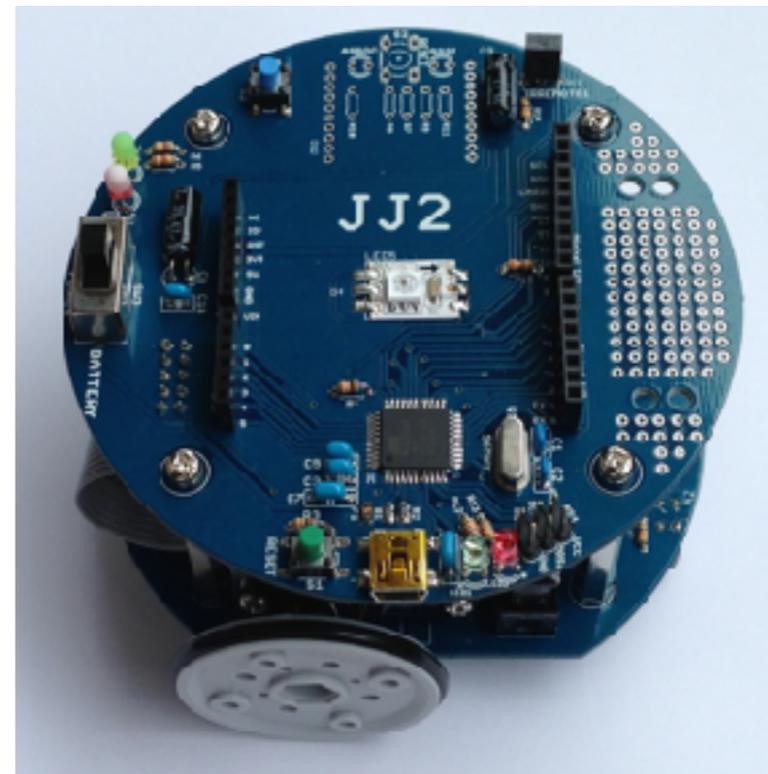
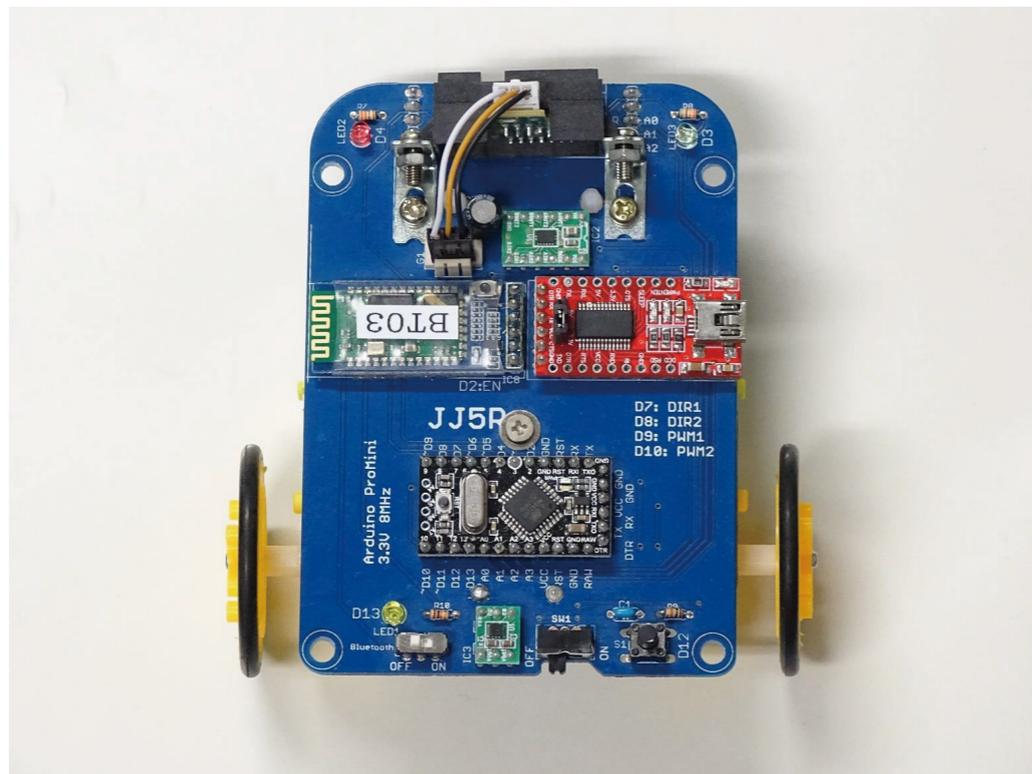


# 無線で動くロボットを作ろう

## 第3回 ロボットのモーター2 (応用動作)



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

川上 博

2017/10/21

## 今日のテーマ

---

◎ 前回の復習：digital/analog, read/write, PWM

---

◎ ロボットの走行実験：前進, 後退, 左折, 右折

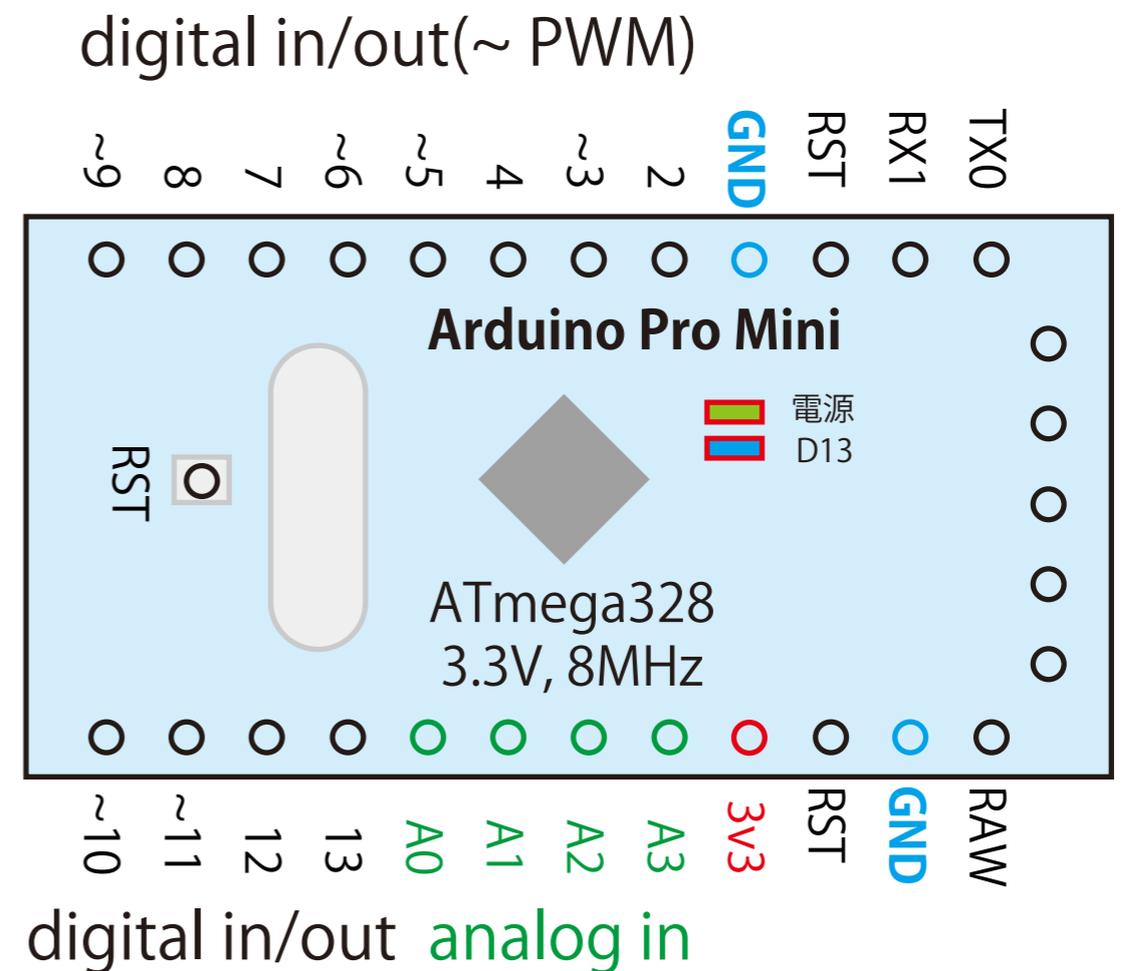
◎ ロボットの走行実験：キャリブレーション

---

◎ Logo: turtle graphics：簡単な図形を描く

# Arduinoボード上のLEDを点滅させる

- ① USBケーブルでPCとArduinoボードをつなぐ
- ② Arduino IDE (ソフト) を起動する
- ③ スケッチを書く :  
ファイル > スケッチの例 > 01.Basics > Blink
- ④ 検証ボタンを押す
- ⑤ マイコンボードに書き込む



## ツールの選択

- ① USBケーブルでPCとArduinoボードをつなぐ
- ② Arduino IDE（ソフト）を起動する



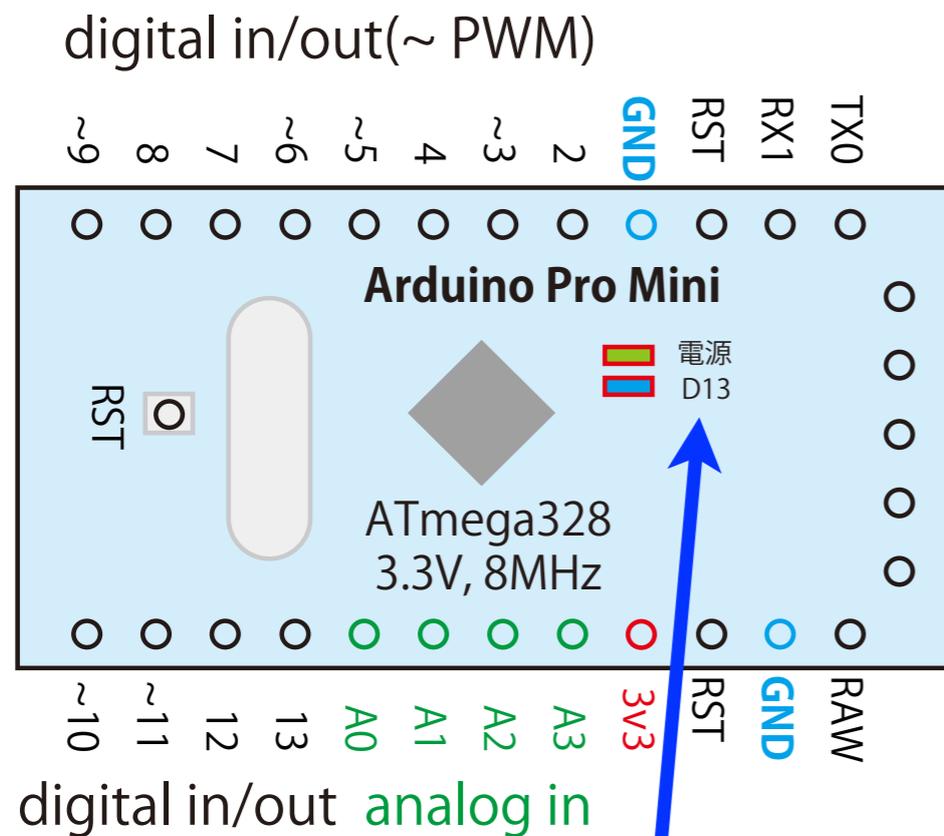
- ③ スケッチを書く：

ファイル > スケッチの例 > 01.Basics > Blink

# Arduino : 最初のプログラム

② プログラムを検証する  
(Verifyボタンを押す)

③ プログラムを書き込む  
(Uploadボタンを押す)



Example201A | Arduino 1.6.9 Hourly Build 2016/04/28 04:12

```

Example201A
/* Example 201A: Blink
   Turns on an LED for one second,
   then off for one second, repeatedly.
 */

const int led=13;

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

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}

```

① スケッチを書く

/dev/cu.usbseria A400F418上©A-duino Pro or Pro Mini, ATmega328 (3.3V, 8 MHz)

④ Arduino上のLEDが点滅する

## プログラムを読んでみよう

---

### 注釈

```
/* Example 201A: Blink  
Turns on an LED for one second,  
then off for one second, repeatedly.
```

```
*/
```

```
const int led = 13;
```

変数の宣言

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

初期設定

```
void loop() {  
  digitalWrite(led, HIGH);  
  delay(1000);  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

繰り返し実行する本体

digital - analog, write - read

# 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)

```
}
```

無限ループ：動きっぱなしのプログラム

# 関数に仕事をさせる

出力変数

入力変数

$$y = f(x)$$

関数名

出力変数

関数名

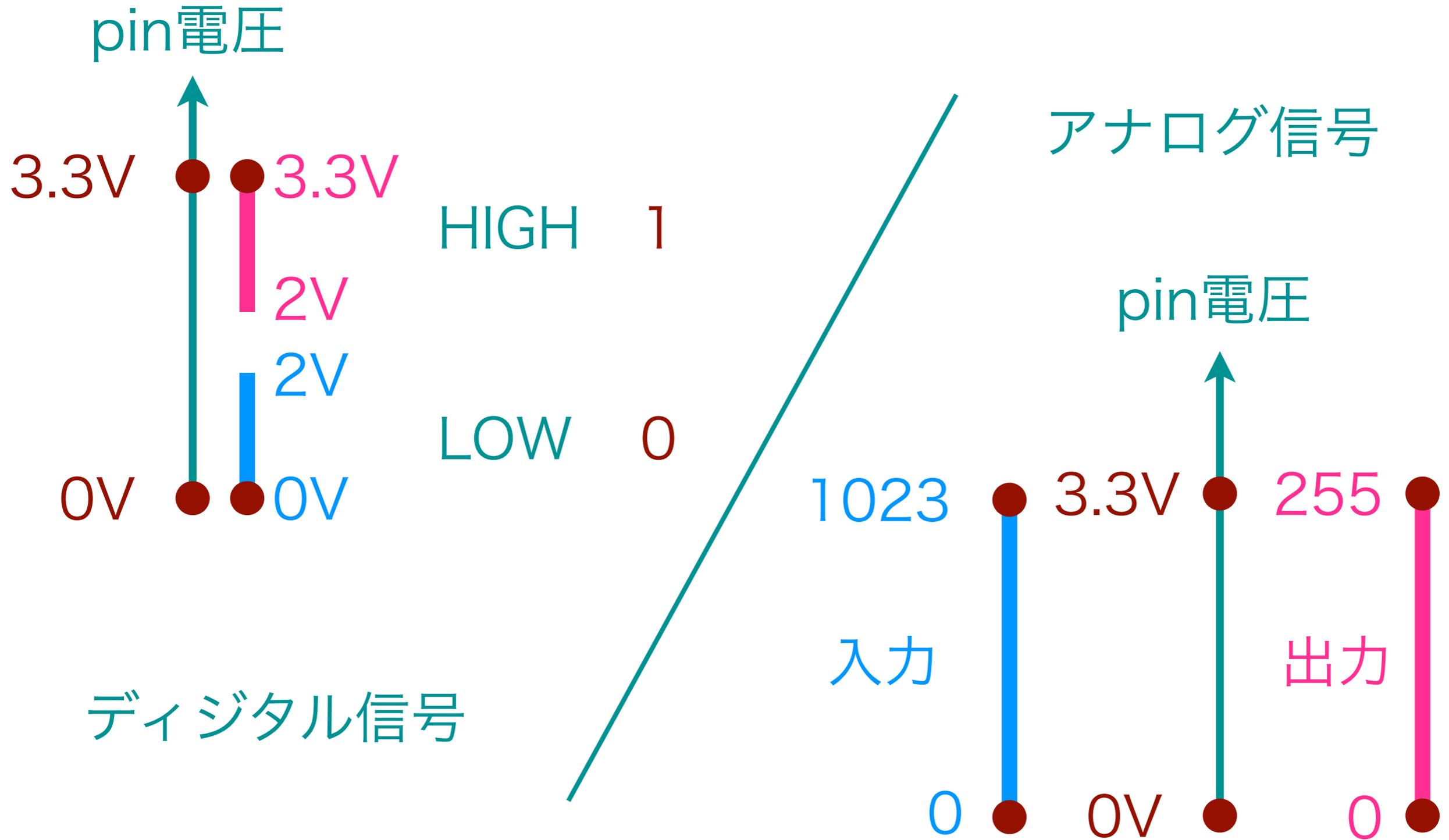
入力変数

```
value = digitalRead(7);  
digitalWrite(12, value);
```

出力値, 出力pin番号

```
pinMode(13, OUTPUT);
```

# Digital信号 と Analog信号

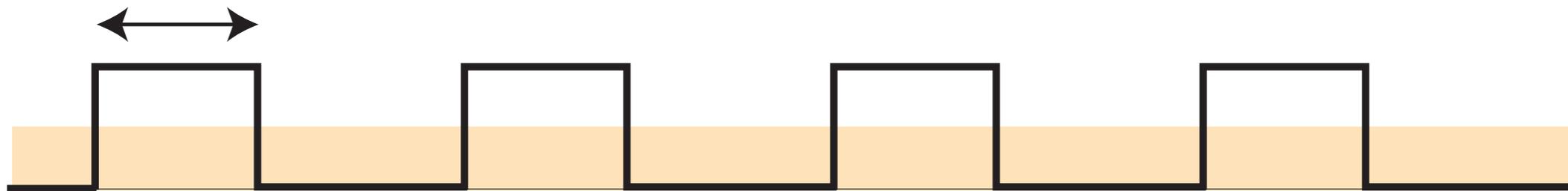
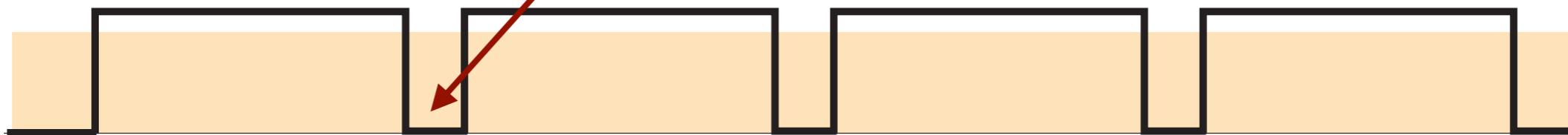


# LEDの明るさが変わっているようにみえる

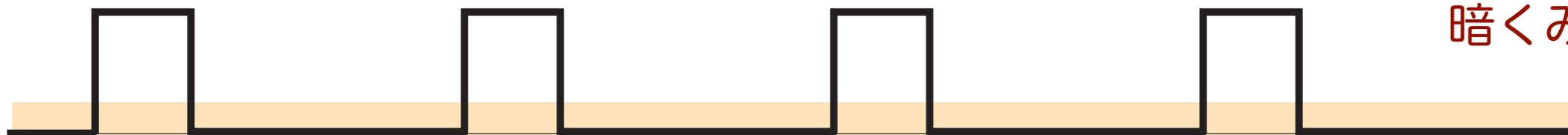
```
digitalWrite(13, HIGH);
```

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

明るくみえる



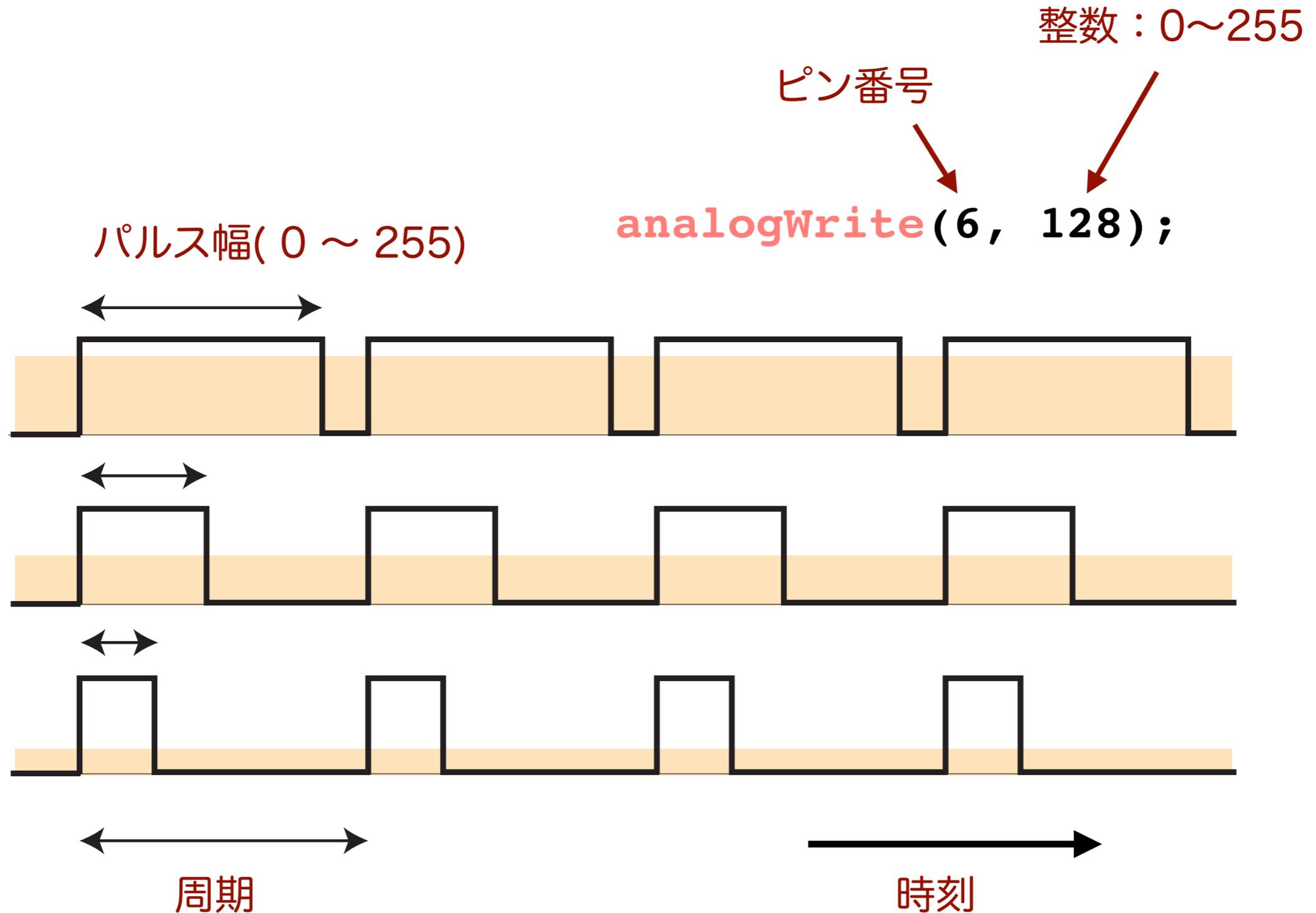
暗くみえる



周期(20 ms)

時刻

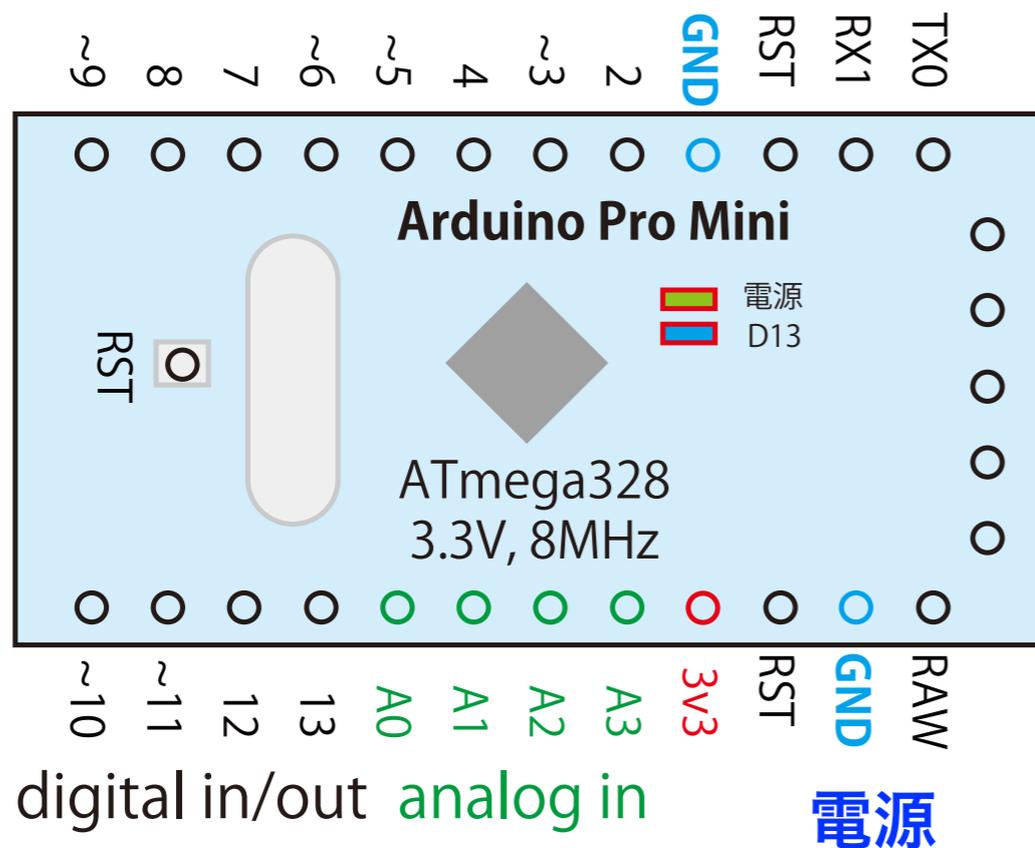
# PWM : pulse width modulation : パルス幅変調



# JJ5rのピン配置

## デジタル・入出力ピン

digital in/out(~ PWM)



デジタル  
入出力ピン

アナログ・入力ピン

```
analogRead(0);
```

```
analogWrite(6, 128);
```

~印のピン : アナログ出力に使う

```
digitalRead(7);
```

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

13ピン : LEDにつながっている

## JJ5 ポート一覧

Arduino Pro Mini 互換 / ATmega 328P / 3.3V/8MHz

ポート	接続先	型番	ライブラリ	注釈
D0	Bluetooth (TX0)	HC-05	Serial	通信速度: 38400bps
D1	Bluetooth (RX1)	HC-05	Serial	ペアリング: 1234
D2	Bluetooth (INT0)	HC-05	digitalWrite	書き込み用
D3~	LED3 (右)	緑	digitalWrite	
D4	LED2 (左)	赤	digitalWrite	
D5~	—			
D6~	—			
D7	左モーター方向	DRV8835	digitalWrite DRV8835Shield	
D8	右モーター方向	DRV8835	digitalWrite DRV8835Shield	
D9~	左モーターPWM	DRV8835	analogWrite DRV8835Shield	
D10~	右モーターPWM	DRV8835	analogWrite DRV8835Shield	
D11~	—			
D12	スイッチ (スタート)	タクト	digitalRead	
D13	LED1 (電源)	黄	digitalWrite	
A0	フォトインタラプタ右	LBR127HLD	analogRead	
A1	フォトインタラプタ中央	LBR127HLD	analogRead	
A2	フォトインタラプタ左	LBR127HLD	analogRead	
A3	距離センサ	GP2Y0A21YK	analogRead	オプション

~ = PWM, analogWrite

辻先生の配布資料より引用

# Read 「読み込む」と Write 「書き出す」

```
analogRead(0);
```

センサー

アナログ入力

アナログ出力

LED

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

```
digitalRead(5);
```

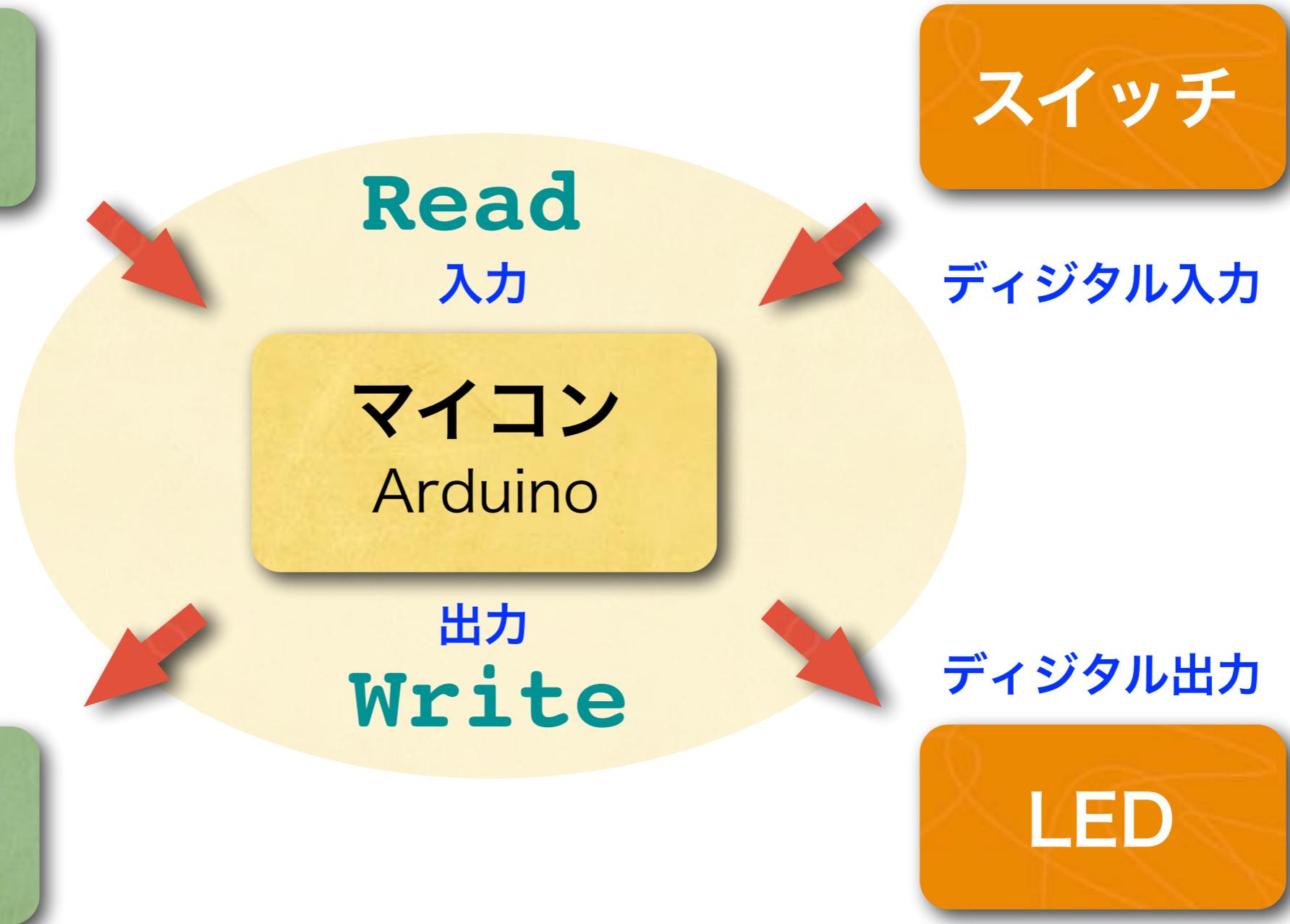
スイッチ

デジタル入力

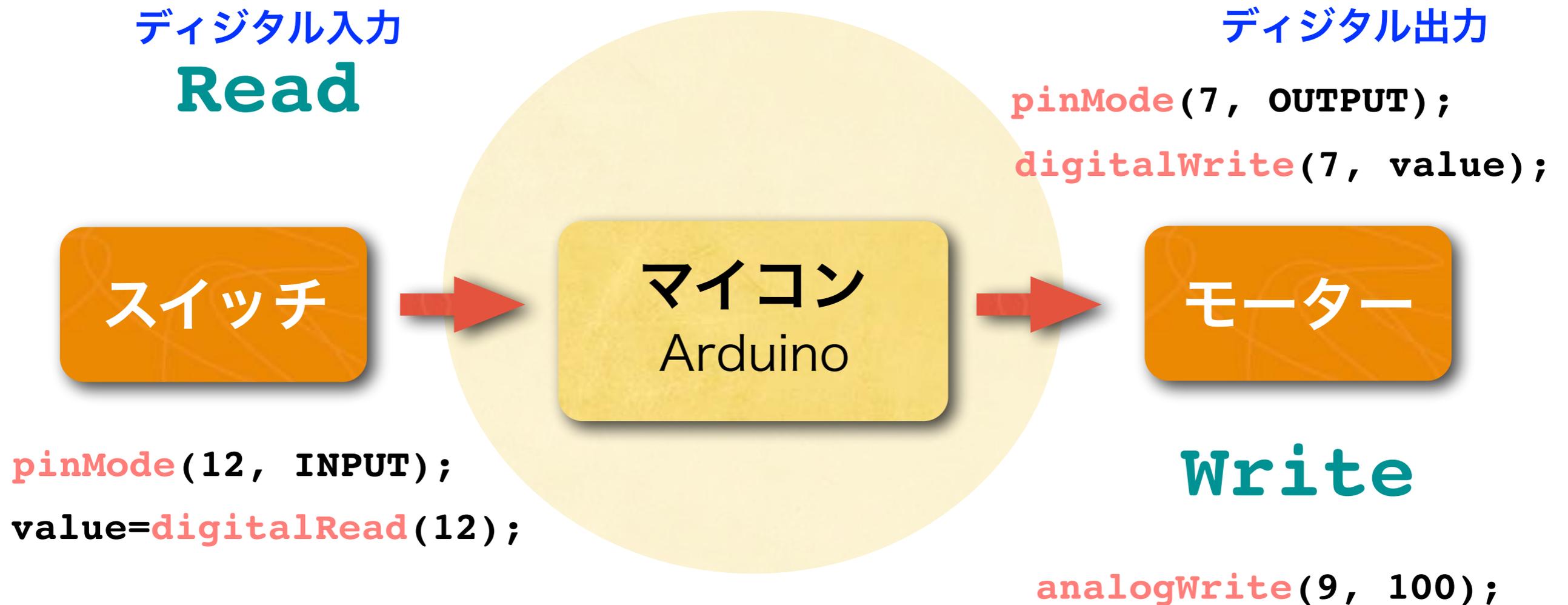
デジタル出力

LED

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

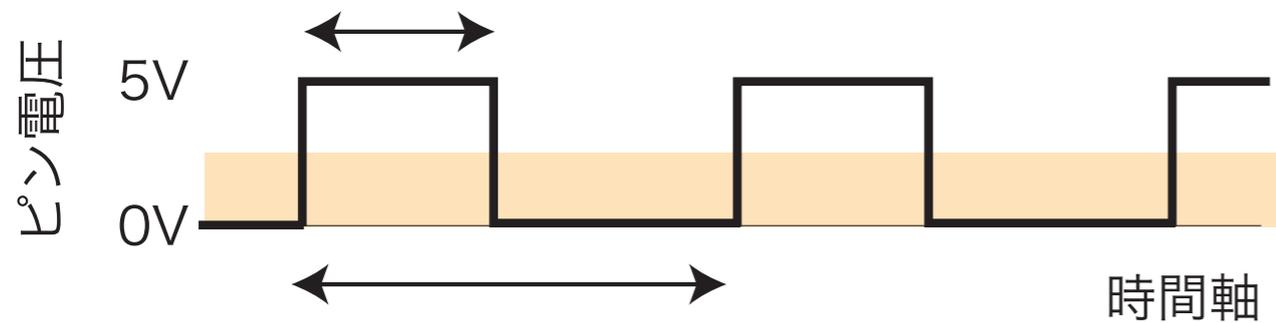


# モーターを回す：回転方向・回転速度



# analogWrite

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



**analogWrite()**はpinMode  
を指定しなくてよい

整数：0～255

ピン番号

```
analogWrite(9, 100);
```

マイコン  
Arduino



モーター

アナログ出力

ブレッドボード上のモーターを制御する

- ◎ **ロボットの走行実験：前進, 後退, 左折, 右折**
- ◎ **ロボットの走行実験：キャリブレーション**
- ◎ **Logo: turtle graphics：簡単な図形を描く**

## JJ5rの運動

ロボット：水平面上にある2つの車輪を持つ剛体

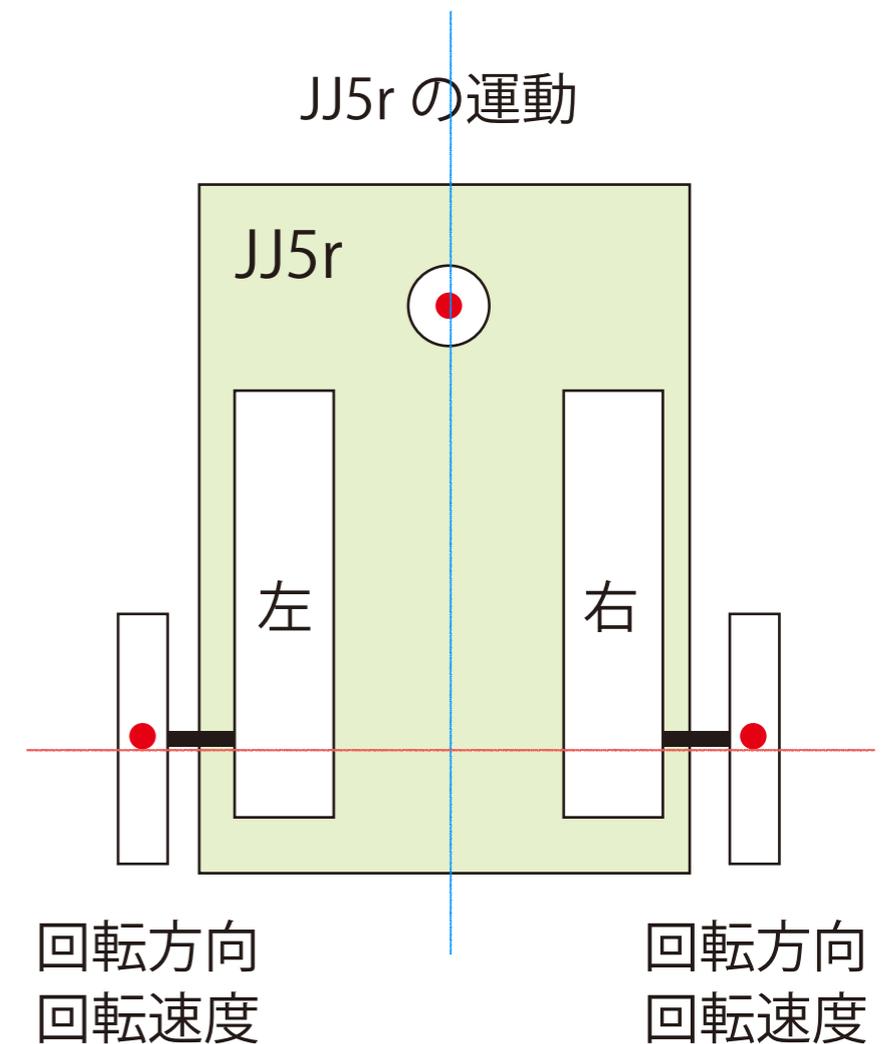
操作量①：左右にある車輪の回転速度

操作量②：操作を加える時間

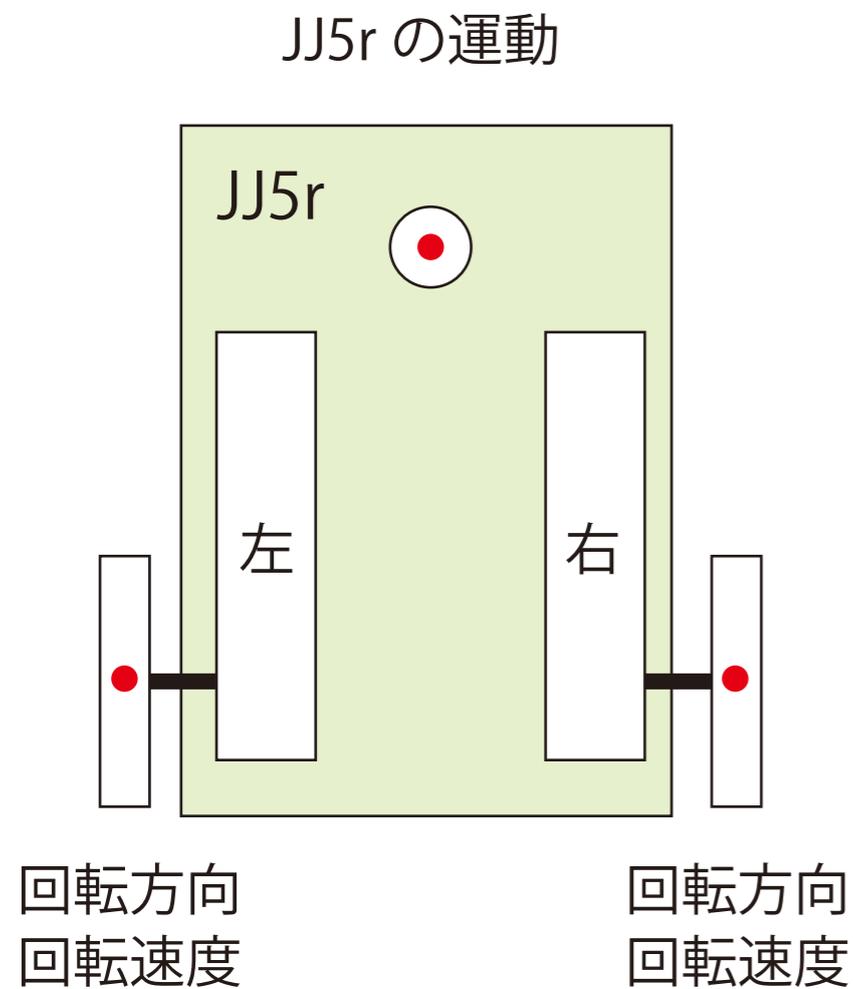
直線運動：左右の回転速度が等しい運動

回転運動：車軸の中心を固定した運動

$$\text{距離} = \text{速度} \times \text{時間}$$



# JJ5rの操作：車輪の回転方向と回転速度



pin番号	回転方向	回転速度
右車輪	<b>D7</b> HIGH/LOW	<b>D9</b> 0 ~ 255
左車輪	<b>D8</b> HIGH/LOW	<b>D10</b> 0 ~ 255
	<b>digitalWrite</b>	<b>analogWrite</b>

共通操作：回転持続時間  $\tau$

```

20 // Example 301: same as Example 104a
// モーター一回転方向
// 前進: LOW
// 後退: HIGH

const int MOTOR_R_CWCCW = 7;
const int MOTOR_L_CWCCW = 8;
const int MOTOR_R_PWM = 9;
const int MOTOR_L_PWM = 10;

void motor(int left, int right, int left_c, int right_c) {
  if (left_c == LOW) {
    digitalWrite(MOTOR_L_CWCCW, LOW); // 左モーターを前進方向に回す
  } else {
    digitalWrite(MOTOR_L_CWCCW, HIGH); // 左モーターを後退方向に回す
  }
  if (right_c == LOW) {
    digitalWrite(MOTOR_R_CWCCW, LOW); // 右モーターを前進方向に回す
  } else {
    digitalWrite(MOTOR_R_CWCCW, HIGH); // 右モーターを後退方向に回す
  }
  analogWrite(MOTOR_L_PWM, right); // 左モーターの回転速度(0 - 255)
  analogWrite(MOTOR_R_PWM, left); // 右モーターの回転速度(0 - 255)
}

```

## JJ5rの運動：直線運動と回転運動

---

```
int sp=100;
motor(sp, sp, LOW, LOW);          fwd(int tau); // forward
delay(1000); // LOW:前進
motor(sp, sp, HIGH, HIGH);       bwd(int tau); // backward
delay(1000); // HIGH:後退
motor(0, 0, HIGH, HIGH);         stp(int tau); // stop
delay(1000); // 停止
motor(sp, sp, HIGH, LOW);        rotL(int tau); //rotation to left
delay(1000); // 左折回転
motor(sp, sp, LOW, HIGH);        rotR(int tau); //rotation to right
delay(1000); // 右折回転
```

## 直線運動と回転運動のキャリブレーション

---

`fwd(int tau); // forward`

10 cm 前進せよ

`bwd(int tau); // backward`

7 cm 後退せよ

`stp(int tau); // stop`

2 秒間停止せよ

`rotL(int tau); // rotation to left`

90° 左に回転せよ

`rotR(int tau); // rotation to right`

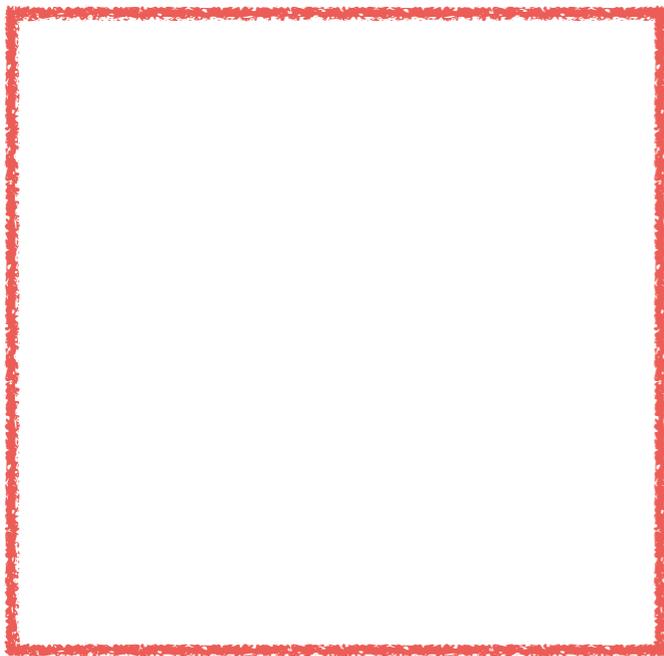
72° 右に回転せよ

## 「10cm進んで90度回転する」を繰り返して描かれる図形

```
void fwd(float cm) {  
  int tau;  
  
  motor(sp, sp, LOW, LOW);  
  tau = 2500.0*cm/20.0;  
  delay(tau);  
}
```

```
void rotL(float deg) {  
  int tau;  
  
  motor(sp, sp, HIGH, LOW);  
  tau = 3780.0*deg/360.0;  
  delay(tau);  
}
```

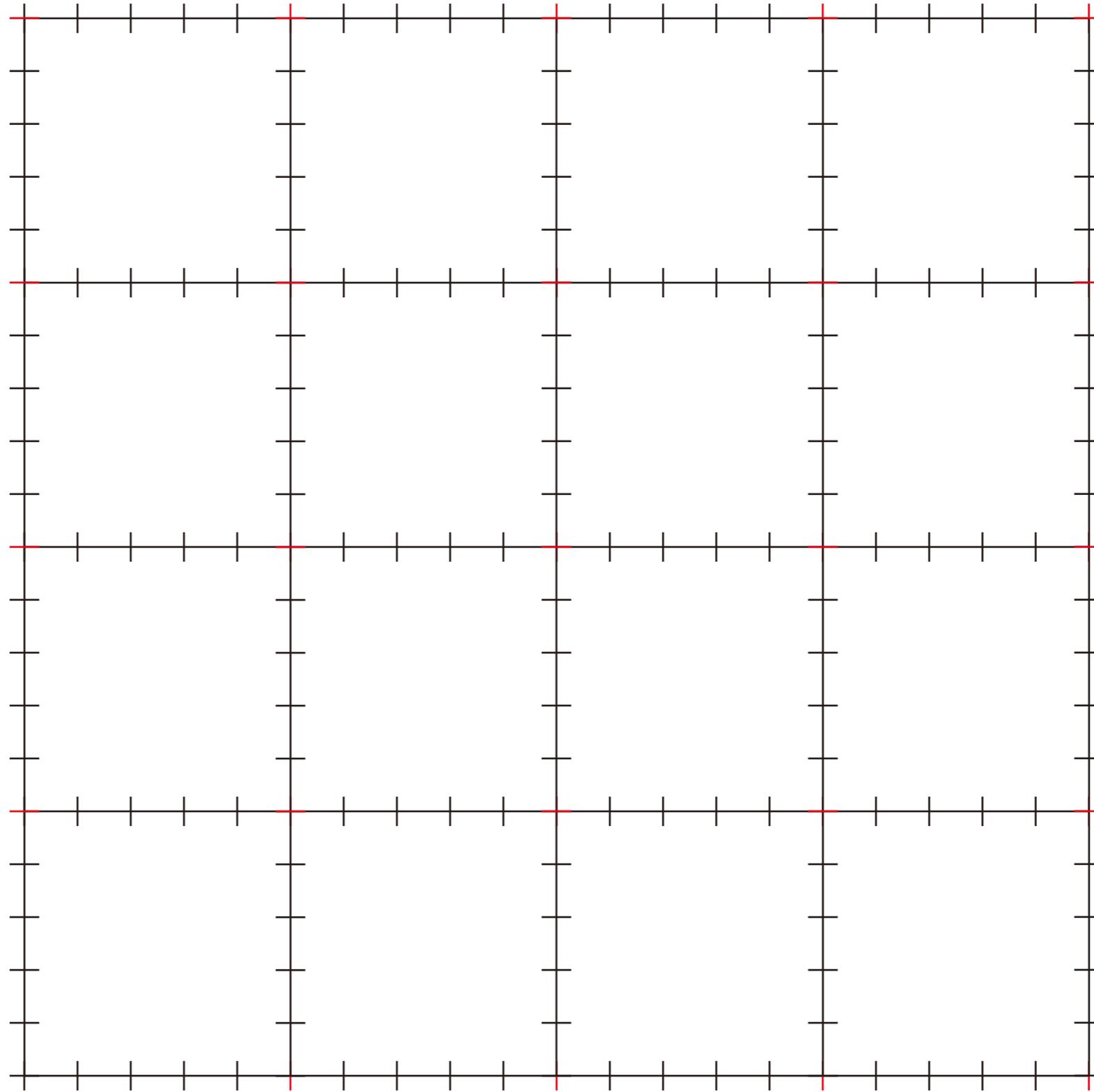
2500msで20cm進んだ, 3780msで360°回転した

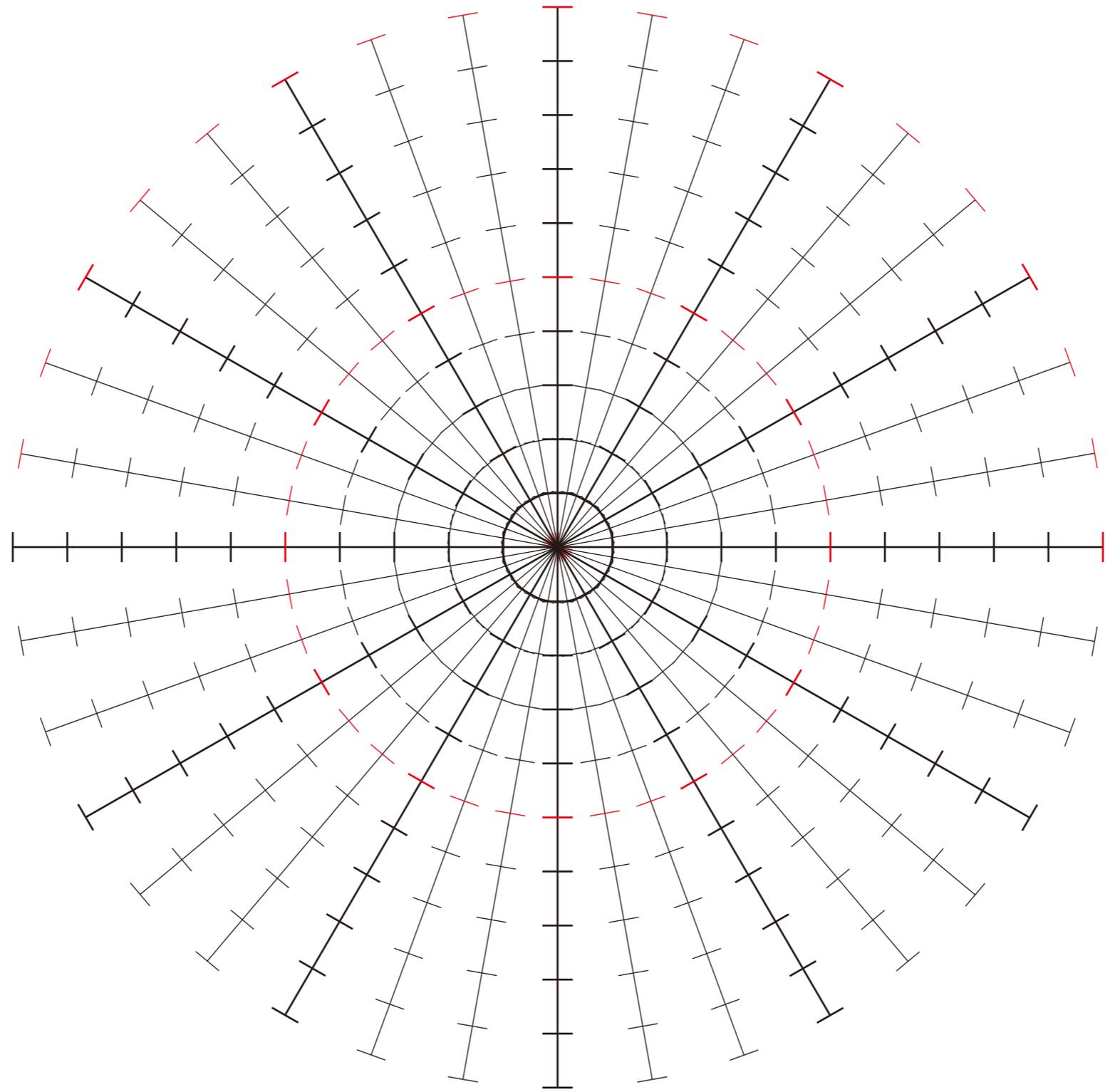


```
void loop() {  
  fwd(10.0); // forward 10.0 cm  
  rotL(90.0); // turn to 90.0 degrees  
  stp(1000); // stop 1000 milli-seconds  
}
```

Logo : turtle graphics

<http://el.media.mit.edu/logo-foundation/index.html>





# 参考書：Arduinoをはじめよう，第3版，2015，¥2,000



Arduinoをはじめよう

第3版

わかりやすく、長く使える

O'REILLY