

AI/IoTセンサのしくみを知ろう(応用編)



徳島大学技術支援部常三島技術部門
技術専門職員 辻 明典 博士(工学)
E-mail: a-tsuji@is.tokushima-u.ac.jp

講座内容

▶ 講師：辻 明典（徳島大学技術支援部）

桑折 範彦（徳島大学名誉教授）

川上 博（徳島大学名誉教授）

▶ 土曜日：10:00～11:30

▶ 日程：

① 10 / 5 概要，環境設定，配布部品の確認

② 10 / 19 復習

③ 10 / 26 ロボットのモーター1（基本動作）

④ 11 / 9 ロボットのモーター2（応用動作）

⑤ 11 / 16 ロボットのセンサ1
（距離センサと モーター）

⑥ 11 / 30 ロボットのセンサ2
（フトリフレクタ）

⑦ 12 / 7 ロボットの制御1
（モーター，センサの協調動作）

⑧ 12 / 14 ロボットの制御2
（ライントレース）

⑨ 12 / 21 ロボットの制御3
（ライントレース）

本日の予定

- ▶ Arduinoのスケッチ
- ▶ LEDの点滅
- ▶ フルカラーLED
- ▶ スイッチとLED
- ▶ スイッチ(無線)とLED

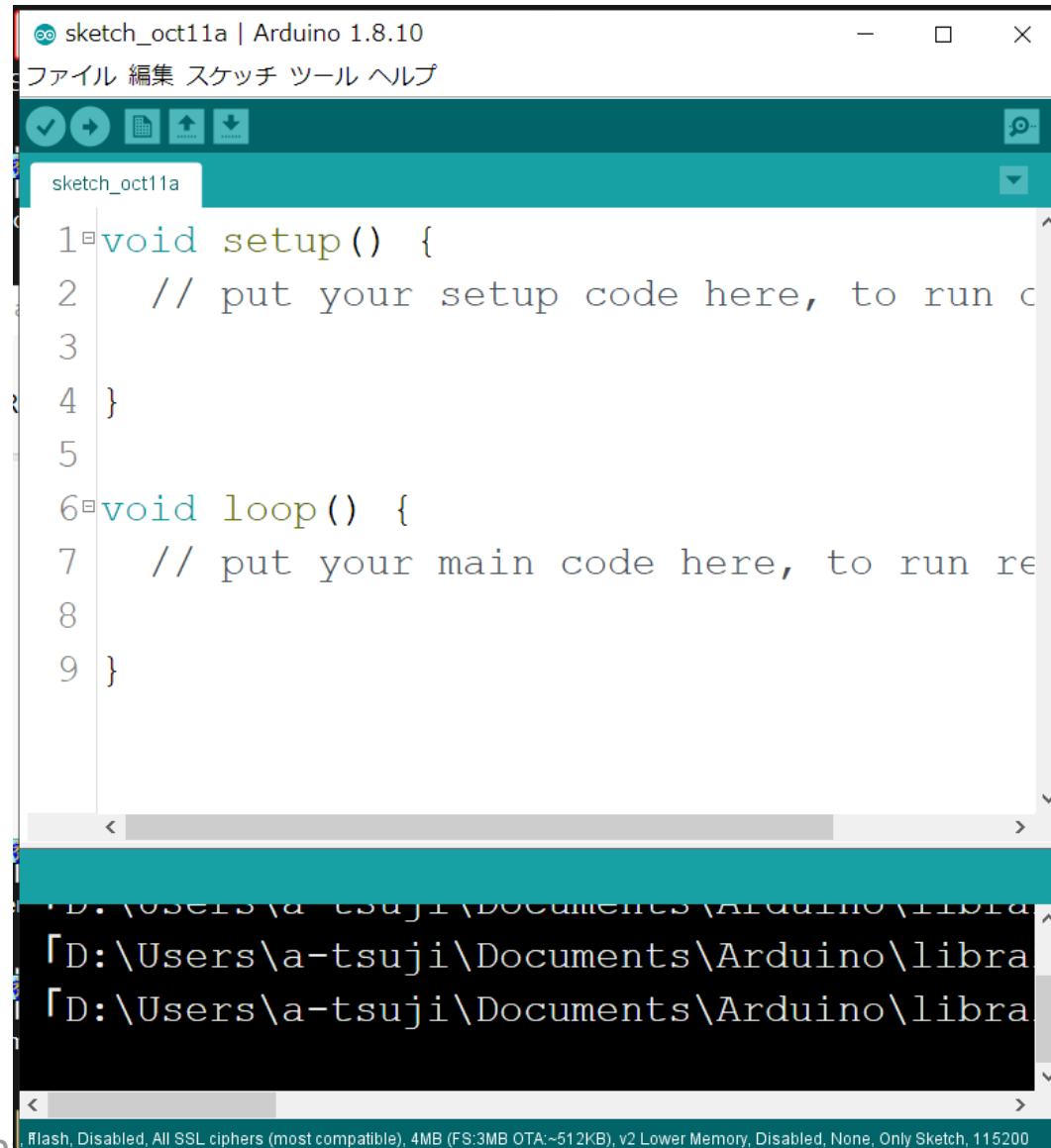
Arduinoのスケッチ

▶ setup()

- マイコンの初期設定

▶ loop()

- 無限ループ
- while(1)
- マイコンのタスク実行

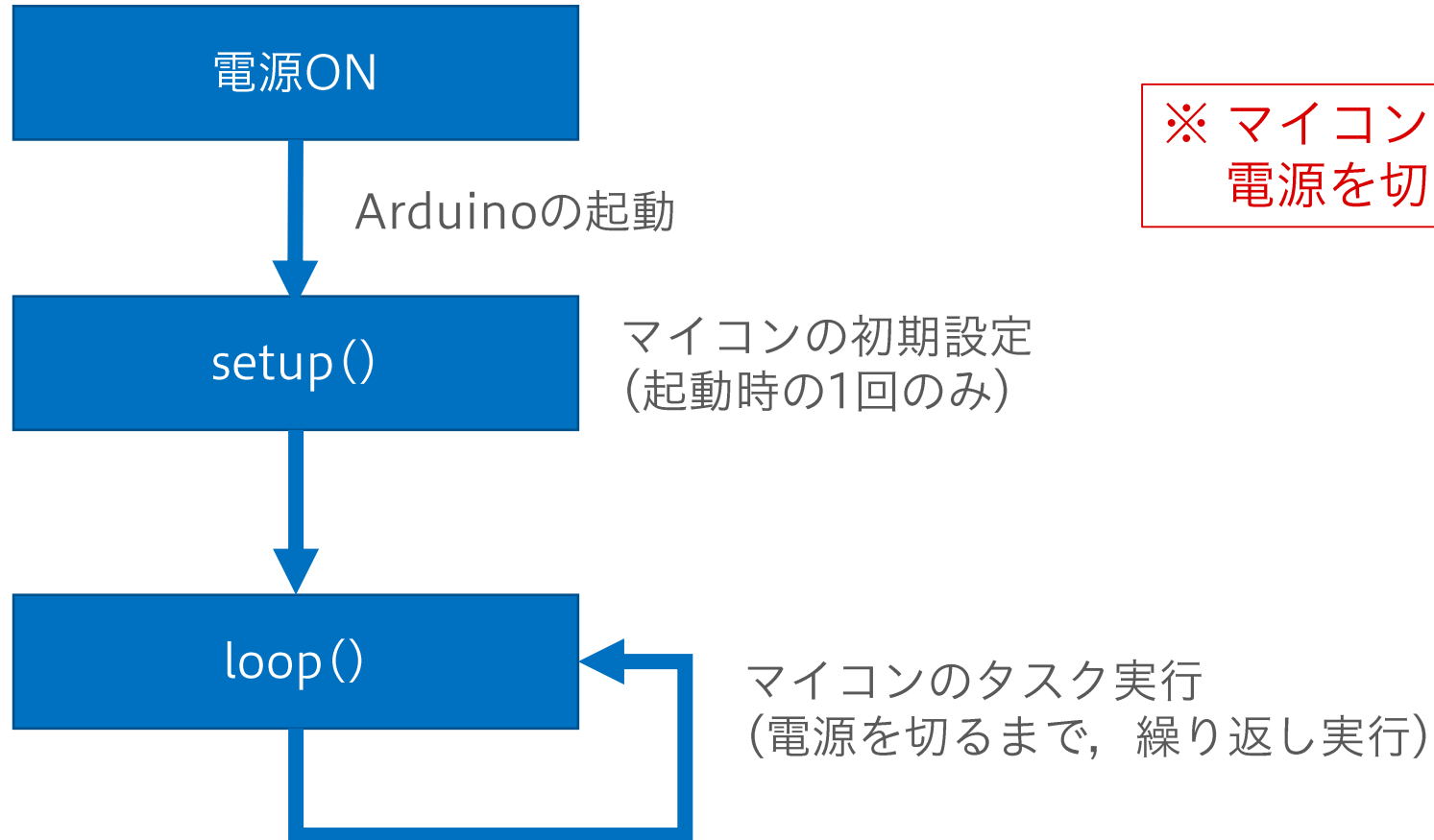


```
sketch_oct11a | Arduino 1.8.10
ファイル 編集 スケッチ ツール ヘルプ
sketch_oct11a
1 void setup() {
2   // put your setup code here, to run c
3
4 }
5
6 void loop() {
7   // put your main code here, to run re
8
9 }
```

D:\Users\a-tsuji\Documents\Arduino\libra
D:\Users\a-tsuji\Documents\Arduino\libra
Flash, Disabled, All SSL ciphers (most compatible), 4MB (FS:3MB OTA:~512KB), v2 Lower Memory, Disabled, None, Only Sketch, 115200

プログラムの実行

- ▶ フローチャート・・・プログラムの流れを説明した図



※ マイコンに書き込んだプログラムは電源を切っても記憶されています。

LEDの点滅

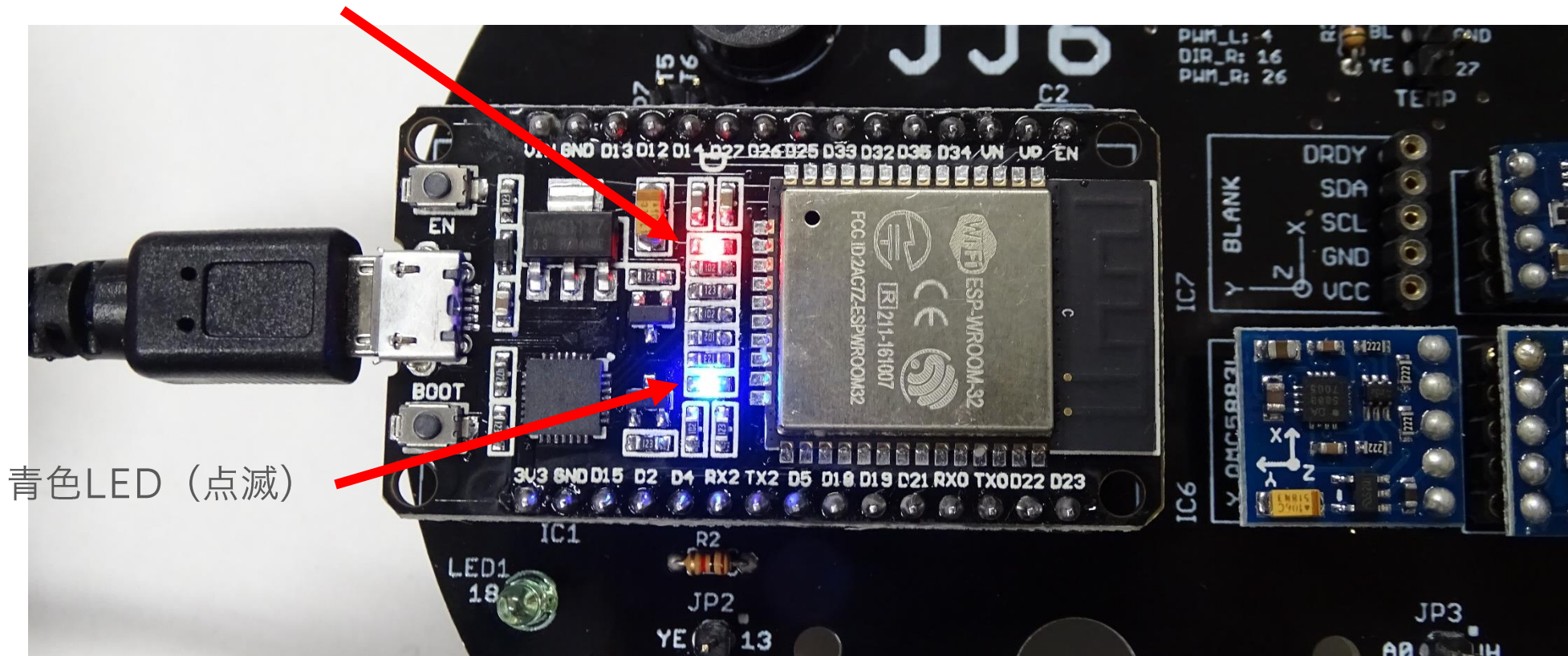
- ▶ メニュー：ファイル→スケッチ例→01.Basics→Blinkを開く

```
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

- ▶ LED(青)が1秒毎に点灯，消灯を繰り返す

赤色LED (電源：点灯)



青色LED (点滅)

Ex0201 | LEDの点滅

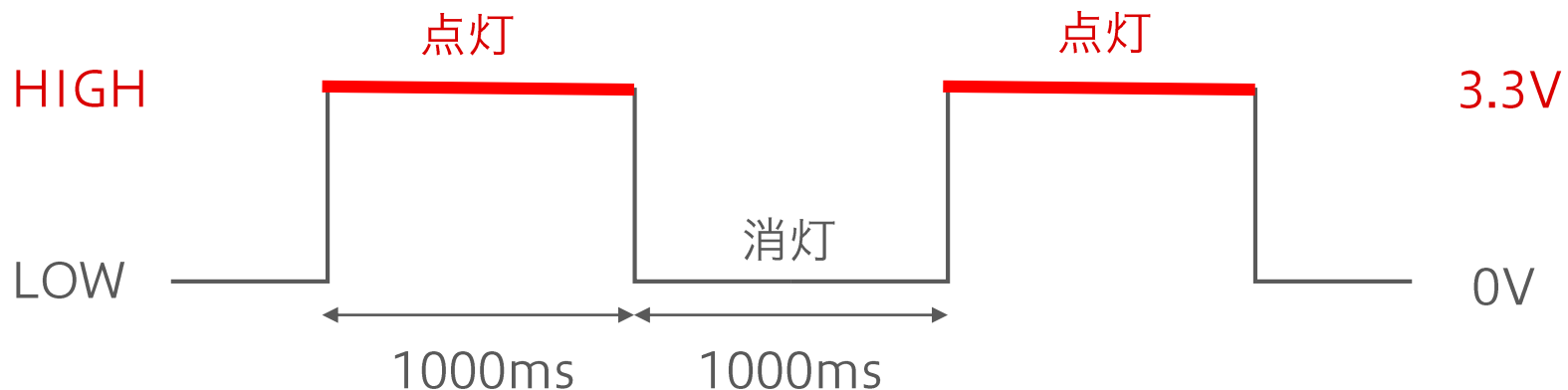
```
// Ex0201 LEDの点滅(1秒毎)
const int LED_PIN = 2;

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

void loop() {
  digitalWrite(LED_PIN, HIGH); // 点灯
  delay(1000); // 1000ms待ち
  digitalWrite(LED_PIN, LOW); // 消灯
  delay(1000); // 1000ms待ち
}
```

スケッチで使われる関数

- ▶ 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 ミリ秒単位の遅延(待ち)
 - 1000ms = 1s



LEDのポート

- ▶ 基板上の緑色LEDを点滅
- ▶ J7ポート一覧を参照
 - ポート番号確認

ポート	接続先	型番	ライブラリ	注釈
2	青色 LED(ボード上)	表面実装	pinMode, digitalWrite	
18	緑色 LED	OST1313G	pinMode, digitalWrite	
25	スイッチ	タクト SW	pinMode, digitalRead	

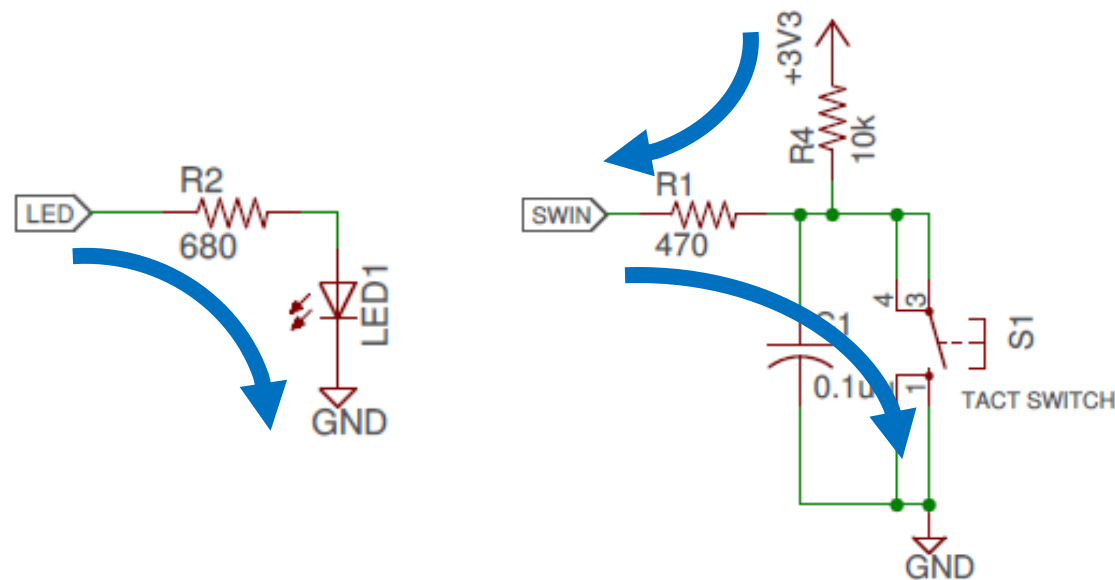
Ex0202 | スイッチとLED

```
// Ex0202 スイッチとLED
const int LED_PIN = 2;
const int SW_PIN = 25;

void setup() {
  pinMode(LED_PIN, OUTPUT); // 出力
  pinMode(SW_PIN, INPUT); // 入力
}

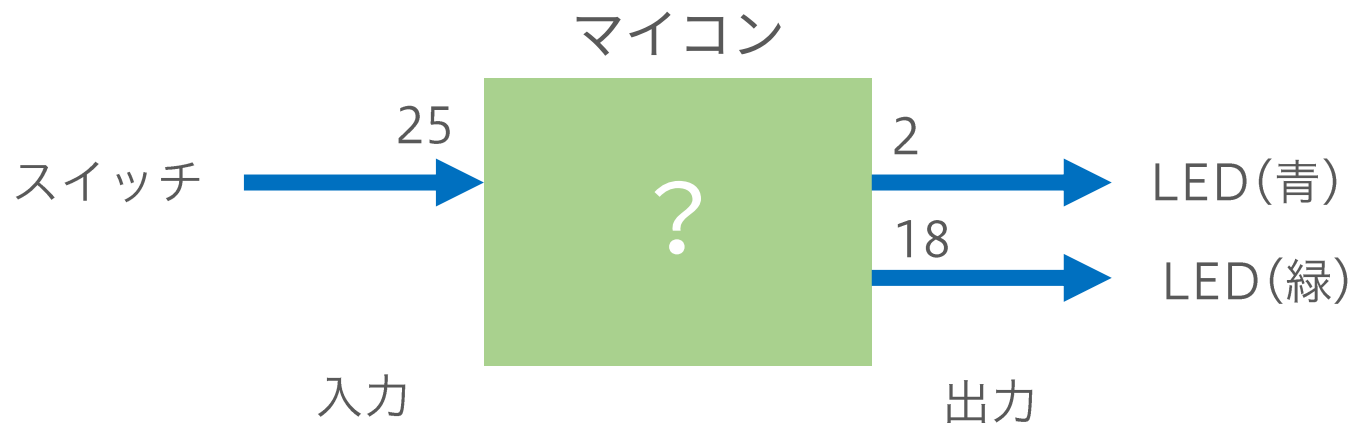
void loop() {
  int sw = digitalRead(SW_PIN);
  if (sw == LOW) {
    digitalWrite(LED_PIN, HIGH); // 点灯
  } else {
    digitalWrite(LED_PIN, LOW); // 消灯
  }
}
```

回路図



まとめ

- ▶ マイコンとLED, スイッチ, センサなどの電氣的な配線
 - マイコンのポートと接続
 - 入力と出力
- ▶ ポート毎の役割
 - デジタルIO, アナログ入力, PWMなど
- ▶ プログラムの役割
 - 接続先のデバイスを制御



リングLED

▶ リングLED

- フルカラーLED 16個
- 1チップに赤, 緑, 青のLEDを内蔵

▶ LEDの色・明るさ設定

- RGB (red, green, blue);
- HSV (hue, saturation, brightness);

▶ LEDの表示

▶ ポート



19

リング LED

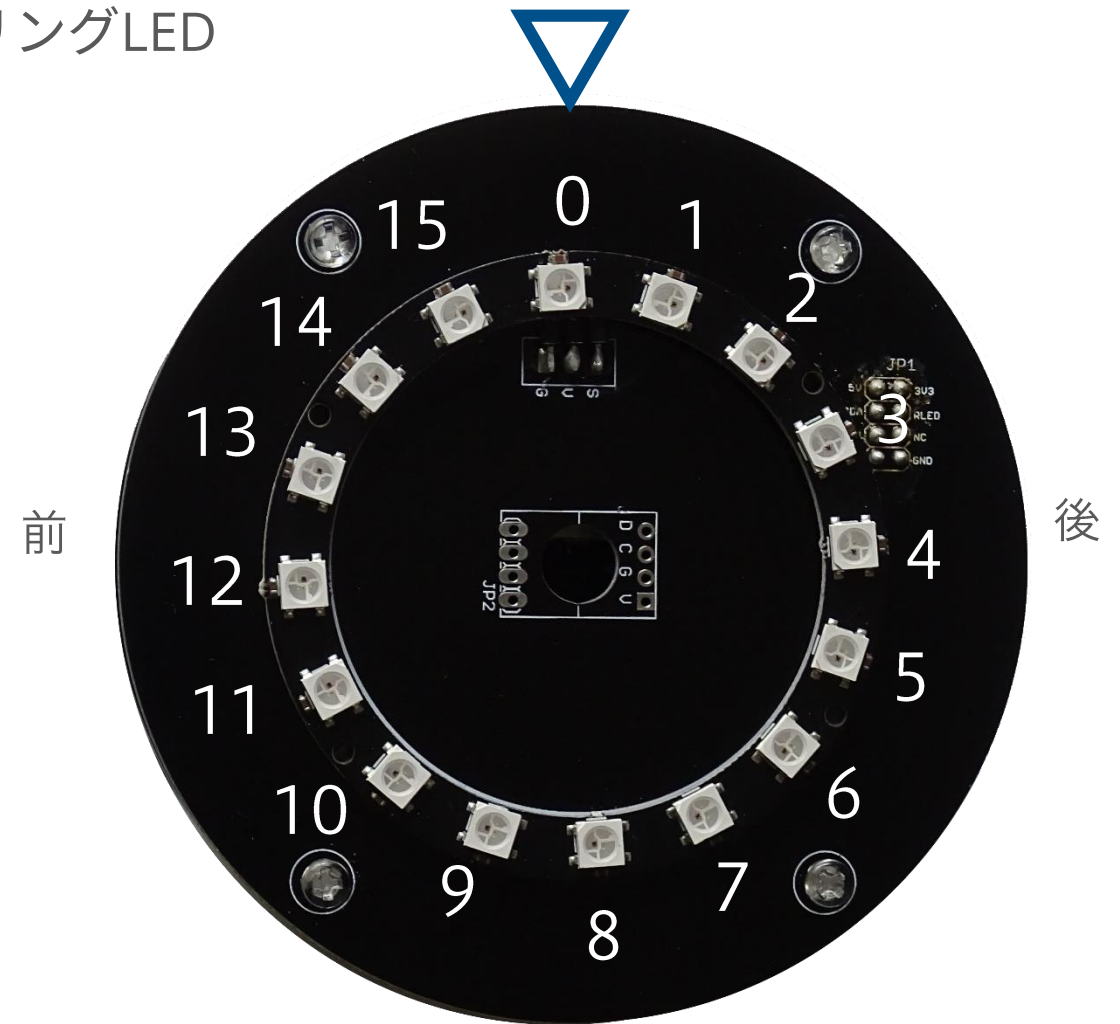
WS2812B

FastLED

リングLED

- ▶ フルカラーLED 16個
 - 1個ずつLEDの色を変えられる
- ▶ LED用配列
 - `leds[n]`, $n=0,1,2,\dots,15$
- ▶ RGB
 - `leds[0] = CRGB(255, 0, 0); //赤`
 - `leds[1] = CRGB(0, 255, 0); //緑`
 - `leds[2] = CRGB(0,0,255); //青`
 - . . .
- ▶ HSV
 - `leds[0] = CHSV(0, 255, 255);`
 - `leds[1] = CHSV(64, 255, 255);`
 - `leds[2] = CHSV(128, 255, 255);`
 - . . .

リングLED



Ex0203 | リングLED

- ▶ FastLEDライブラリを使用 <https://github.com/FastLED/FastLED/wiki/Pixel-reference>
 - 初期設定(変更しない)

```
// Ex0203 リングLED
#include <FastLED.h> // ライブラリの読み込み

// リングLED 16個
const int DATA_PIN = 19; // ピン番号
const int NUM_LEDS = 16; // 16個
CRGB leds[NUM_LEDS]; // LEDの配列

void setup() {
  // リングLEDの設定
  FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS);
  FastLED.setBrightness(32); // 最大輝度 : 0-255
}
```

Ex0203 | リングLED

▶ LEDの点滅

- 色をセット→色の表示→表示する時間
- CRGBを変更して色を変える

```
void loop() {  
  // 赤色で点灯  
  for (int i = 0; i < NUM_LEDS; i++) {  
    leds[i] = CRGB(255, 0, 0);  
  }  
  FastLED.show(); // 表示を更新  
  delay(1000); // 表示時間  
  
  // 消灯  
  for (int i = 0; i < NUM_LEDS; i++) {  
    leds[i] = CRGB(0, 0, 0);  
  }  
  FastLED.show(); // 表示を更新  
  delay(1000); // 表示時間  
}
```

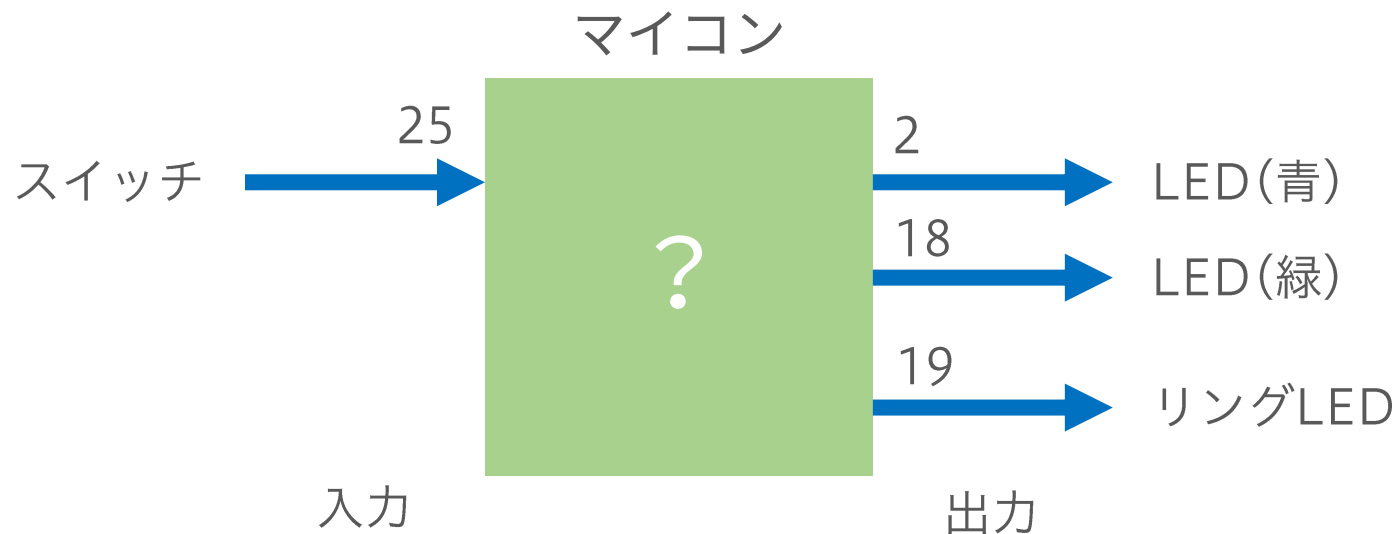


LED(すべて)をまとめて光らせる関数 fill_solid使用

```
void loop() {  
  // 赤色で点灯  
  fill_solid(leds, NUM_LEDS, CRGB(255, 0, 0));  
  FastLED.show(); // 表示を更新  
  delay(1000); // 表示時間  
  
  // 消灯  
  fill_solid(leds, NUM_LEDS, CRGB(0, 0, 0));  
  FastLED.show(); // 表示を更新  
  delay(1000); // 表示時間  
}
```

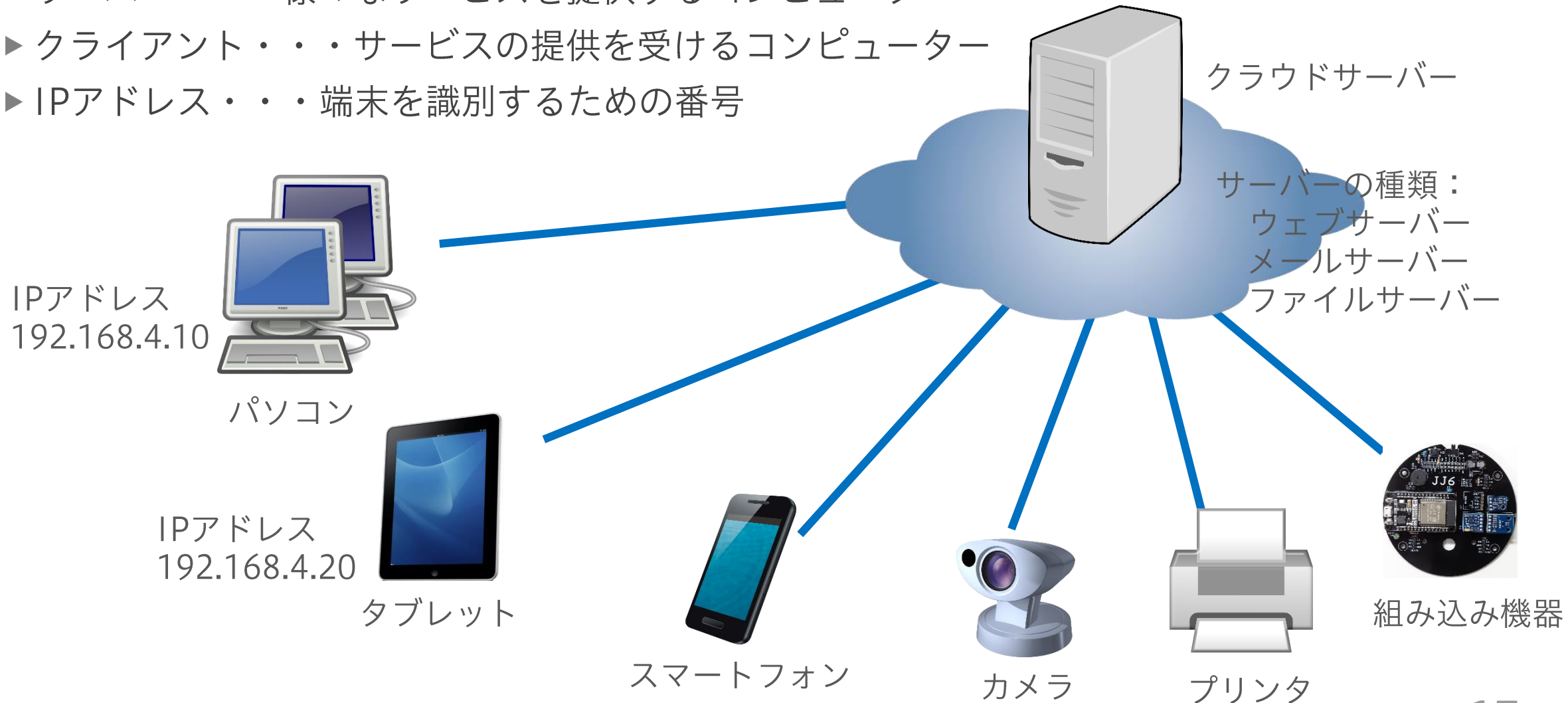
Ex0204 | スイッチとリングLED

- ▶ スイッチが押されたとき，リングLEDを点灯
- ▶ スイッチが押されていないとき，リングLED消灯



ネットワーク

- ▶ サーバー・・・様々なサービスを提供するコンピューター
- ▶ クライアント・・・サービスの提供を受けるコンピューター
- ▶ IPアドレス・・・端末を識別するための番号



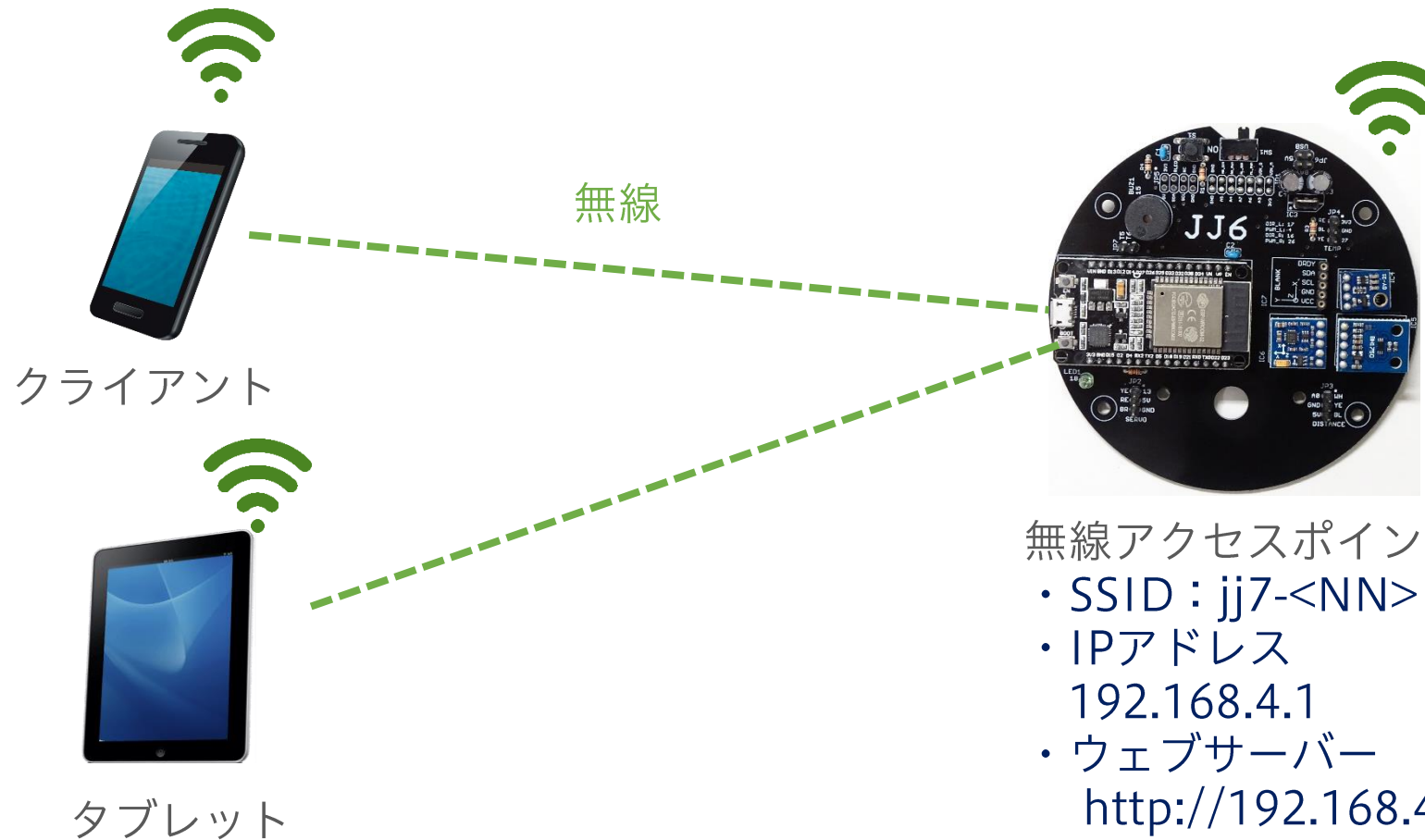
無線Wi-Fi

- ▶ 無線Wi-Fi・・・パソコンやタブレット等，ネットワーク機器を無線の電波でインターネットに接続できる



無線Wi-Fi | 講座の方式

- ▶ インターネット接続なし



無線Wi-Fi | 設定手順

1. 無線APのSSID(無線APの識別子), パスワードの設定

```
// WiFiの設定
```

```
const char *ssid = "jj7-"; // 各自のSSIDを入力  
const char *password = "12345678";
```

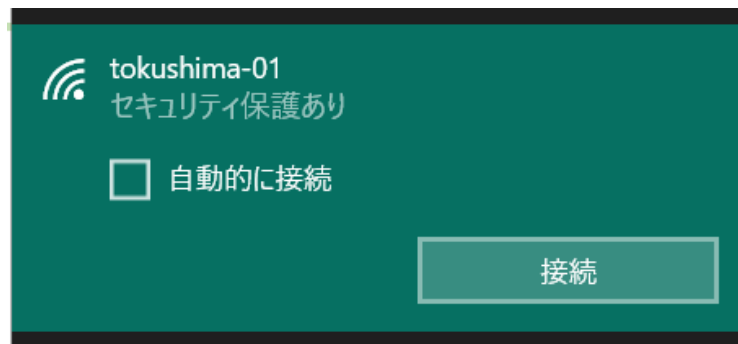
```
const IPAddress ip(192, 168, 4, 1);
```

```
const IPAddress subnet(255, 255, 255, 0);
```

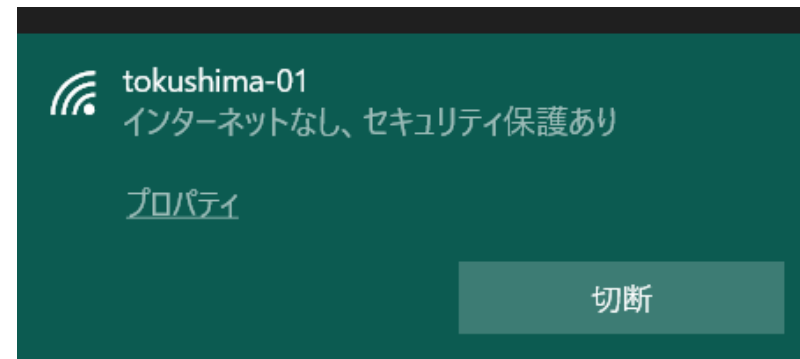
2. スケッチの書き込み

※ ツール→ESP32 Sketch Data Uploadを実行(ESPUIライブラリを使用する最初の1回のみ)

3. クライアントから無線APに接続



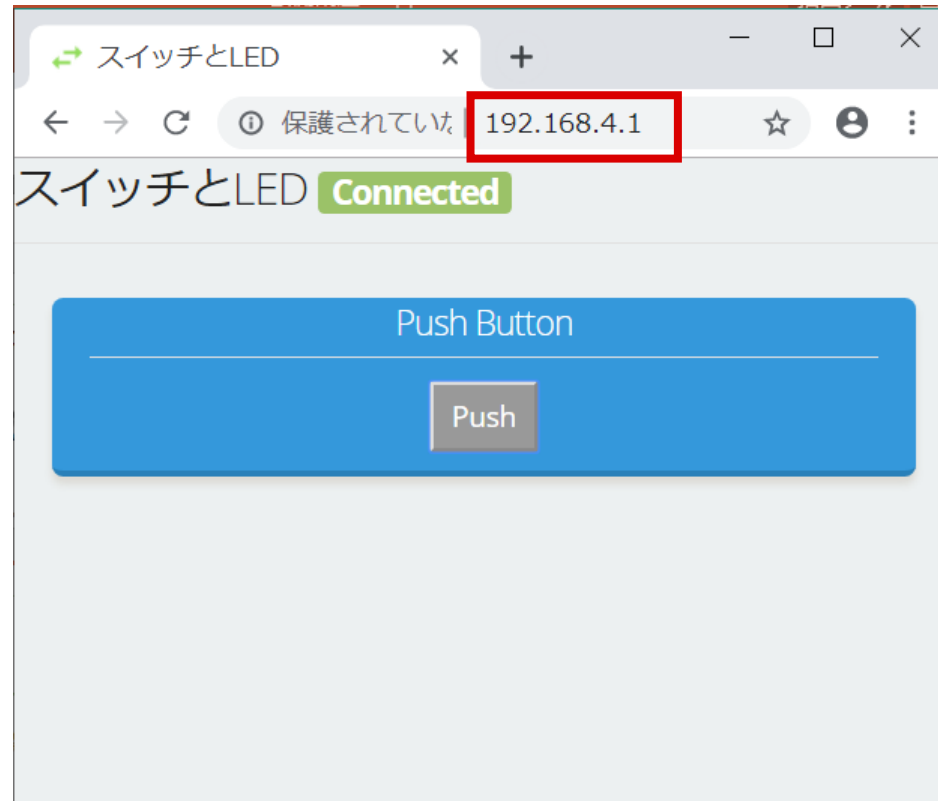
パスワード入力



無線Wi-Fi | 設定手順(つづき)

4. ブラウザの起動 (推奨 : Chrome)
URL: 192.168.4.1

https://www.google.com/intl/ja_jp/chrome/



<http://192.168.4.1/>

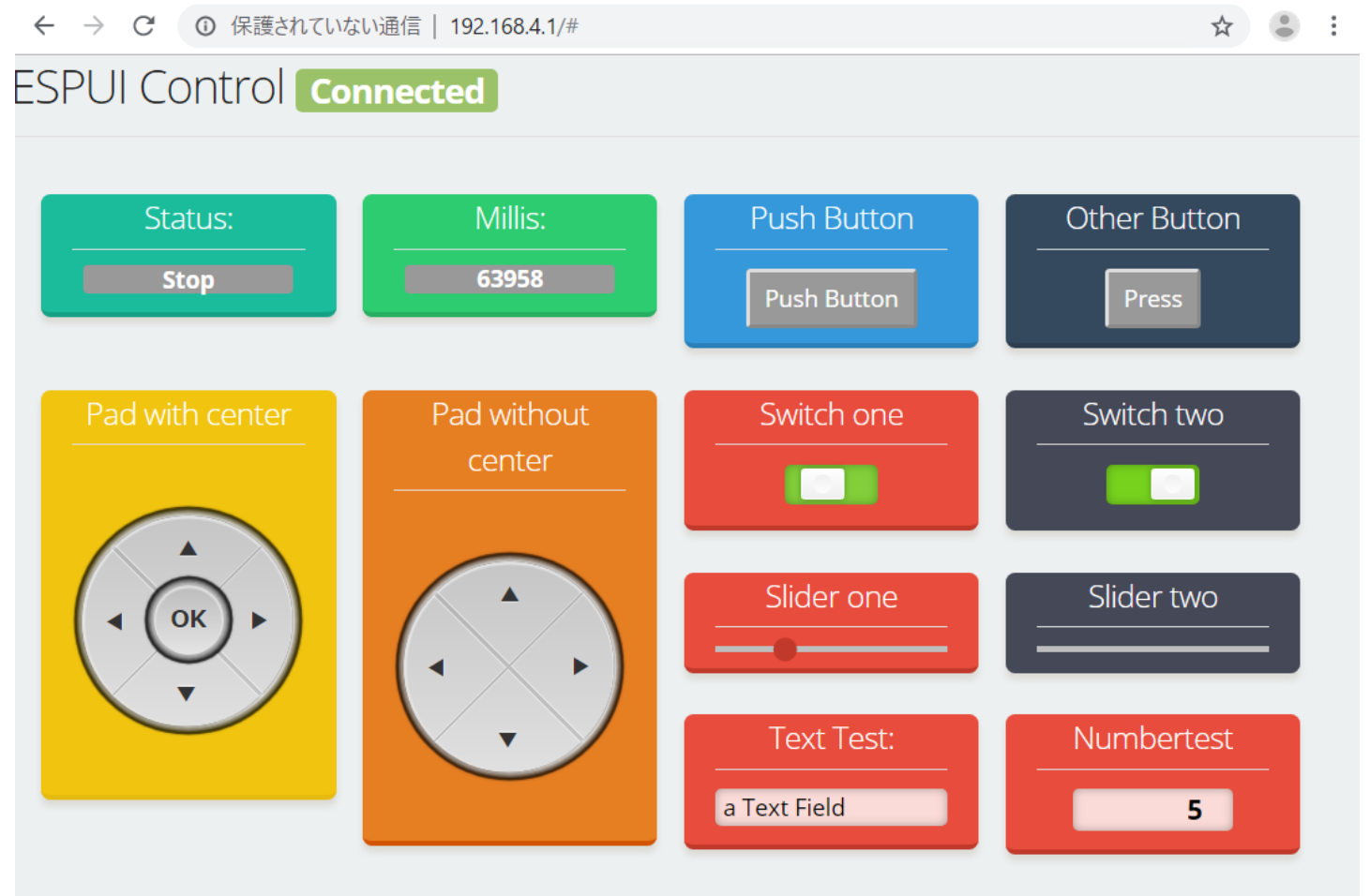
画面の作成

- ▶ ESPUIライブラリ・・・ウェブ上のユーザインタフェース(UI)作成

種類：

- ▶ ラベル
- ▶ ボタン
- ▶ スイッチ
- ▶ コントロールパッド
- ▶ コントロールパッド
(中央ボタン付き)
- ▶ スライダー

- ▶ 背景の色や文字を変更できる。
- ▶ イベントハンドラーにより動作を指定できる。



イベントハンドラー

- ▶ イベントが発生したときに呼ばれる関数
 - メリット | イベントがないときは何もしなくても良い

イベント登録

- ボタン
Button
- テキスト
Text
- スライダー
Slider
- など

イベント発生！

- ボタンが押された
- ボタンを放した
- テキストが入力された
- スライダーが動いた
- など

何のイベントか??

イベントハンドラー

```
void ButtonPush()  
void ButtonRelease()  
void TextInput()  
void SliderChange()
```

イベントに対応した関数を実行

Ex0205 | スイッチとLED

```
#include <ESPUI.h>

void setup() {
  ESPUI.button("Push Button", &buttonCallback,
              COLOR_PETERRIVER, "Push");
  ESPUI.begin("スイッチとLED");
}

void loop() {
}

void buttonCallback(Control sender, int type) {
  switch (type) {
    case B_DOWN:
      digitalWrite(LED_PIN, HIGH);
      break;
    case B_UP:
      digitalWrite(LED_PIN, LOW);
      break;
  }
}
```

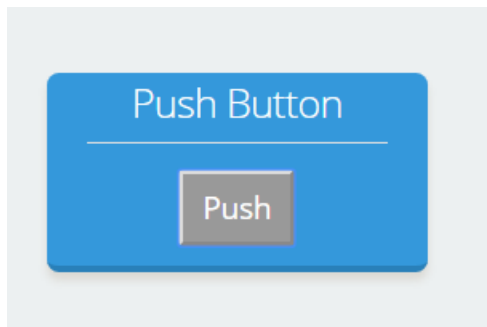
2019年10月12日(土)



スイッチのイベントハンドラー

▶ イベントハンドラーの登録

- `ESPUI.button("Push Button", &buttonCallback, COLOR_PETERRIVER, "Push");`



イベント発生！

スイッチの状態が変化

ON (B_DOWN)

または

OFF (B_UP)

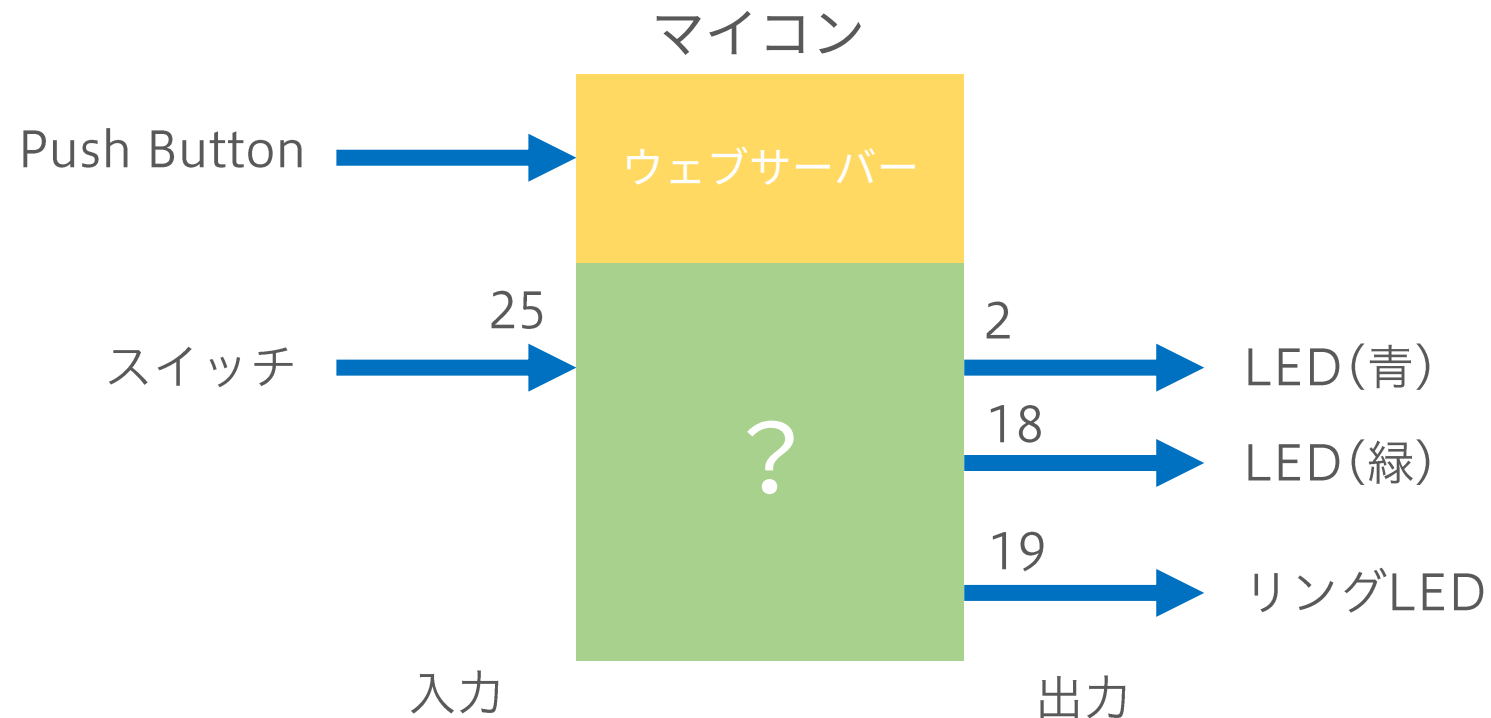


イベントハンドラー

```
void buttonCallback(Control sender, int type) {  
  switch (value) {  
    case B_DOWN:  
      digitalWrite(LED_PIN, HIGH);  
      break;  
    case B_UP:  
      digitalWrite(LED_PIN, LOW);  
      break;  
  }  
}
```

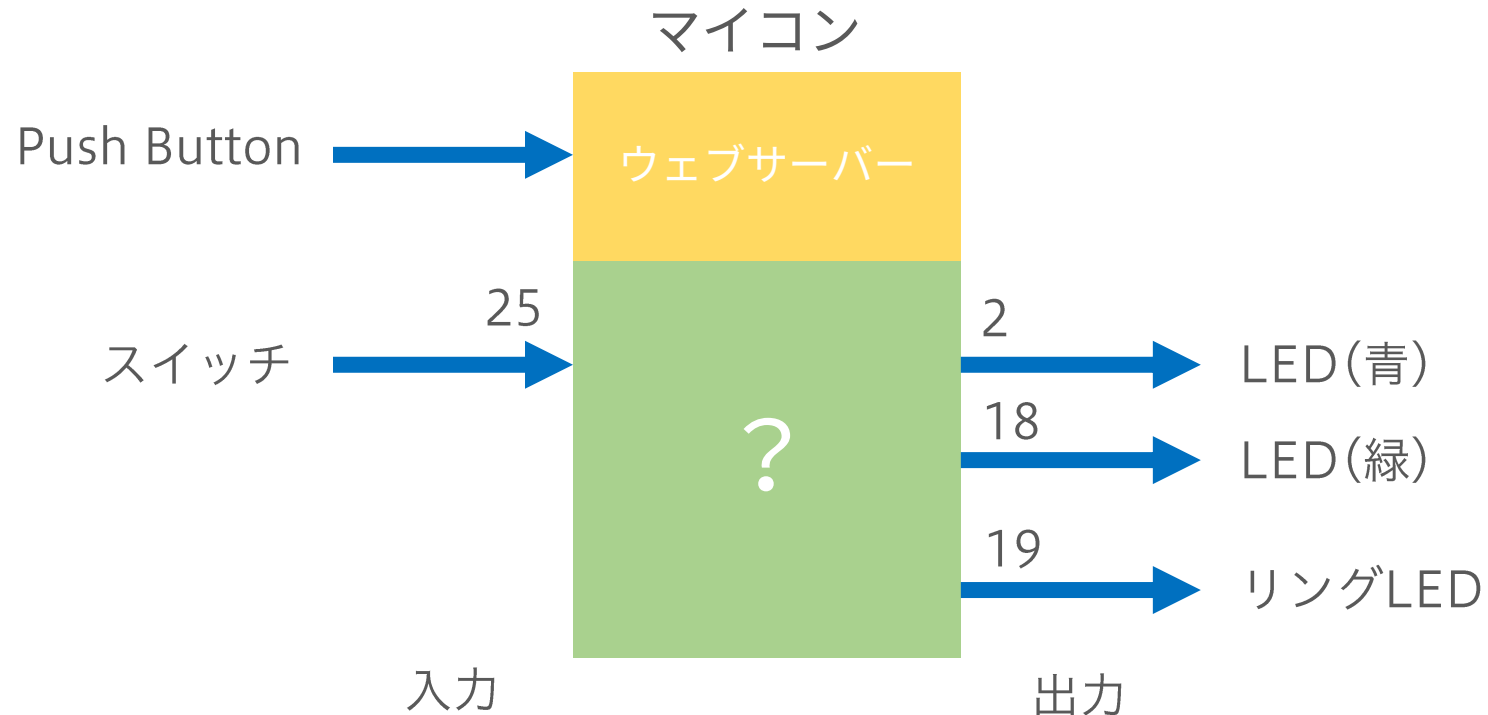
Ex0206 | スイッチとリングLED

- ▶ スイッチが押されたとき，リングLEDを点灯
- ▶ スイッチが押されていないとき，リングLED消灯



仮想化 (Virtualization)

- ▶ 物理的なデバイス (ハードウェア)
 - LED, スイッチ, センサなど
- ▶ ハードウェアの仮想化技術
 - ソフトウェアによって複数のハードウェアを統合し, 自由なスペックでハードウェアを再現する技術



講座の資料

▶ 講座資料(スライド, サンプルスケッチ等)

- <http://cms.db.tokushima-u.ac.jp/DAV/person/S10767/Share/R01Robot/>
- (例) スケッチの使い方
 - Example01.zipをダウンロード
 - ZIP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)

※ 必ずZIPファイルを展開!

展開(E)

キャンセル

The screenshot shows a CMS web interface. At the top, there is a navigation bar with the path: CMS > IDX > 徳島大学 > 工学部 > 知能情報工学科 > 辻 明典 > Share. The user profile is for 辻 明典 / Tuji, Akinori (Share/H29Sensor). A red box highlights a notification: 《EDB/CMSからの重要なお知らせ》 (August 30, 2015更新) お知らせを見る. Below this is a section titled 'コンテンツエリア' with a sub-section '場所の情報とサーバ切替 (ヘルプ)'. This section lists several links for different user roles: 一般閲覧用, 学生閲覧用(学内), 学生閲覧用(学外), 教職員閲覧・登録 (ID&Pass), and 教職員閲覧・登録 (EDB/PKI). Each link has a '切替' button. At the bottom, there is a 'コンテンツ' table with columns for Name, Last modified, Size, and Description. The table shows two entries: 'Parent Directory' and 'Lecture01/'.

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	-	