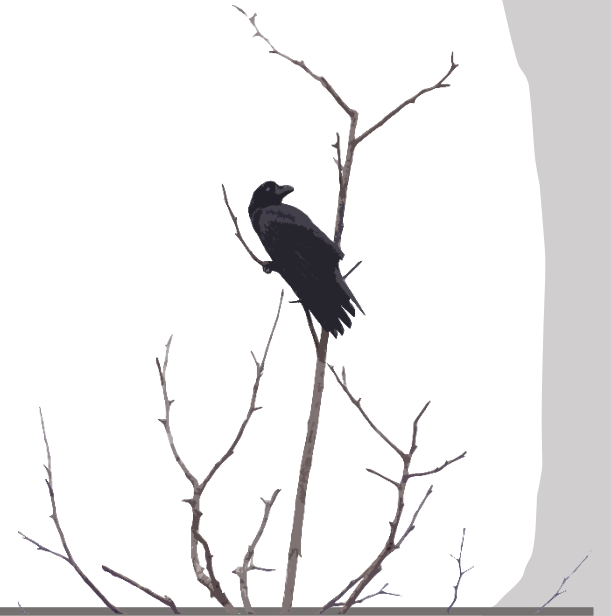


徳島大学 大学開放実践センター 公開講座

## 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)

- ▶ 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

# 概要

## ▶ 講座の目的

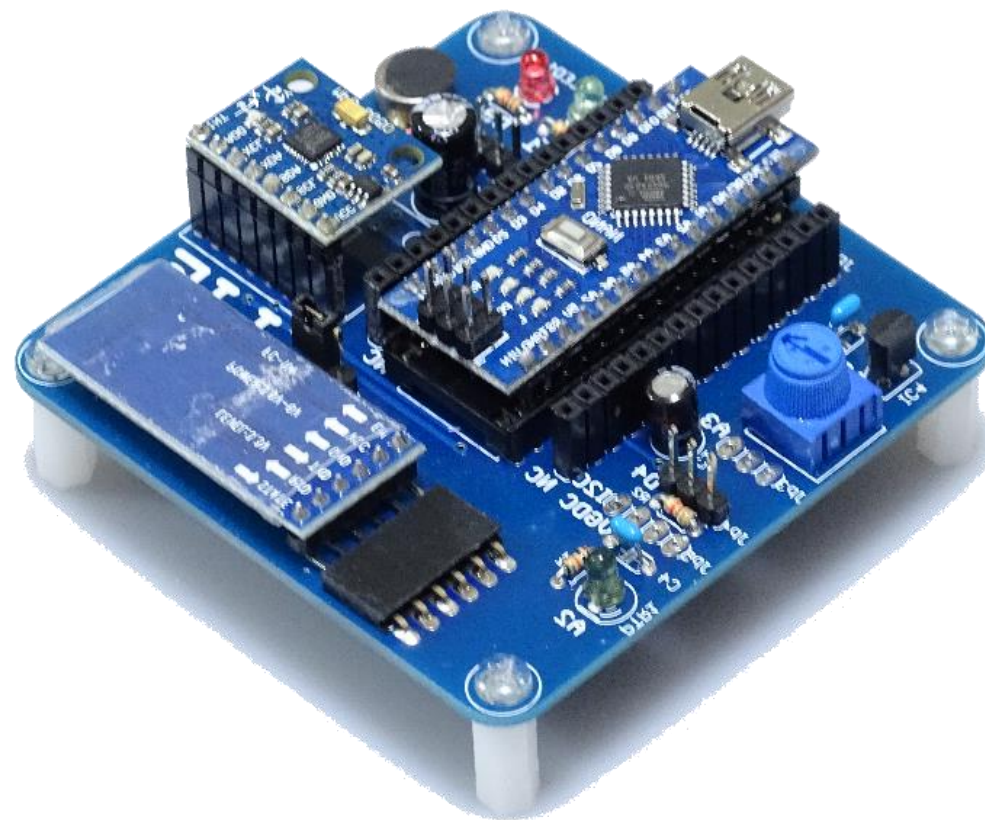
- センサのしくみを理解して使いこなせる

## ▶ 講座の実施方法

- 講義
- 演習

## ▶ 本日の予定

- 講座概要
- 配布部品の確認
- センサ開発環境をつくる
- はじめてのスケッチ
- ブレッドボード



## 配布部品の確認

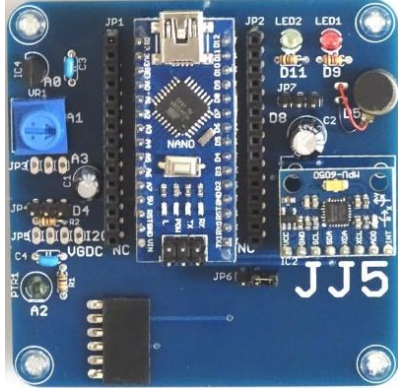
---



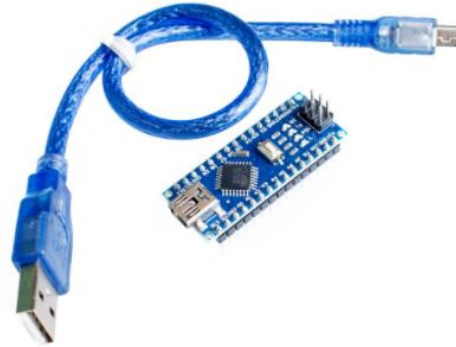
# 配布部品一覧

	名称	規格	個数
1	マイコンボード	JJ5, Arduino Nano/5V/16MHz/328P	1
2	ケーブル	USBケーブル	1
3	加速度・ジャイロセンサ	MPU6050	1
4	温度センサ	DS18B20	1
5	サーボモーター	SG-90	1
6	スペーサー	ねじ, スペーサー	4
7	タクトスイッチ	黒	1
8	LED	LED3mm赤	1
9	LED	LED3mm黄緑	1
10	LED	LED3mm黄	1
11	抵抗	330 $\Omega$	3
12	抵抗	10k $\Omega$	1
13	ブレッドボード	EIC	1
14	ジャンパワイヤ	数本	1

# 配布部品(1/2)



1. マイコンボード



2. USBケーブル



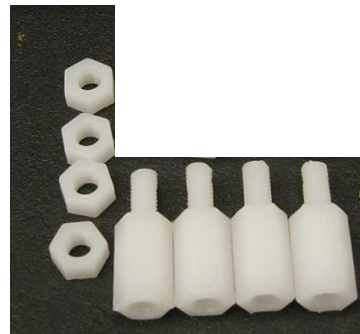
3. 加速度・ジャイロセンサ



4. 温度センサ



5. サーボモータ



6. スペーサー

(次のページに続く)

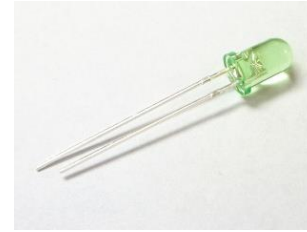
## 配布部品(2/2)



7. タクトスイッチ  
1個



8. LED (赤色)  
1個



9. LED (黄緑色)  
1個



10. LED (黄色)  
1個



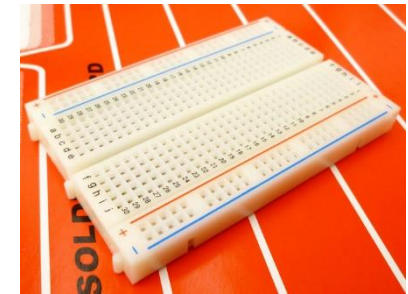
11. 抵抗  $330\Omega$   
5個



12. 抵抗  $10k\Omega$   
4個



13. ジャンパ線  
1式



14. ブレッドボード  
1個

# センサの開発環境

---





# センサ開発環境をつくる

## ▶ Arduino

- ダウンロード先
  - Arduino-1.8.2 : <http://www.arduino.cc/>

## ▶ USBドライバ

- ダウンロード先
  - CH341: [http://www.wch.cn/download/CH341SER\\_EXE.html](http://www.wch.cn/download/CH341SER_EXE.html)

## ▶ ライブラリ

- DallasTemperature: 温度センサ用
  - <http://www.arduinolibraries.info/libraries/dallas-temperature>
- I2Cdev: I<sup>2</sup>C通信用
  - <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/I2Cdev>
- MPU6050: 加速度・地磁気センサ用
  - <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>
- OneWire: 1線通信用
  - <http://www.arduinolibraries.info/libraries/one-wire>

- ▶ 講座資料(スライド, サンプルスケッチ等)
  - <http://goo.gl/IEWYSn>
- ▶ スケッチの例の使い方
  - Example01.zipをダウンロード
  - ファイルを右クリック: 「すべて展開」
  - 参照: 「ドキュメント¥Arduino」を選択して「展開」

← 圧縮 (ZIP 形式) フォルダの展開

展開先の選択とファイルの展開

ファイルを下のフォルダに展開する(F):

C:\Users#a-tsuji\Documents

参照(R)...

☒ 完了時に展開されたファイルを表示する(H)

展開(E)

キャンセル

CMS > IDX > 徳島大学 > 工学部 > 知能情報工学科 > 辻 明典 > Share > | 検索 | 人名索引 | 組織索引 | 利用方法 |

【個人】 辻 明典 / Tuji, Akinori EDB  
(Share/H29Sensor)

! 《EDB/CMSからの重要なお知らせ》 (August 30, 2015更新) お知らせを見る

コンテンツエリア

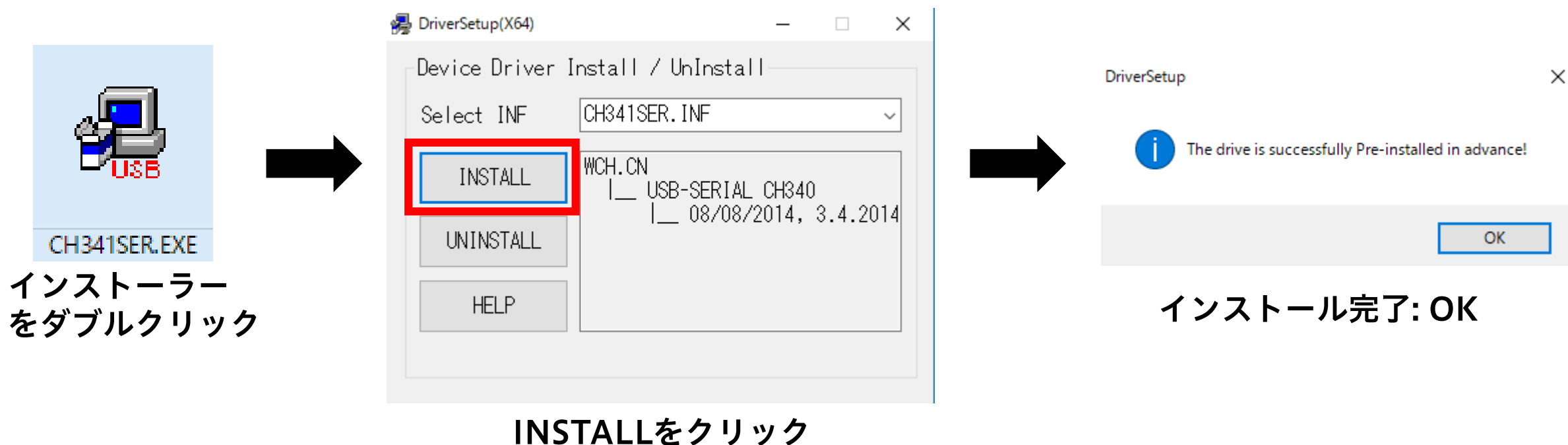
場所の情報とサーバ切替 (ヘルプ)

一般閲覧用	: <a href="http://cms.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/">http://cms.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/</a>	
学生閲覧用 (学内)	: <a href="http://cms-ldap.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/">http://cms-ldap.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/</a>	切替
学生閲覧用 (学外)	: <a href="https://cms-ldap.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/">https://cms-ldap.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/</a>	切替
教職員閲覧・登録 (ID&Pass)	: <a href="https://cms.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/">https://cms.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/</a>	切替
教職員閲覧・登録 (EDB/PKI)	: <a href="https://cms-pki.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/">https://cms-pki.db.tokushima-u.ac.jp/DAV/person/S10767/Share/H29Sensor/</a>	切替

コンテンツ

Name	Last modified	Size	Description
Parent Directory		-	このフォルダはインターネットからアクセス可能です。 (→詳細)
Lecture01/	15-May-2017 12:44	-	

# USBドライバのインストール

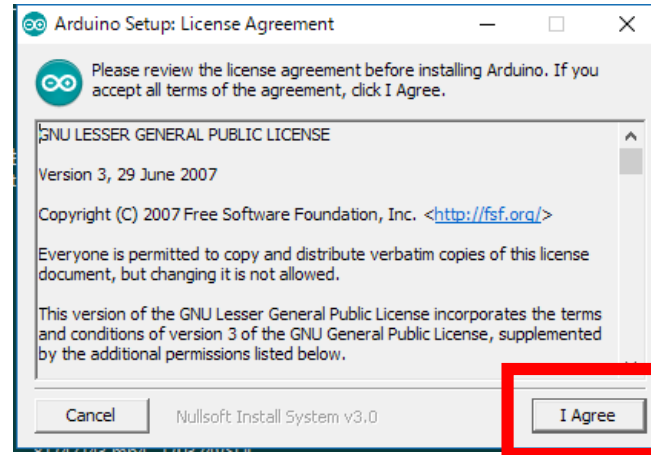


※ USBドライバをインストールする前に、マイコンボードを接続しない

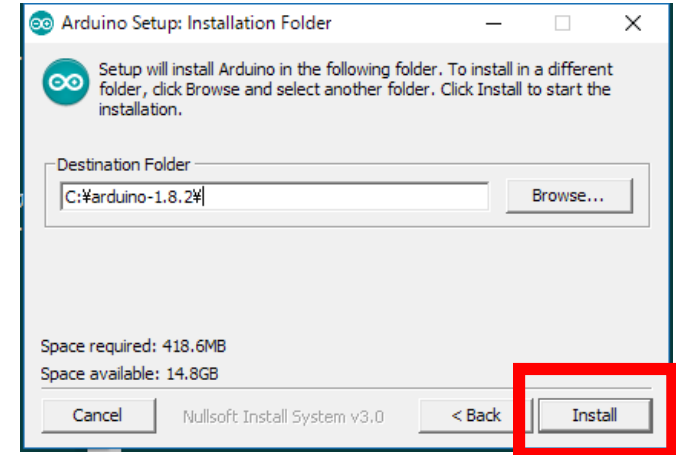
# Arduinoのインストール



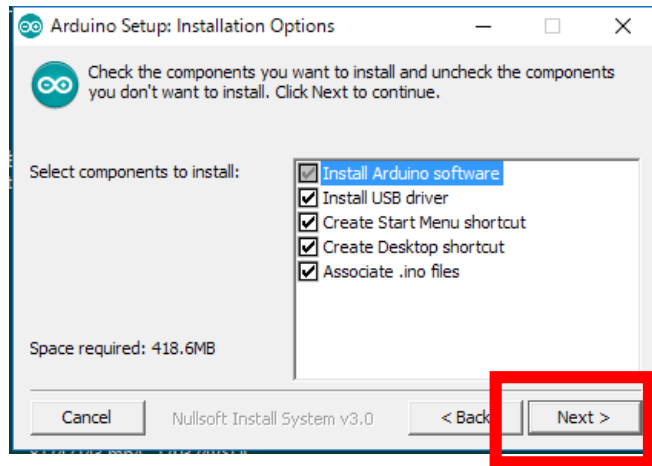
インストーラー  
をダブルクリック



ライセンスの同意 : I Agree



インストール先 : Install

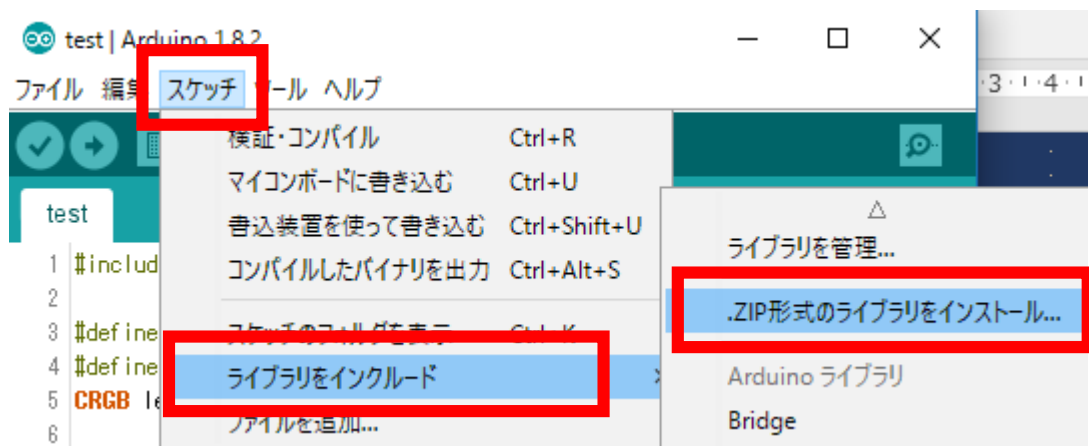


ドライバ等のインストール : Next

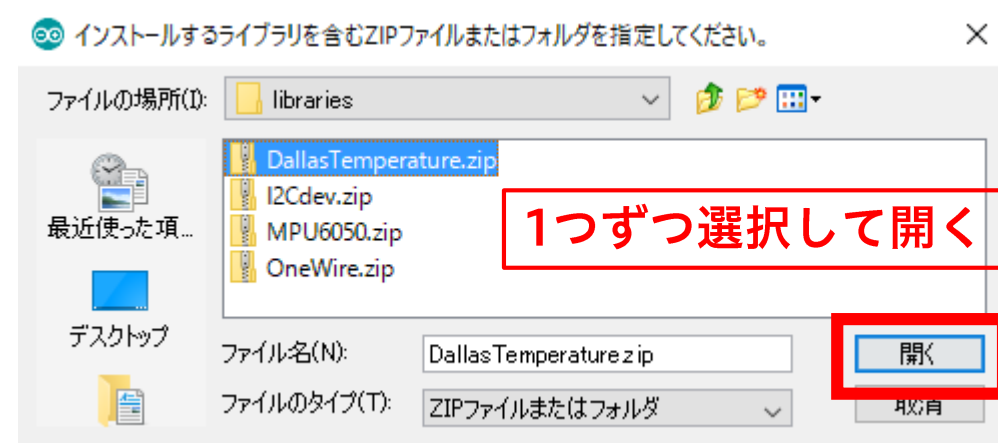


デスクトップにArduinoを起動するための  
アイコンができる

# ライブラリのインストール



手順1：スケッチ→ライブラリをインクルード  
→ZIP形式のライブラリをインストール



手順2：ライブラリのZIPファイルを選択して開く

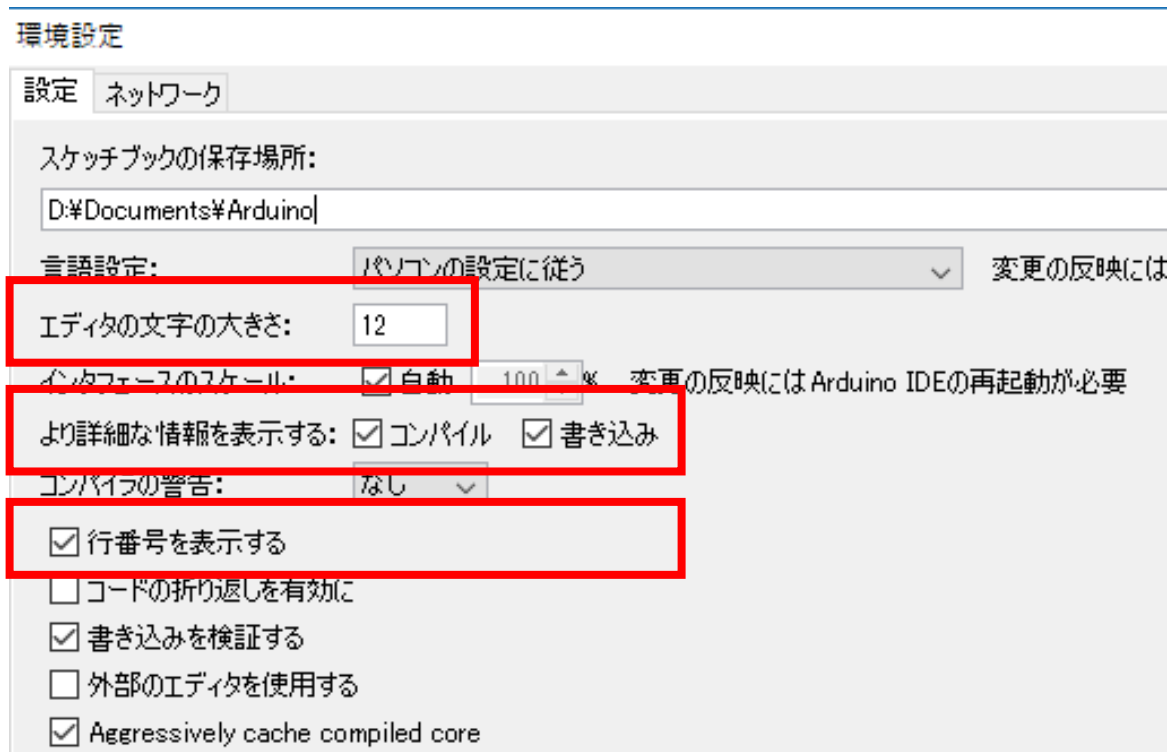
手順1, 手順2の順に繰り返し、すべてのライブラリをインストールする。

DallasTemperature.zip	・・・温度センサ用
I2Cdev.zip	・・・I <sup>2</sup> C通信用
MPU6050.zip	・・・加速度・ジャイロセンサ用
OneWire.zip	・・・1線式通信用

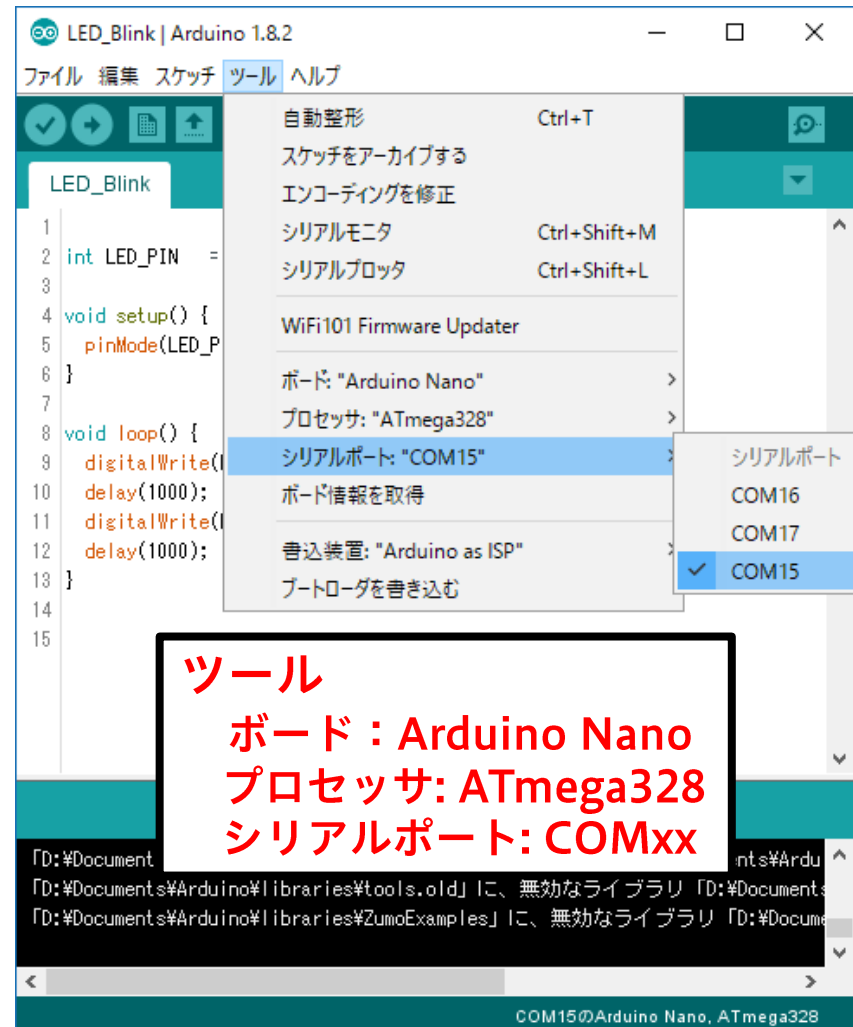
# 環境設定

## ファイル→環境設定

- ① エディタの文字の大きさ：見やすい大きさに変更
- ② コンパイル，書き込みにチェック
- ③ 行番号を表示する。にチェック

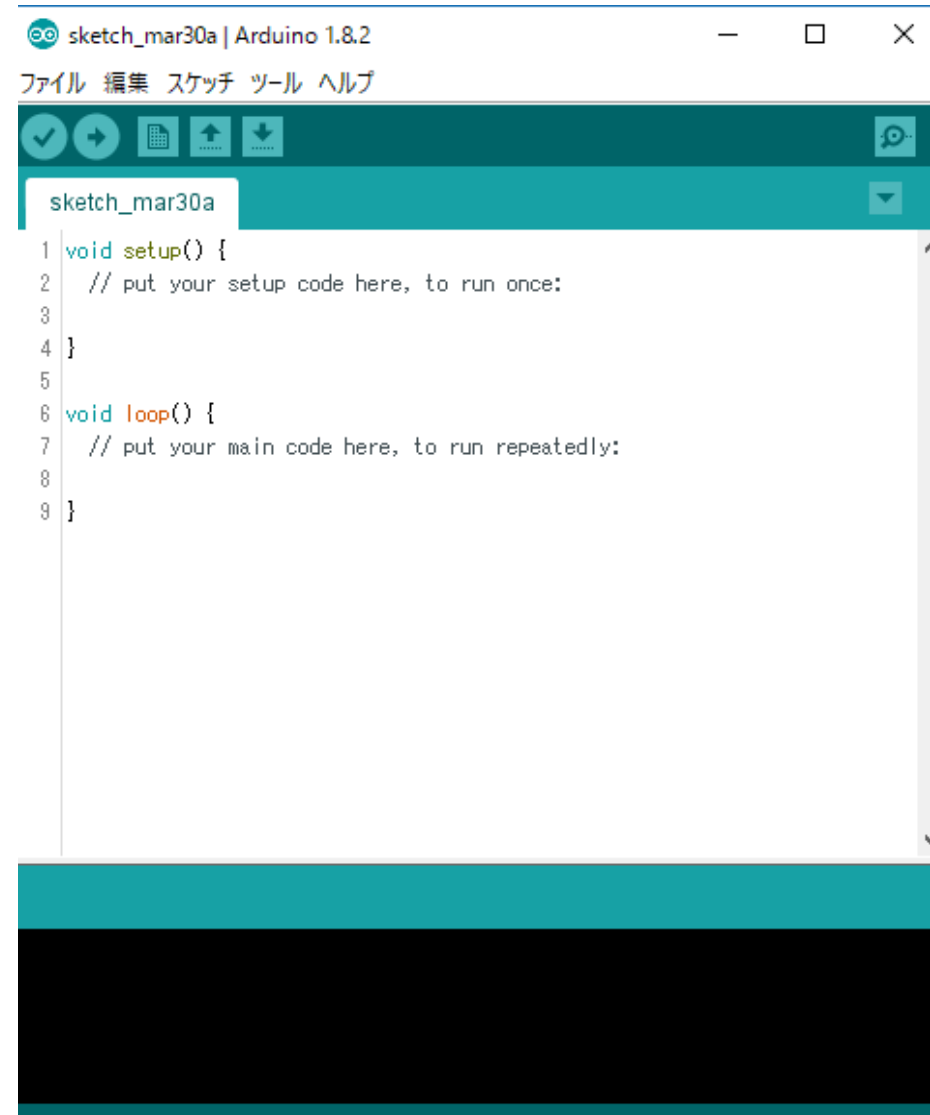


## ④ マイコンボード，プロセッサ，シリアルポートの設定



# Arduino

- ▶ Arduino(あるでゅいーの)
- ▶ 初心者でも簡単にマイコンの開発ができる環境
- ▶ プロトタイプ（試作品）製作
- ▶ 開発コミュニティが活発



# JJ5マイコンボード(Arduino Nano互換)

- ▶ JJ5(Jyosan Jima 5世代目)
- ▶ Arduino Nano互換

D9 : LED(赤色)  
D11 : LED(緑色)

A0 : 温度センサ(LM61CIZ)

A1 : 可変抵抗

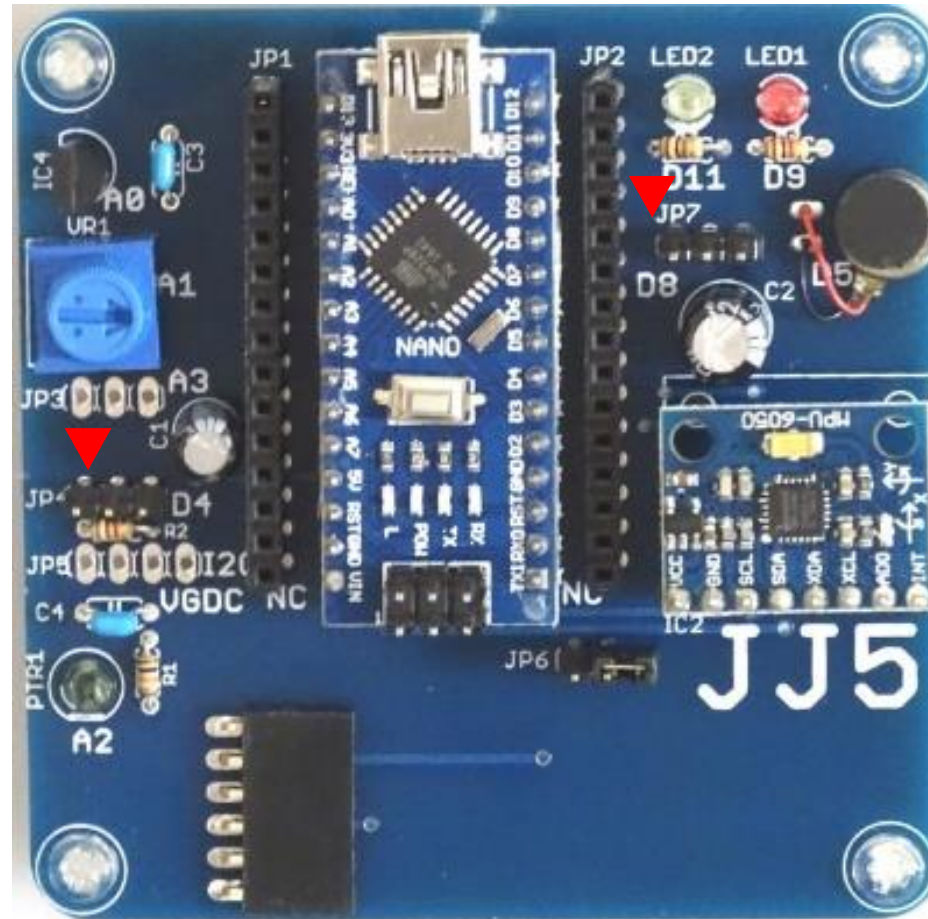
D4 : デジタル温度センサ  
(DS18B20)

A2 : 照度センサ  
(NJL7502L)

D5 : 振動モーター

JP7 : サーボモーター

MPU6050 :  
加速度・ジャイロセンサ





## センサ

- ▶ 温度センサ . . . . . 温度をはかる
- ▶ 可変抵抗 . . . . . 電圧の変化をはかる
- ▶ デジタル温度センサ . . . . . 温度をはかる
- ▶ 照度センサ . . . . . 明るさをはかる
- ▶ 加速度・地磁気センサー . . . . 動きをはかる

## アクチュエーター

- ▶ LED(赤色, 緑色)
- ▶ 振動モーター
- ▶ サーボモーター

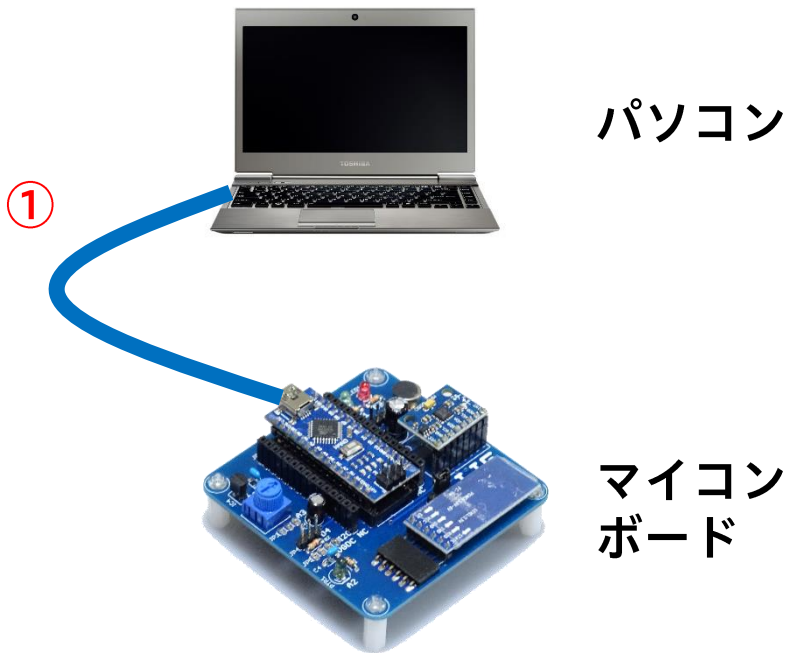
# はじめてのスケッチ

---



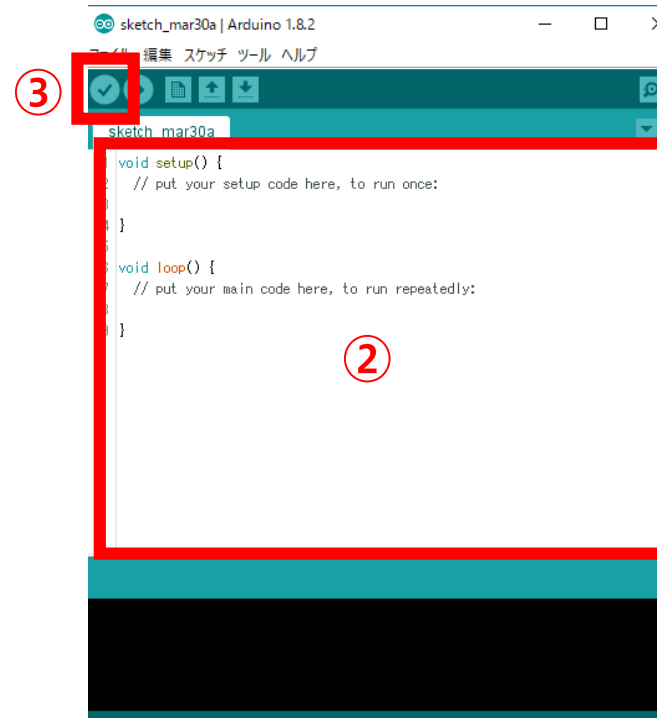
# 開発方法

- ① パソコンとマイコンボードを  
USBケーブルで接続



マイコンボードの電源LEDが点灯  
していることを確認

- ② スケッチ(プログラム)の作成  
Arduino
- ③ スケッチの検証



- ④ プログラムの書き込み



# はじめてのスケッチ：LEDの点滅(1秒ごと)

## ▶ ファイル→スケッチ例→01.Basics→Blinkを開く

```
24 // the setup function runs once when you press reset or power the board
25 void setup() {
26     // initialize digital pin LED_BUILTIN as an output.
27     pinMode(LED_BUILTIN, OUTPUT);
28 }
29
30 // the loop function runs over and over again forever
31 void loop() {
32     digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
33     delay(1000);                        // wait for a second
34     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
35     delay(1000);                        // wait for a second
36 }
```

# LEDの点滅(電子回路)を考える

- ▶ スイッチを押す  
→ LEDが点く
- ▶ スイッチを離す  
→ LEDが消える

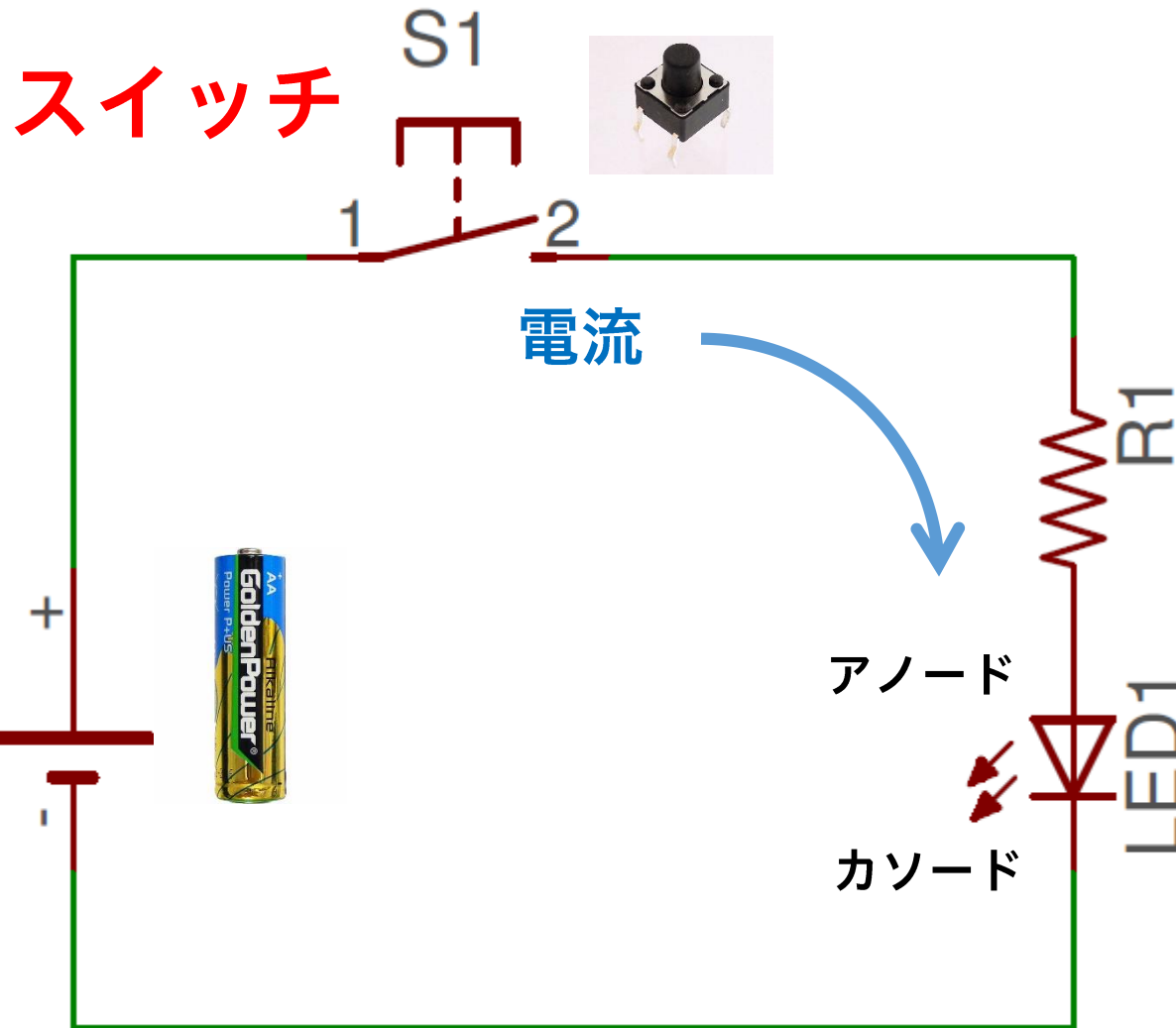
・ LEDの極性



乾電池

5V

GND



抵抗



LED

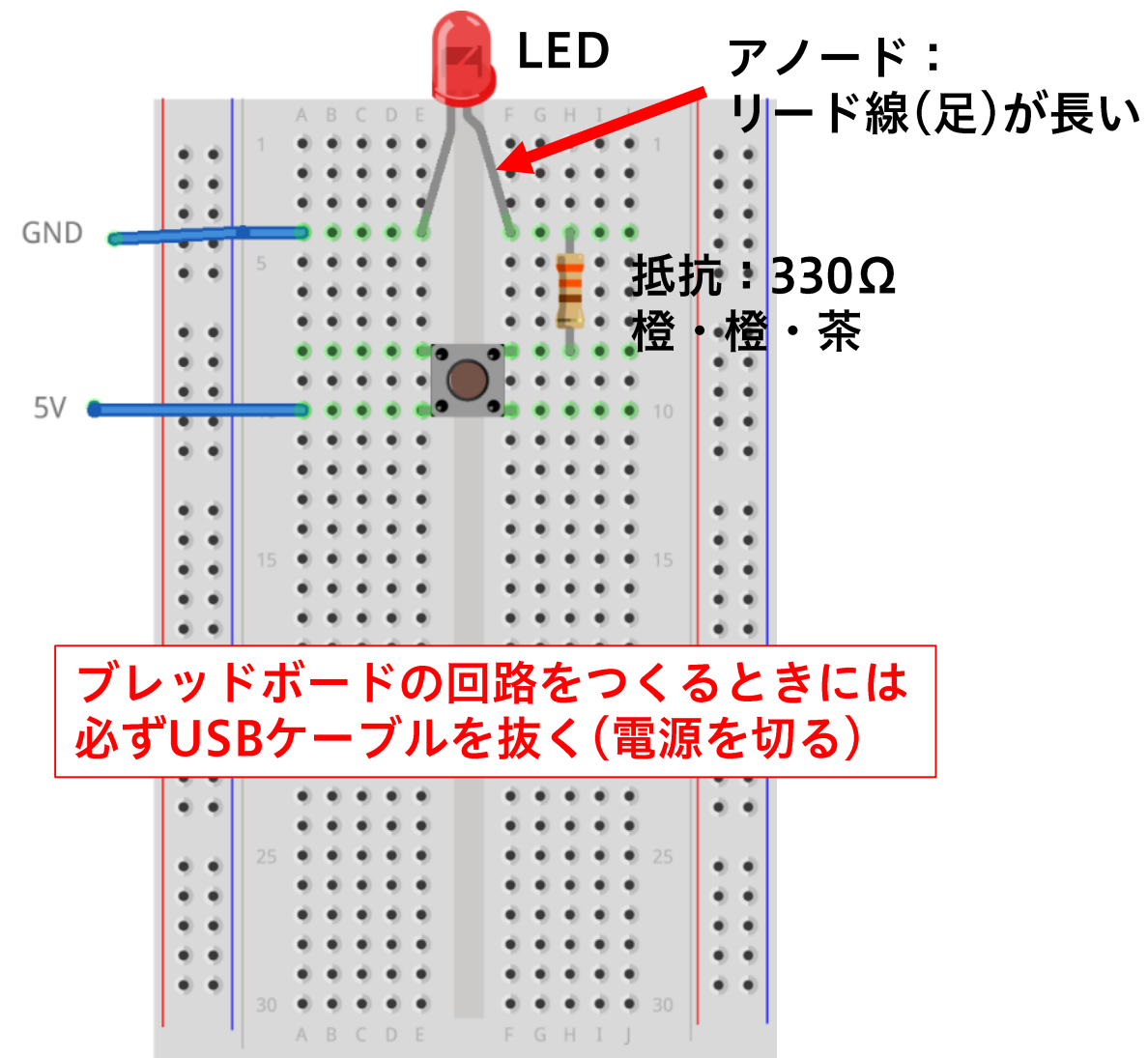


# LEDの点滅(電子回路)を考える

## ▶ 電池の代わり (5V, GND)

JP1

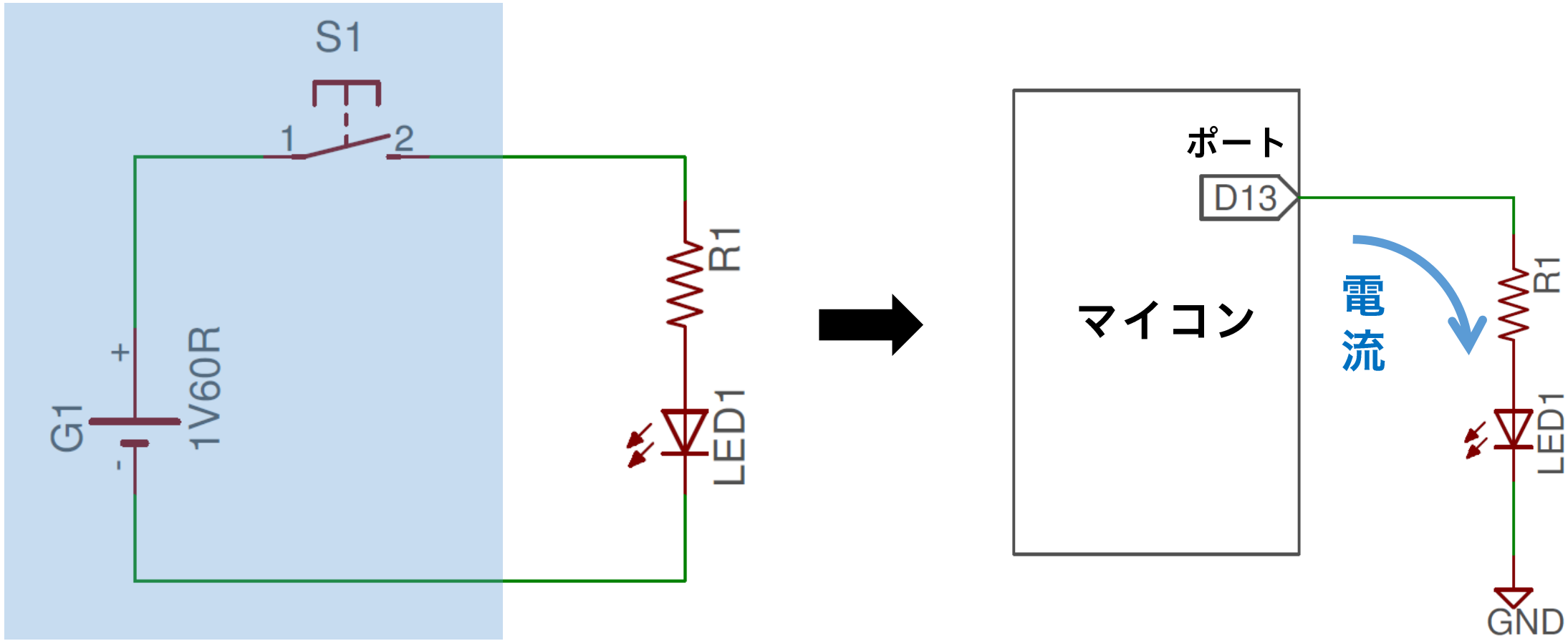
D13  
3V3  
REF  
A0  
A1  
A2  
A3  
A4  
A5  
A6  
A7  
5V  
RST  
GND  
VIN



fritzing

# LEDの点滅(マイコン)では？

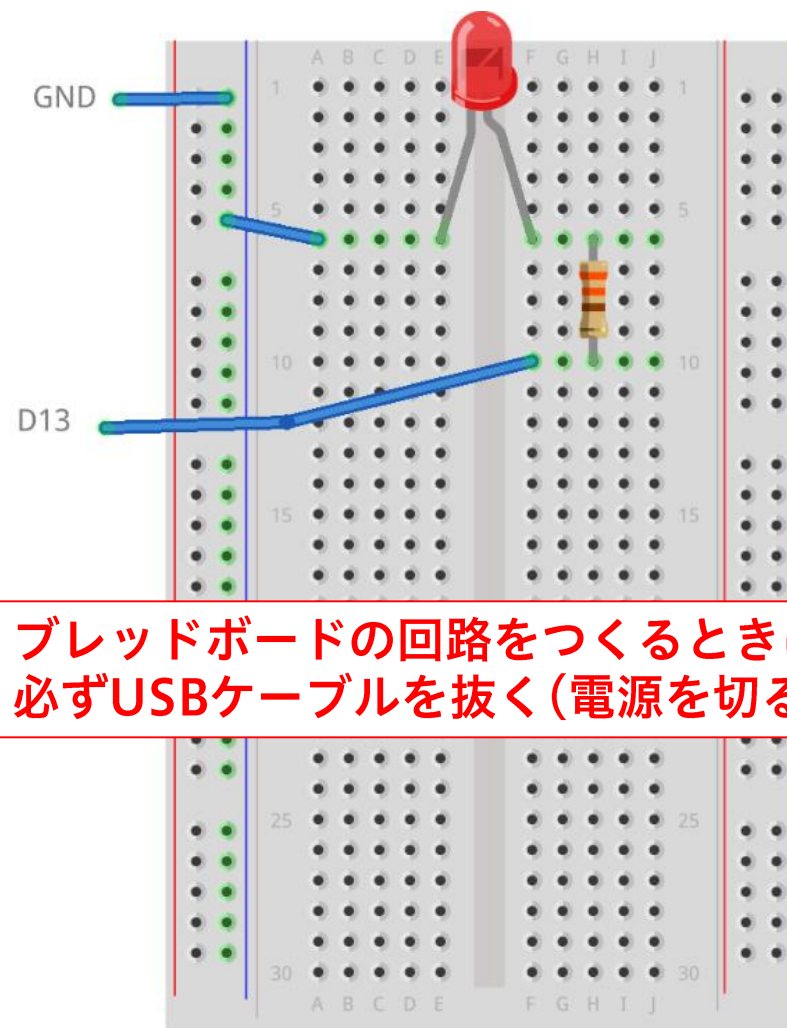
- ▶ LEDの点滅 → 電子回路からマイコンへ



# LEDの点滅(マイコン)では？

JP1

D13  
3V3  
REF  
A0  
A1  
A2  
A3  
A4  
A5  
A6  
A7  
5V  
RST  
GND  
VIN



ブレッドボードの回路をつくる時には  
必ずUSBケーブルを抜く(電源を切る)



# Exapmle101: LEDの点滅(1秒ごと)

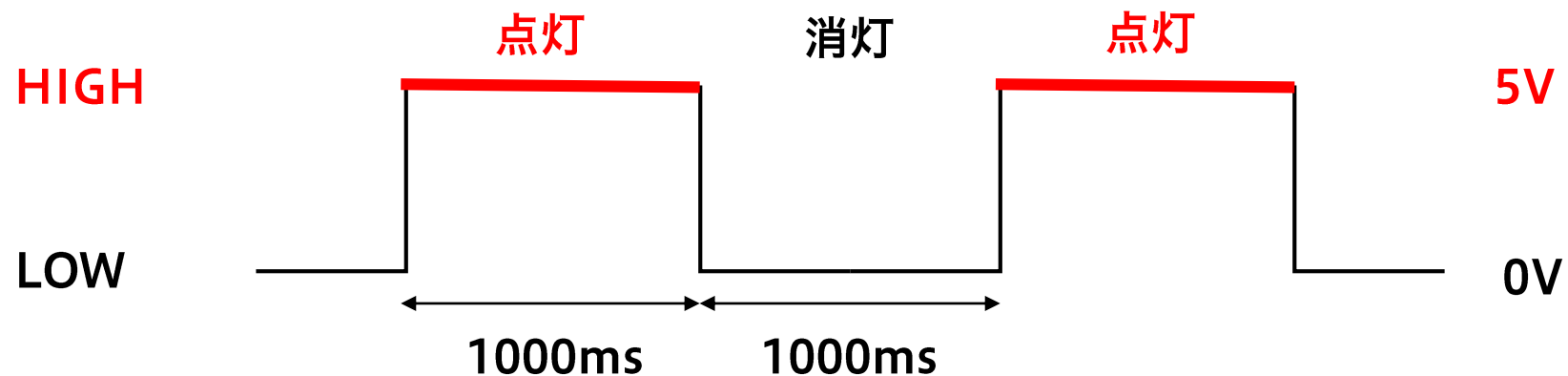
```
// Example101
const int LED_PIN = 13;

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

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

スケッチの各関数の意味

- ▶ `setup()`・・・マイコンの初期設定(一度だけ実行)
- ▶ `loop()`・・・無限ループ(繰り返し実行)
- ▶ `pinMode`・・・ポートの入出力設定
- ▶ `digitalWrite`・・・ポートのHIGH,LOW出力
- ▶ `delay`・・・ms単位の遅延(待ち)



# Exapmle101: LEDの点滅(1秒ごと)

- ▶ スケッチにコメントを入れる。
  - コメントの行はプログラムと解釈されない。

```
// Example101
/* LEDを1秒ごとに点滅させるスケッチ */

// ピンの設定
const int LED_PIN = 13 // D13ピンにLEDを接続

void setup() {
  pinMode(LED_PIN, OUTPUT); // D13ピンを出力に設定
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // D13ピンをHIGHにしてLEDを点ける
  delay(1000);                 // 1000 ms 待つ
  digitalWrite(LED_PIN, LOW);  // D13ピンをLOWにしてLEDを消す
  delay(1000);                 // 1000 ms 待つ
}
```

- ▶ コメント(注釈)
  - 行の先頭が"//"ではじまる。  
例) // これはコメントです。
  - コメントが"/\*"と"\*/"で囲まれている。  
例) /\* これはコメントです。 \*/  
例) /\* これは  
コメントです。  
\*/

## Example102. デジタルポートD9のLED(赤)を点滅

- ・ ヒント : Example101のLED\_PINを書き換える

## Example103. デジタルポートD11のLED(緑)を点滅

- ・ ヒント : Example101のLED\_PINを書き換える

## Example104. D9のLEDの点灯, 消灯の時間を500ミリ秒に変更

- ・ ヒント : delay()を書き換える

## Example105. D9のLEDの点灯, 消灯の時間を10ミリ秒に変更

このときLEDはどのように見えるか？

- ・ ヒント : delay()を書き換える

## Example106. D9のLEDを点灯500ミリ秒, 消灯500ミリ秒を3回繰り返す,

その後1秒消灯, これを繰り返す

forループについて調べる

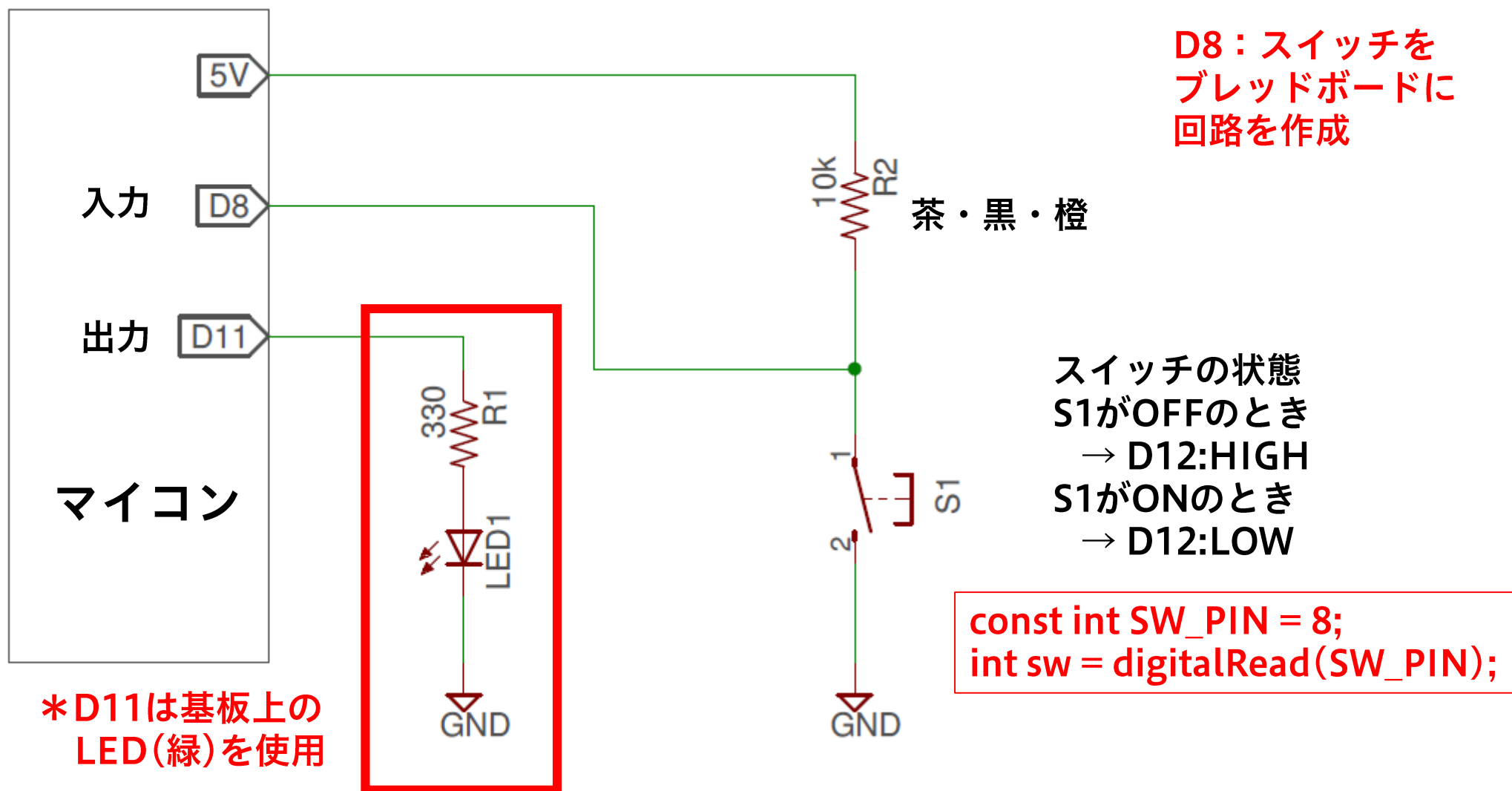
- ・ ヒント : for (int i = 0; i < 3; i++) {}

## Example107. LEDの点滅(電子回路)と同じ動作(次のページ)

if文について調べる

- ・ ヒント : スイッチの状態を調べる
- ・ `int sw = digitalRead(SW_PIN); if (sw == LOW) { LED点灯 } else { LED消灯 }`

# ヒント : Example107

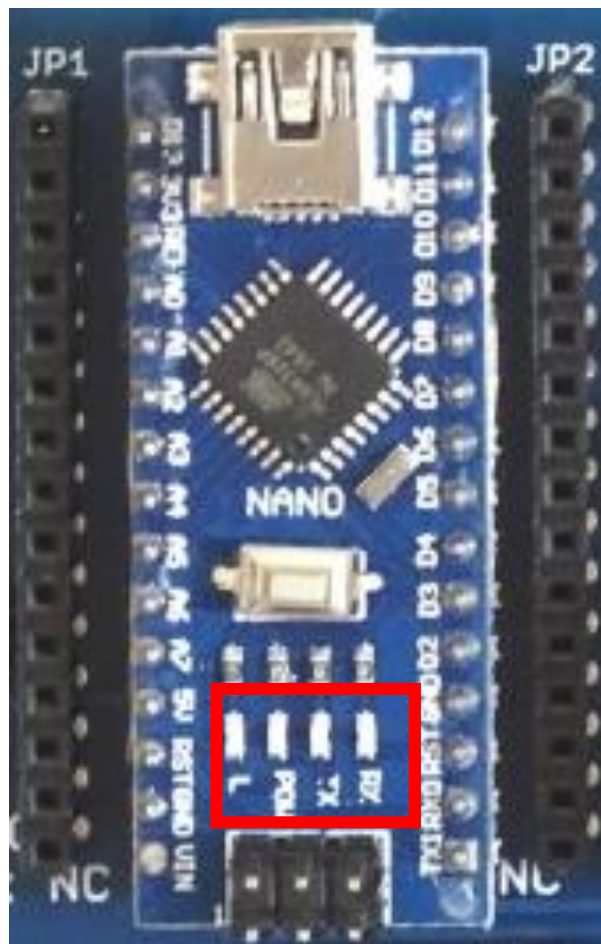


# 付録：Arduino Nano のピン配置

JP1

D13
3V3
REF
A0
A1
A2
A3
A4
A5
A6
A7
5V
RST
GND
VIN

Arduino Nano



JP2

D12
D11~
D10~
D9~
D8
D7
D6~
D5~
D4
D3~
D2
GND
RST
RXD
TXD



D13 電源 TX RX

	デジタルポート
	アナログポート
	シリアルポート
	電源系