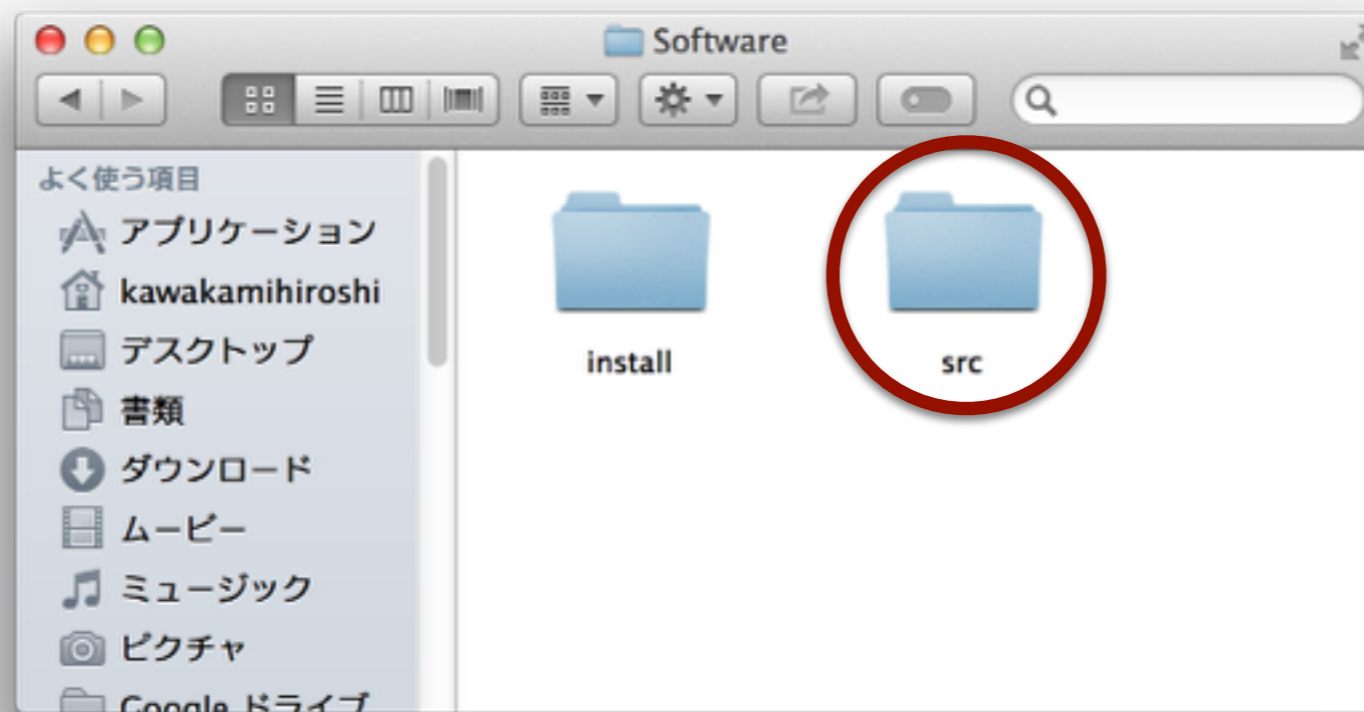
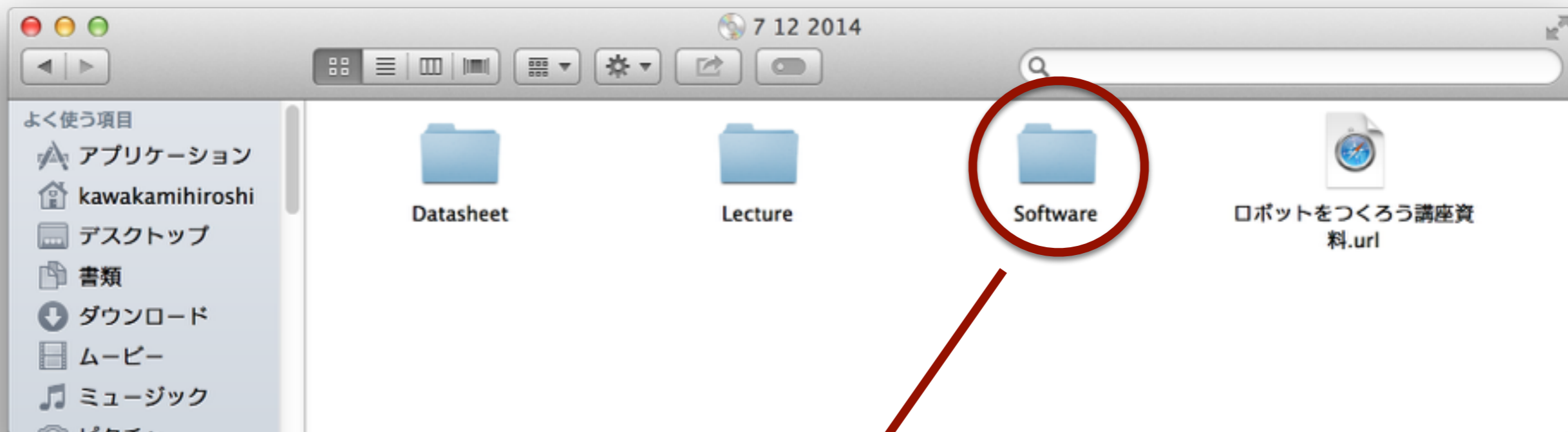
光センサーからの入力

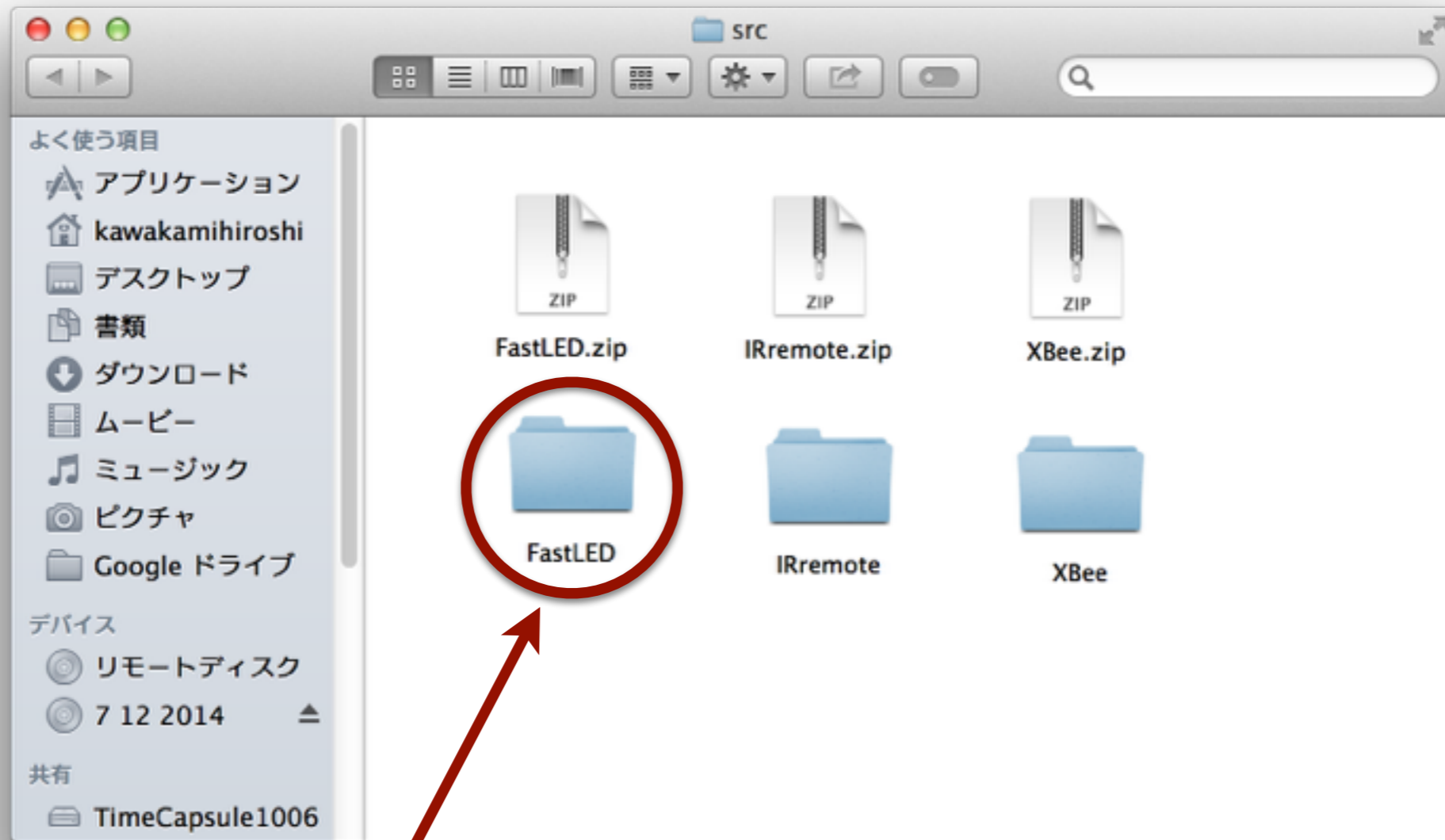






**JJ2 ボード上のLED5を点灯してみよう**

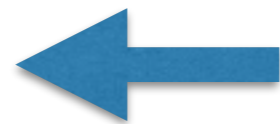




「Arduino」フォルダの中の「libraries」フォルダに入れる

```
/*  
Example 403A  
*/
```

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



LED5を使うためのライブラリー

```
const int ledPin=4;  
CRGB leds[1];
```

LED5はピン番号4のピンにつながっている

```
void setup(){
```

```
  delay(2000);
```

```
  LEDES.setBrightness(255);
```

```
  LEDES.addLeds<WS2811, ledPin>(leds, 1);
```

```
}
```

この数値を変えて明るさを調整する

```
void loop(){
```

```
  for(int i=0; i<255; i++){
```

```
    fill_solid(&(leds[0]), 1, CRGB(0, 0, i));
```

```
    LEDES.show();
```

```
    delay(30);
```

```
  }
```

```
  for(int i=255; i>0; i--){
```

```
    fill_solid(&(leds[0]), 1, CRGB(0, 0, i));
```

```
    LEDES.show();
```

```
    delay(30);
```

```
  }
```

```
}
```

この部分を変えてプログラムする

## Example 404A

---

```
void loop() {
  float r, g, b, val;

  for (val=0; val<1024; val++) {
    r = (cos(6.28 / 256 * val) + 1) * 128;
    g = (cos(6.28 / 256 * (val + 85)) + 1) * 128;
    b = (cos(6.28 / 256 * (val - 85)) + 1) * 128;
    leds[0].r = r;
    leds[0].g = g;
    leds[0].b = b;
    fill_solid(&(leds[0]), 1, CRGB(r, g, b));
    delay(100);
  }
}
```

**ブザー 1 を鳴らす**

## Example 405A

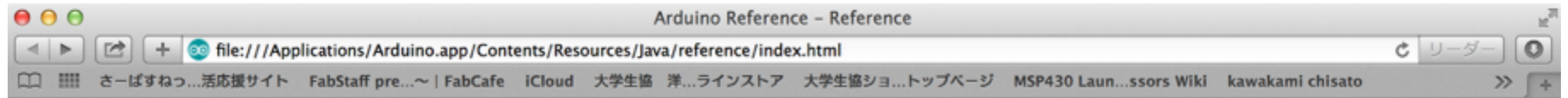
---

`#include "pitches.h"` ← ヘッダーファイルをつくる

```
const int buzzerPin=10;
int melody[] = {NOTE_C4, NOTE_G3,NOTE_G3, NOTE_A3,
                NOTE_G3,0, NOTE_B3, NOTE_C4};
int noteDurations[] = {4,8,8,4,4,4,4,4};

void setup() {
  for (int i = 0; i < 8; i++) {
    int noteDuration = 1000/noteDurations[i];
    tone(buzzerPin, melody[i],noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(10);
  }
}

void loop() {
  // no need to repeat the melody.
}
```



[Buy](#)
[Download](#)
[Getting Started](#)
[Learning](#)
[Reference](#)
[Hardware](#)
[FAQ](#)

[Reference](#)
[Language](#) | 
 [Libraries](#) | 
 [Comparison](#) | 
 [Changes](#)

## Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

### Structure

- [setup\(\)](#)
- [loop\(\)](#)

#### Control Structures

- [if](#)
- [if...else](#)
- [for](#)
- [switch case](#)
- [while](#)
- [do... while](#)
- [break](#)
- [continue](#)
- [return](#)

### Variables

#### Constants

- [HIGH](#) | [LOW](#)
- [INPUT](#) | [OUTPUT](#) | [INPUT\\_PULLUP](#)
- [true](#) | [false](#)
- [integer constants](#)
- [floating point constants](#)

#### Data Types

- [void](#)
- [boolean](#)
- [char](#)
- [float](#)
- [int](#)
- [long](#)
- [String](#)

### Functions

#### Digital I/O

- [pinMode\(\)](#)
- [digitalWrite\(\)](#)
- [digitalRead\(\)](#)

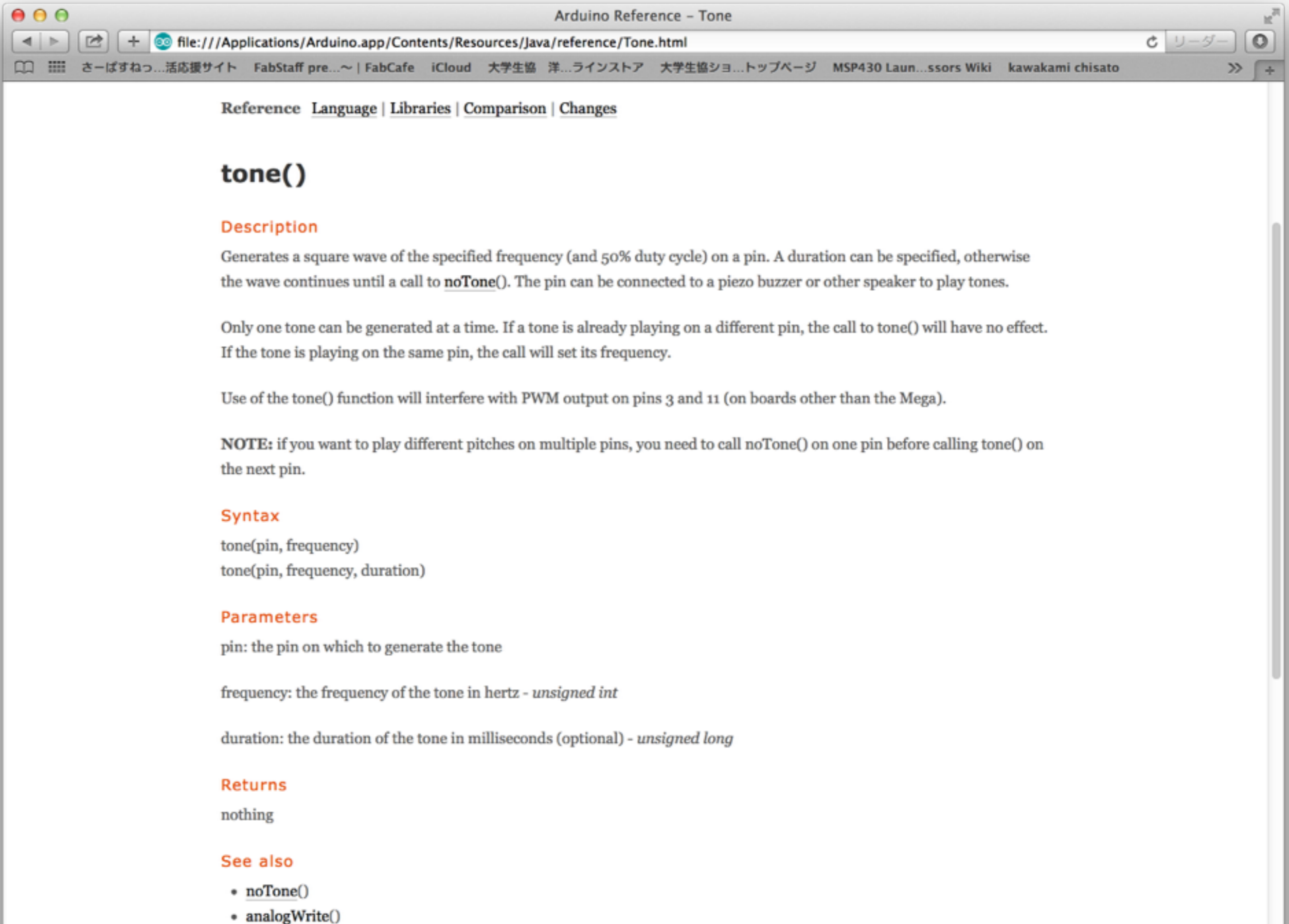
#### Analog I/O

- [analogReference\(\)](#)
- [analogRead\(\)](#)
- [analogWrite\(\)](#) - *PWM*

#### Advanced I/O

- [tone\(\)](#)
- [noTone\(\)](#)





Arduino Reference - Tone

file:///Applications/Arduino.app/Contents/Resources/Java/reference/Tone.html

さーばすねっ...活応援サイト FabStaff pre...~ | FabCafe iCloud 大学生協 洋...ラインストア 大学生協シヨ...トップページ MSP430 Laun...ssors Wiki kawakami chisato

[Reference](#) | [Language](#) | [Libraries](#) | [Comparison](#) | [Changes](#)

## tone()

### Description

Generates a square wave of the specified frequency (and 50% duty cycle) on a pin. A duration can be specified, otherwise the wave continues until a call to `noTone()`. The pin can be connected to a piezo buzzer or other speaker to play tones.

Only one tone can be generated at a time. If a tone is already playing on a different pin, the call to `tone()` will have no effect. If the tone is playing on the same pin, the call will set its frequency.

Use of the `tone()` function will interfere with PWM output on pins 3 and 11 (on boards other than the Mega).

**NOTE:** if you want to play different pitches on multiple pins, you need to call `noTone()` on one pin before calling `tone()` on the next pin.

### Syntax

```
tone(pin, frequency)
tone(pin, frequency, duration)
```

### Parameters

pin: the pin on which to generate the tone

frequency: the frequency of the tone in hertz - *unsigned int*

duration: the duration of the tone in milliseconds (optional) - *unsigned long*

### Returns

nothing

### See also

- `noTone()`
- `analogWrite()`

# Example 405A

```

Example405A | Arduino 1.0.5
Example405A pitches.h
/*
Melody : Example405A
Plays a melody
circuit:
 * 8-ohm speaker on digital pin 10

created 21 Jan 2010
modified 30 Aug 2011
by Tom Igoe

This example code is in the public domain.
http://arduino.cc/en/Tutorial/Tone
*/
#include "pitches.h"

const int buzzerPin=10;
int melody[] = {NOTE_C4, NOTE_G3,NOTE_G3, NOTE_A3, NOTE_G3,0};
int noteDurations[] = {4,8,8,4,4,4,4,4};

```

マイコンボードへの書き込みが完了しました。  
コンパイル後のスケッチのサイズ：7,336バイト（最大容量28,672バイト）

8 Arduino Leonardo on /dev/cu.usbmodem1421

```

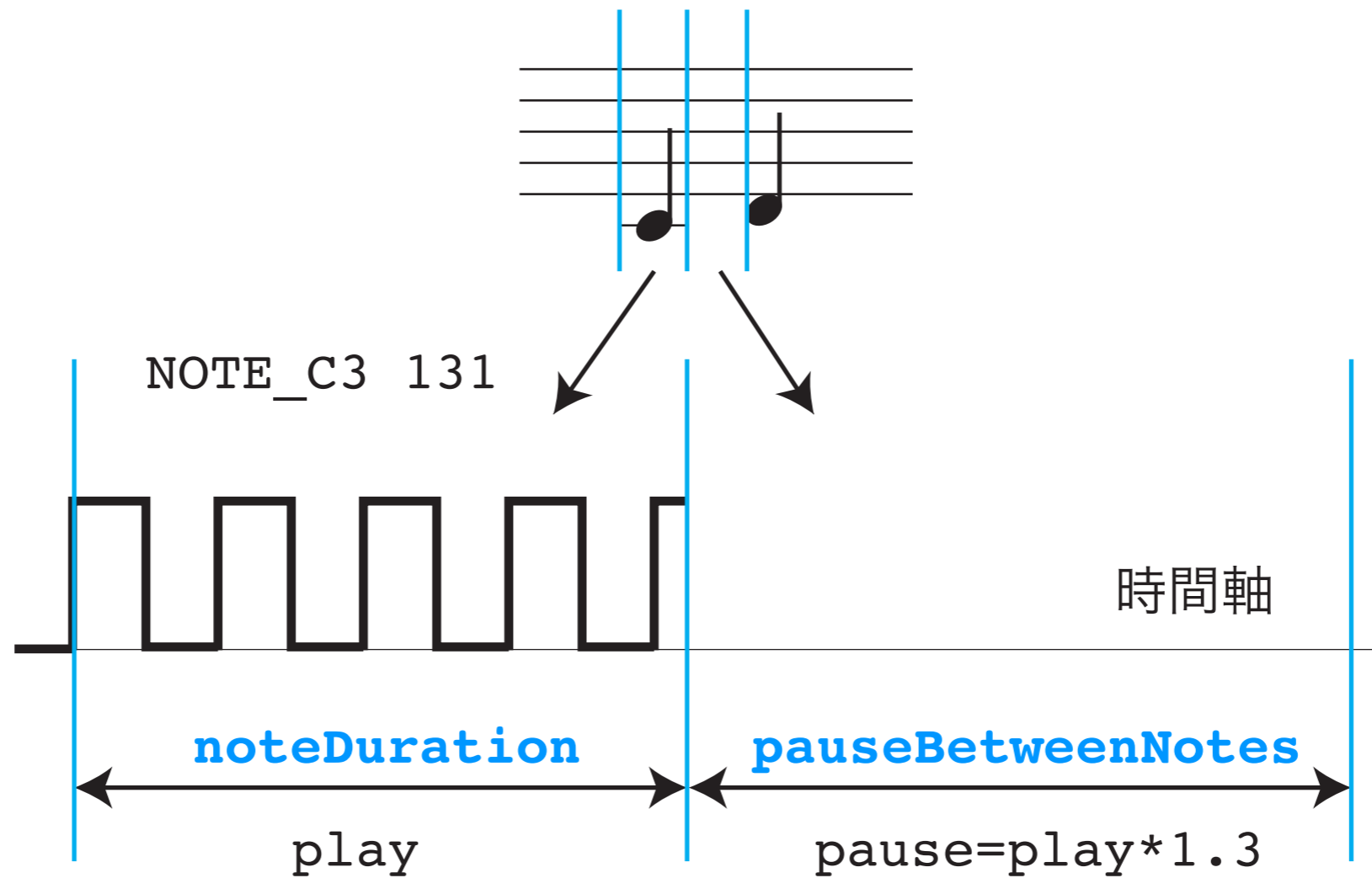
Example405A | Arduino 1.0.5
Example405A pitches.h
/*****
 * Public Constants
 *****/

#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69

```

マイコンボードへの書き込みが完了しました。  
コンパイル後のスケッチのサイズ：7,336バイト（最大容量28,672バイト）

1 Arduino Leonardo on /dev/cu.usbmodem1421

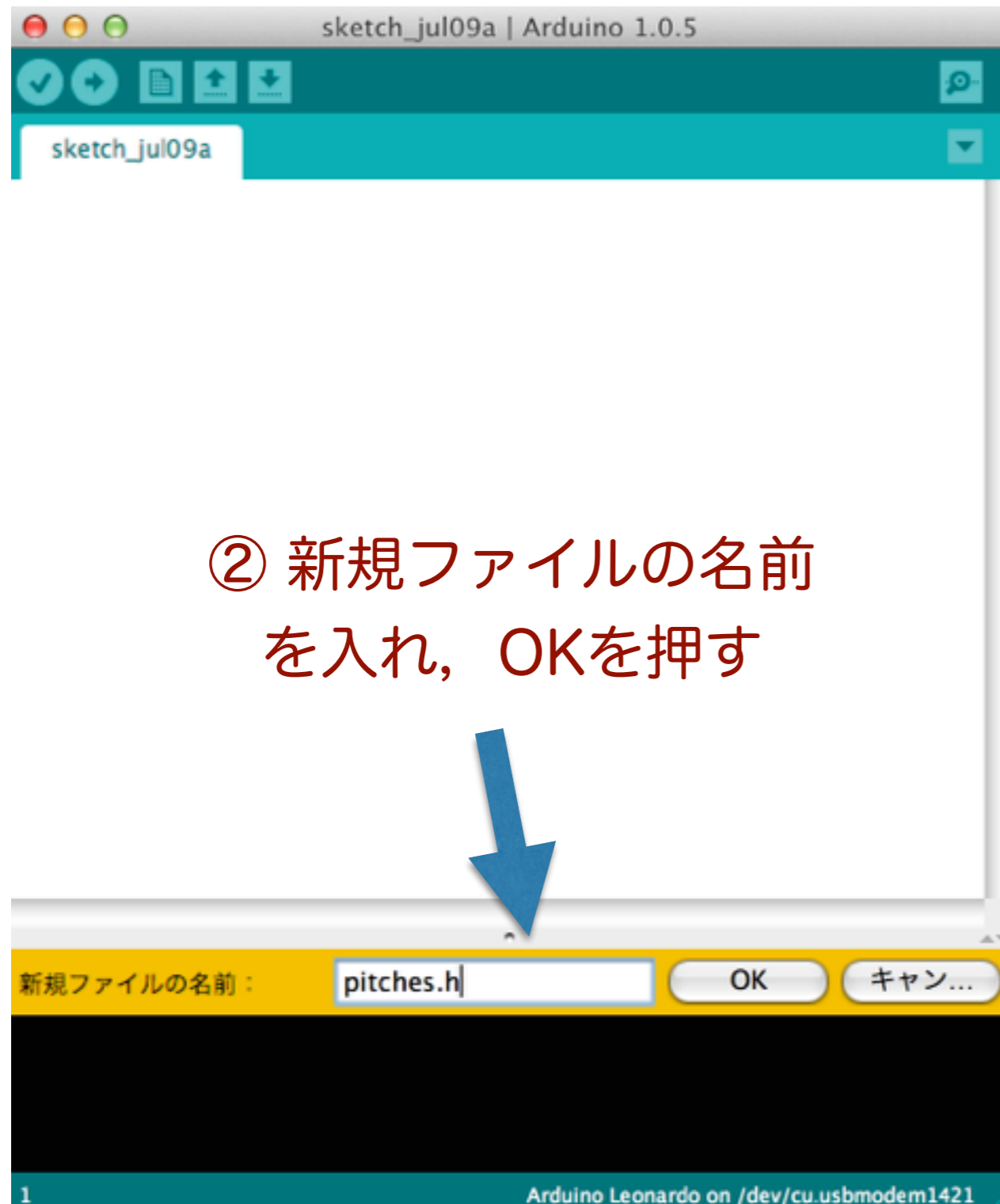


```

for (int i = 0; i < 8; i++) {
  int noteDuration = 1000/noteDurations[i];
  tone(buzzerPin, melody[i],noteDuration);
  int pauseBetweenNotes = noteDuration * 1.30;
  delay(pauseBetweenNotes);
  noTone(10);
}

```

## ヘッダー・ファイルをつくる



① この部分をクリック  
「新規タブ」を選択

② 新規ファイルの名前  
を入れ，OKを押す

③ 新規ファイルpitches.h  
ができるので，コード  
をコピーして終わり