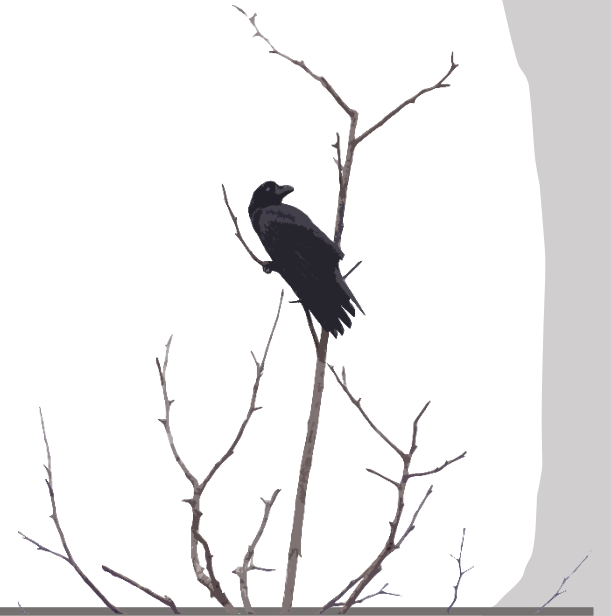


徳島大学 大学開放実践センター 公開講座

## センサのしくみを知ろう(基礎編)



徳島大学大学院社会産業理工学研究部総合技術センター  
徳島大学理工学部理工学科情報光システムコース  
技術専門職員 辻 明典 博士(工学)

# 講座日程

- ▶ センサのしくみを知ろう(基礎編)
- ▶ 講師：辻 明典(徳島大学大学院社会産業理工学研究部総合技術センター)  
桑折 範彦(徳島大学名誉教授)  
川上 博(徳島大学名誉教授)
- ▶ 曜日・時間：土曜日 10時00分～11時30分
- ▶ スケジュール：
  - ① 5/20 概要, 開発環境
  - ② 5/27 温度をはかる
  - ③ 6/3 明るさをはかる
  - ④ 6/10 モータを動かす
  - ⑤ 6/17 動きをはかる
  - ⑥ 6/24 センサを組み合わせる

# アクチュエーター

## ▶ アクチュエーター

- 入力されたエネルギーを物理的運動に変換，機械・電気回路を構成する機械要素



サーボモーター

制御基板，モーター，ギヤボックスが一体化され，制御信号により一定角を保持

応用例) ラジコン，ロボットなど



振動モーター

重心に偏りを持たせて作られた回転子を回転させて振動を発生

応用例) 携帯電話，ゲームコントローラー

# JJ5マイコンボード (Arduino Nano互換)

D9 : LED(赤色)  
D11 : LED(緑色)

A0 : 温度センサ(LM61CIZ)

A1 : 可変抵抗

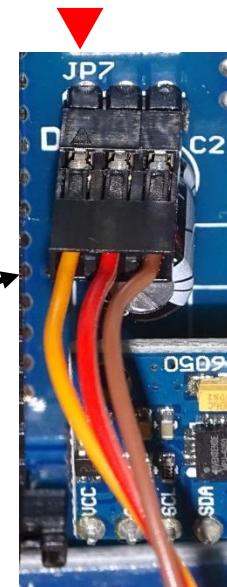
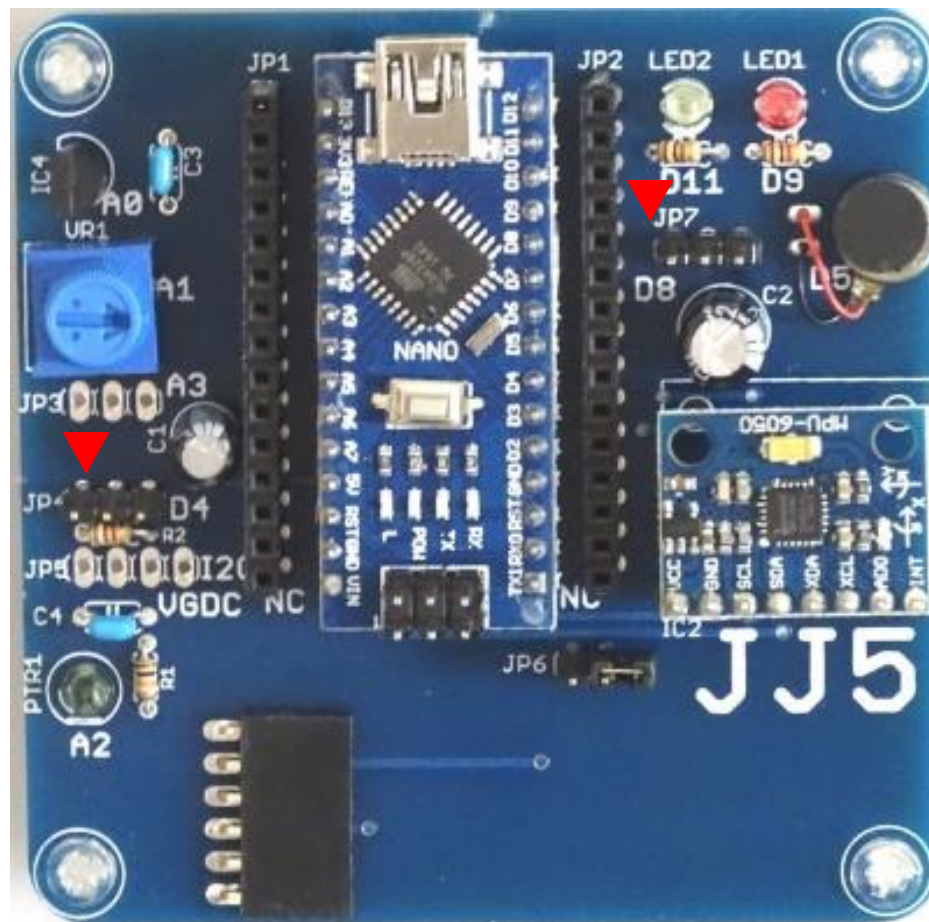
D4 : デジタル温度センサ  
(DS18B20)

A2 : 照度センサ  
(NJL7502L)

D5 : 振動モーター

D6 : サーボモーター

MPU6050 :  
加速度・ジャイロセンサ



# サーボモーター

## ▶ サーボモーター

## ▶ 仕様

- ・ マイクロサーボ SG-90
- ・ PWMサイクル：20 ms
- ・ 制御パルス：0.5 ms ～2.4 ms
- ・ 制御角：±約90° (180° )
- ・ 配線：茶 = GND, 赤 = 電源 [+], 橙 = 制御信号  
[JRタイプ]
- ・ トルク：1.8 kgf・cm
- ・ 動作速度：0.1 秒 / 60度
- ・ 動作電圧：4.8 V (～5 V)
- ・ 温度範囲：0℃～55℃
- ・ 外形寸法：22.2 x 11.8 x 31 mm重量：9 g



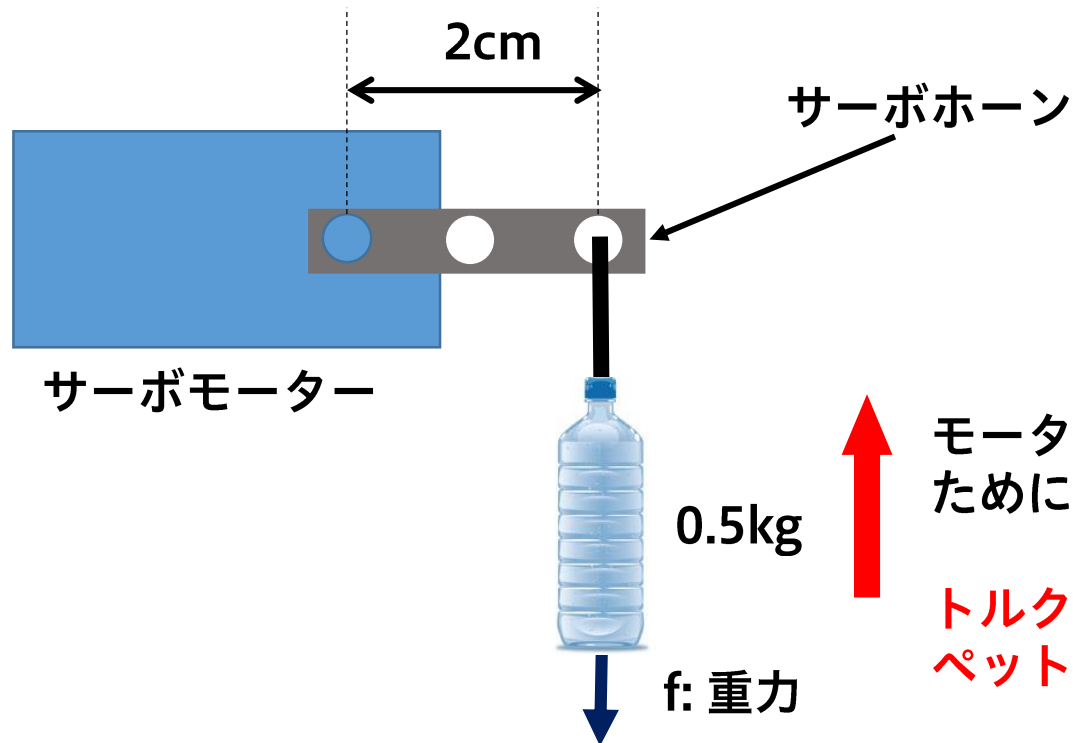
サーボモーター

トルクの単位：kgf・cm  
モーター軸から1cm先でN[kg]  
のものを持ち上げる力

# サーボモーターのトルク

## ▶ トルク

- 単位 :  $\text{kgf} \cdot \text{cm}$
- モーター軸から1cm先でN[kg]のものを持ち上げる力



kg(キログラム) : 重さ (質量)  
f(フォース) : 力(地球上では重力)  
1kgf(キログラムフォース) : 0.98N  
1N(ニュートン) : 0.102 kgf

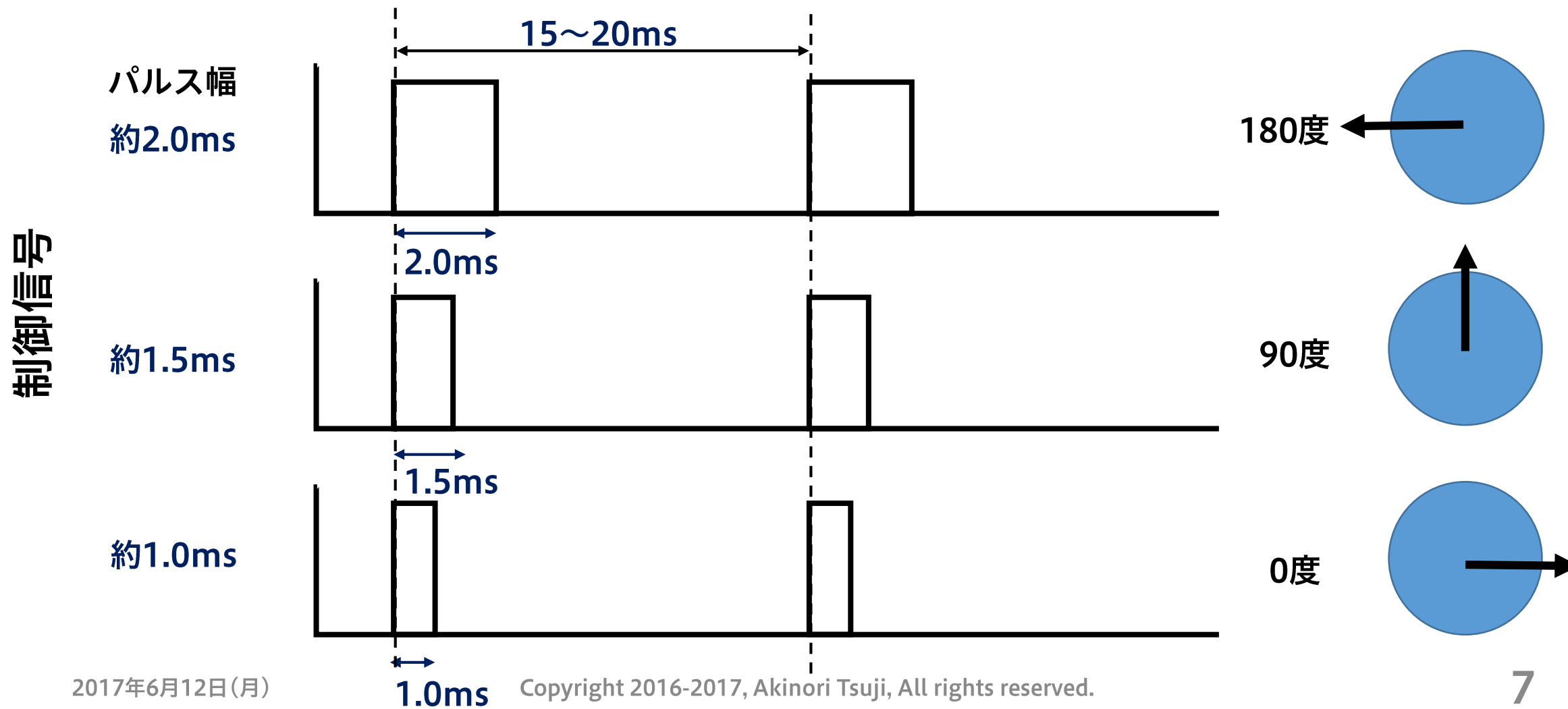
モーター軸から2cmにつり下げた0.5kgのものを持ち上げるために必要なトルク :  $0.5\text{kg} \times 2\text{cm} = 1.0 \text{ kgf} \cdot \text{cm}$

トルク :  $1.0 \text{ kgf} \cdot \text{cm}$  以上のサーボモーターを使用  
ペットボトル(500ml)に水を入れて試してみる

# サーボモーターの角度制御(一般的)

## ▶ 制御信号を与えることで指定角を保持

- パルス幅(約1.0ms~2.0ms)に回転角が対応, 90度がパルス幅約1.5ms



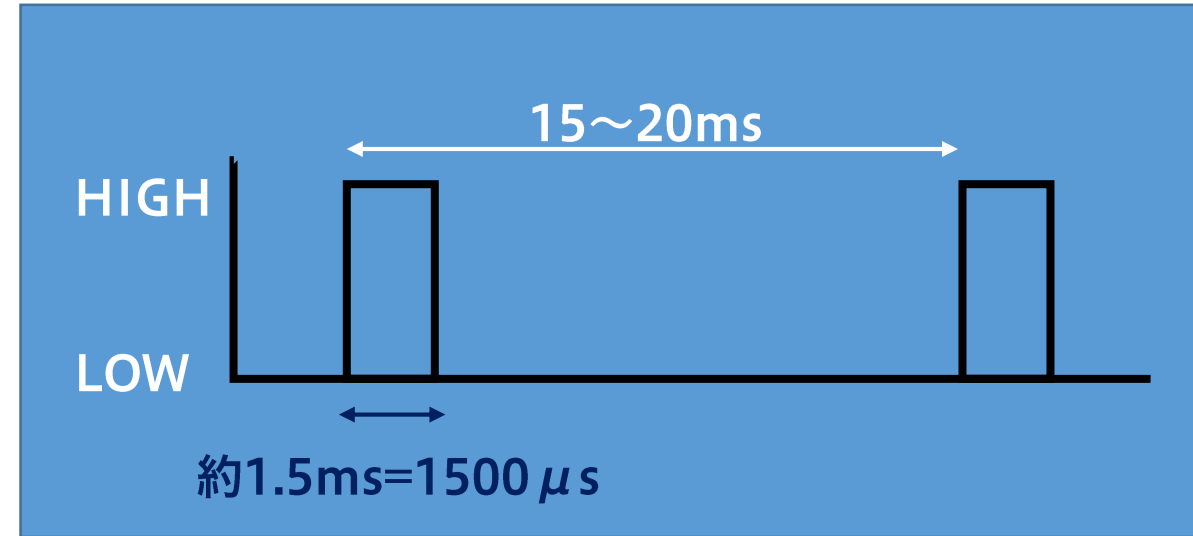
# Example0401: サーボモーターの角度制御

## ▶ サーボモーターのパルス幅を調整

```
int SERVO_PIN = 6; // 6番ピン：サーボモーター
```

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

```
void loop() {  
  // パルスを繰り返し生成  
  digitalWrite(SERVO_PIN, HIGH);  
  delayMicroseconds(1500); // パルス幅の間隔を変える  
  digitalWrite(SERVO_PIN, LOW);  
  delay(15); // 15ms間待つ  
}
```



パルス幅を変更して、0度、90度、180度を示すよう、パルス幅の間隔を調整

使用しているサーボモーターの仕様に合わせてパルス幅を調整してみよう

※ パルス幅が許容範囲を超えるとモーターから異音がするので注意



# Example0402: サーボモーターの角度制御(ライブラリ使用)

## ▶ サーボモーターの回転角を調整

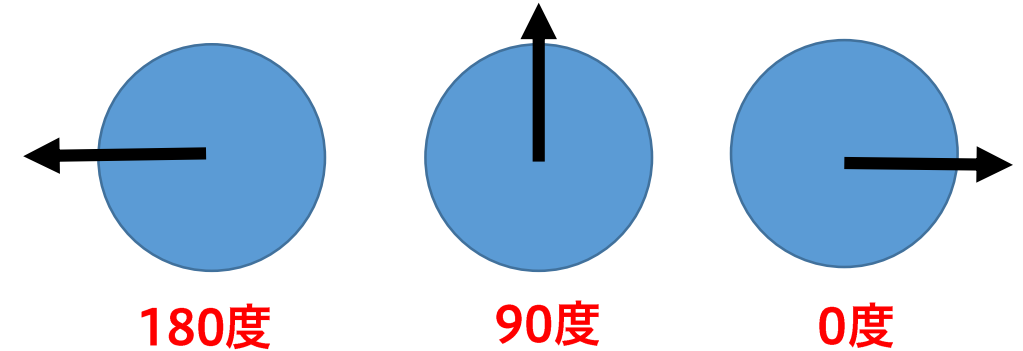
- ライブラリを使うと角度(0~90)1度毎に指定できる
- サーボモーターをなめらかに回すには Example0401の方法が良い

```
#include <Servo.h> // Servoライブラリを使用
```

```
Servo srv; // サーバ用のオブジェクトを生成  
int SERVO_PIN = 6; // 6番ピン: サーボモーター
```

```
void setup() {  
  srv.attach(SERVO_PIN); // 6番ピンにサーボモーターを接続  
  // srv.attach(SERVO_PIN, 500, 2400); // パルス幅の最小値, 最大値を追加して補正  
}
```

```
void loop() {  
  srv.write(90); // 90度(角度を指定)  
  delay(15); // 15ms待つ  
}
```



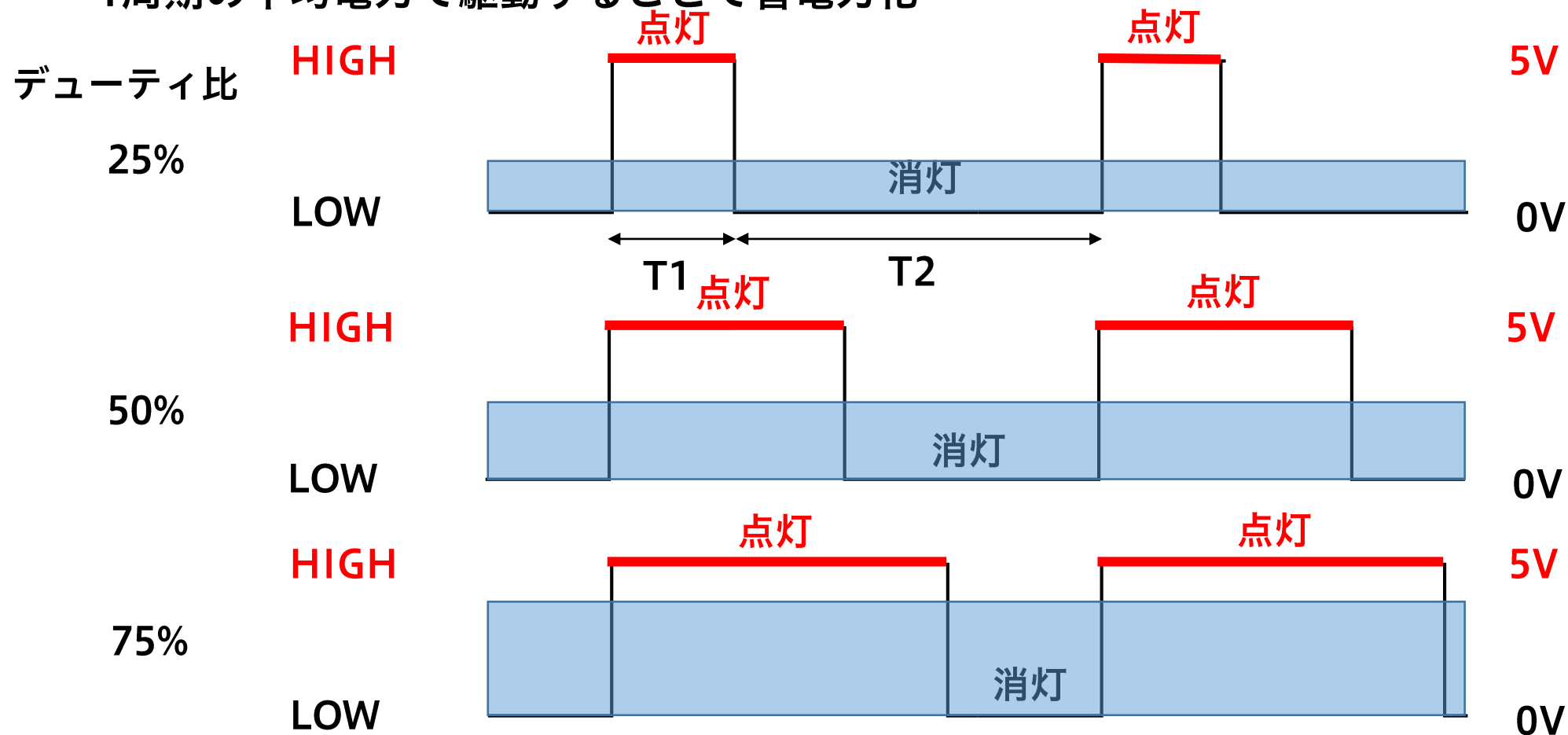
角度を変更して, 0度, 90度, 180度  
になるよう調整

角度を指定しても0度, 90度, 180度を  
示さない場合は, srv.attachにパルス幅  
の最小値, 最大値を追加しよう

# パルス幅変調(PWM)

## ▶ PWM(Pulse Width Modulation)

- デューティ比 (HIGHとLOWの比率) =  $T1 / (T1 + T2) \times 100\%$
- 1周期の平均電力で駆動することで省電力化



# Example0403: LEDのPWMによる調光

## ▶ LEDの明るさをPWMで調光

```
const int LED_PIN = 12;
void setup() {
  pinMode(LED_PIN, OUTPUT);
}
void loop() {
  // 徐々に明るく
  for (int brightness = 0; brightness < 255; brightness++) {
    analogWrite(LED_PIN, brightness);
    delay(2);
  }
  // 徐々に暗く
  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(LED_PIN, brightness);
    delay(2);
  }
  delay(100);
}
```

analogWrite関数を使うとPWM波を生成可能

analogWrite(val) val: 0~255  
(デューティ比: 0%~100%)

周波数: 976Hzまたは490Hz

# Example0404: 振動モーターのPWM制御

- ▶ 振動モーターを500ms間ON, 1000ms間OFF

```
const int MOTOR_PIN = 5;
```

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

```
void loop() {  
  analogWrite(MOTOR_PIN, 255); // モーター回転 : 0~255  
  delay(500);  
  analogWrite(MOTOR_PIN, 0); // モーター停止  
  delay(1000);  
}
```

デューティ比やdelayの時間を調整して  
振動の大きさや伝わる時間を変更

