2019年度 人と地域共創センター公開講座(春・夏)

AI/IoTセンサのしくみを知ろう(基礎編)
第7回 Wi-Fi無線をつかう1

徳島大学技術支援部
辻　明典　博士(工学)
E-mail: a-tsuji@is.tokushima-u.ac.jp

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

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 5/11 ガイダンス，PC環境設定
② 5/18 プログラミングをはじめよう
③ 5/25 LED を光らせる(川上先生)
④ 6/ 1 温度・湿度をはかる(桑折先生)
⑤ 6/ 8 距離をはかる
⑥ 6/15 動きをはかる
⑦ 6/22 Wi-Fi無線をつかう1
⑧ 6/29 Wi-Fi無線をつかう2
⑨ 7/ 6 まとめ，振り返り

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

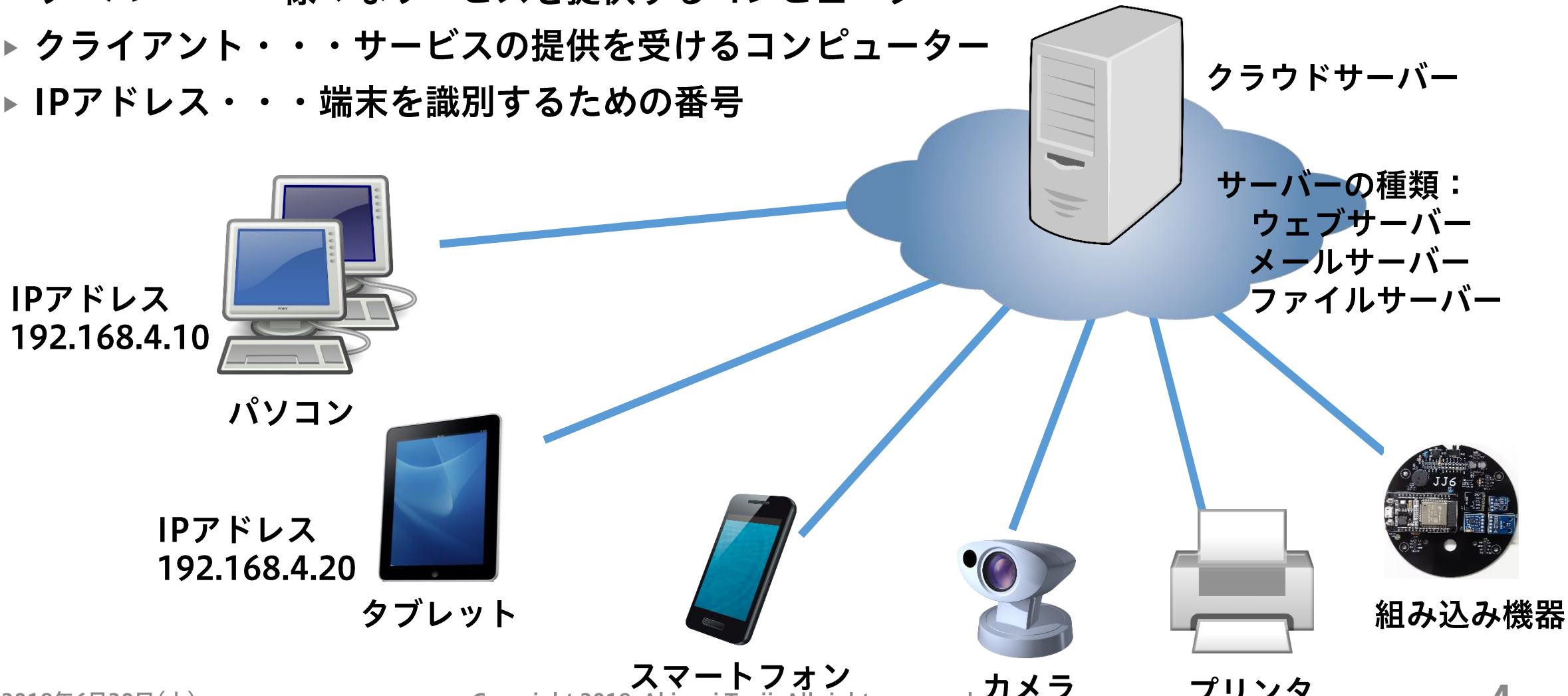
概要

- ネットワーク
  - WiFi無線
- ネットワーク構成
- リングLED取り付け
- 演習
  - Wi-Fi無線とLED
  - Wi-Fi無線とスイッチ
  - Wi-Fi無線と温湿度
  - Wi-Fi無線とスピーカー
- 考えてみよう

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

ネットワーク

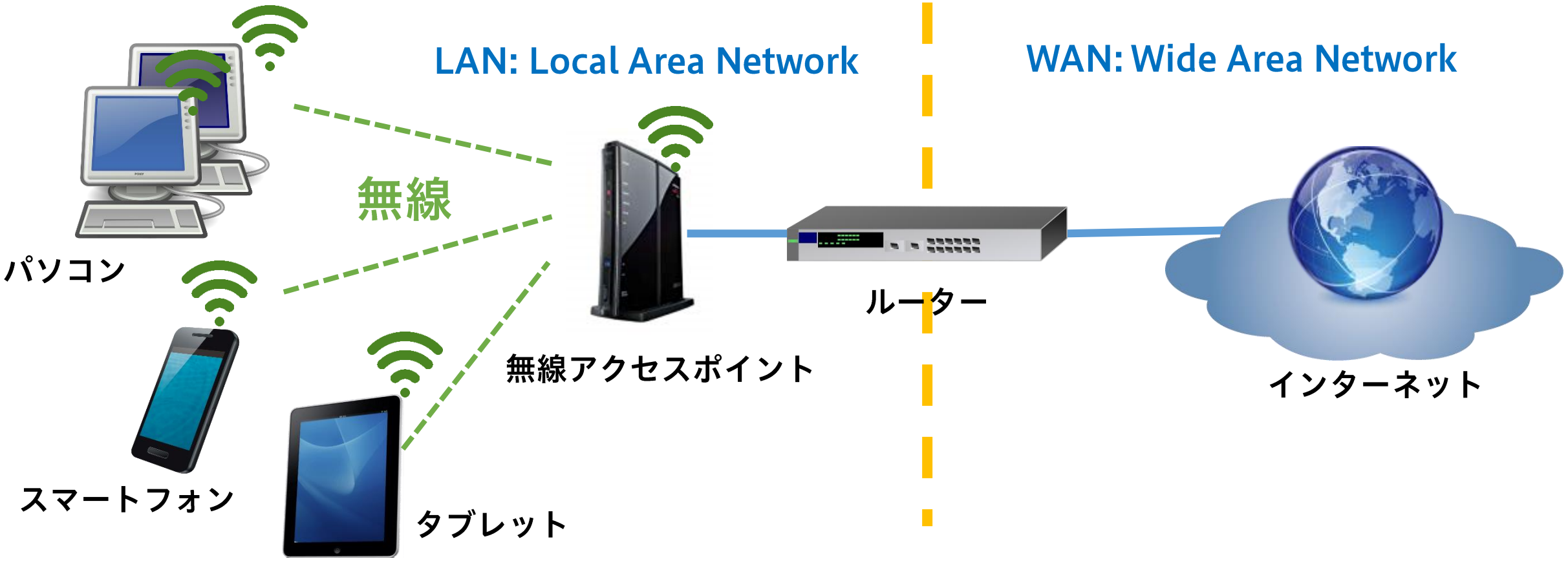
- サーバー・・・様々なサービスを提供するコンピューター
- クライアント・・・サービスの提供を受けるコンピューター
- IPアドレス・・・端末を識別するための番号



Copyright 2018, Akinori Tsuji, All rights reserved.

Wi-Fi無線

▶Wi-Fi無線・・・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



Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

ネットワーク構成(方式1)

- 方式1・・・インターネット未接続

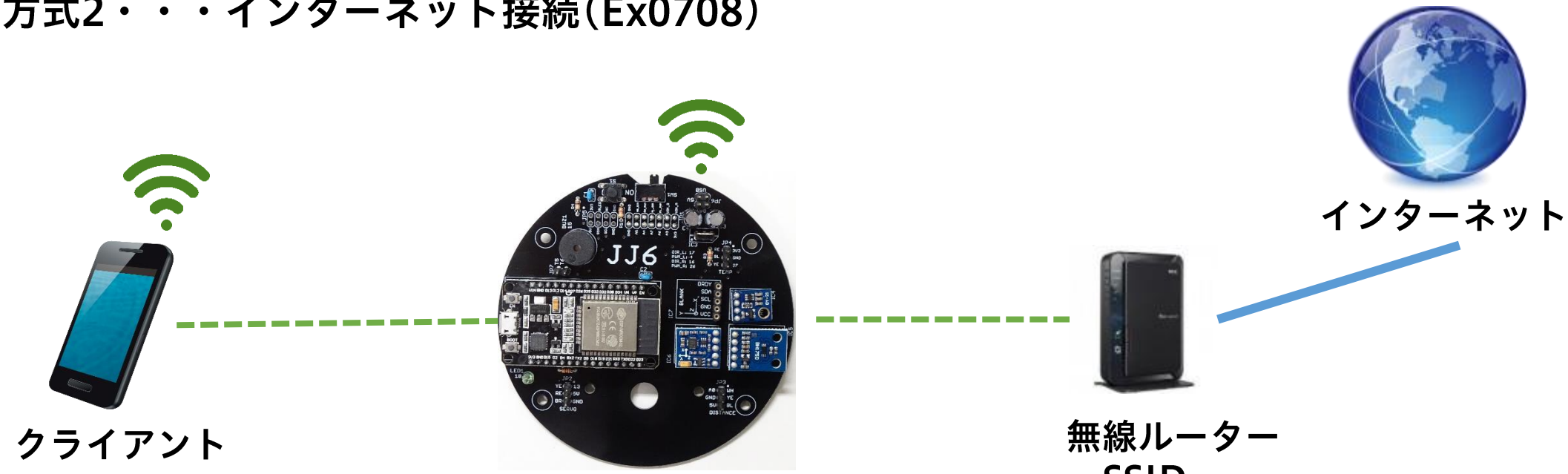
クライアント

無線アクセスポイント
兼ルーター
・SSID：jj6-<NN>
・IPアドレス
192.168.4.1
・ウェブサーバー

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

ネットワーク構成(方式2)

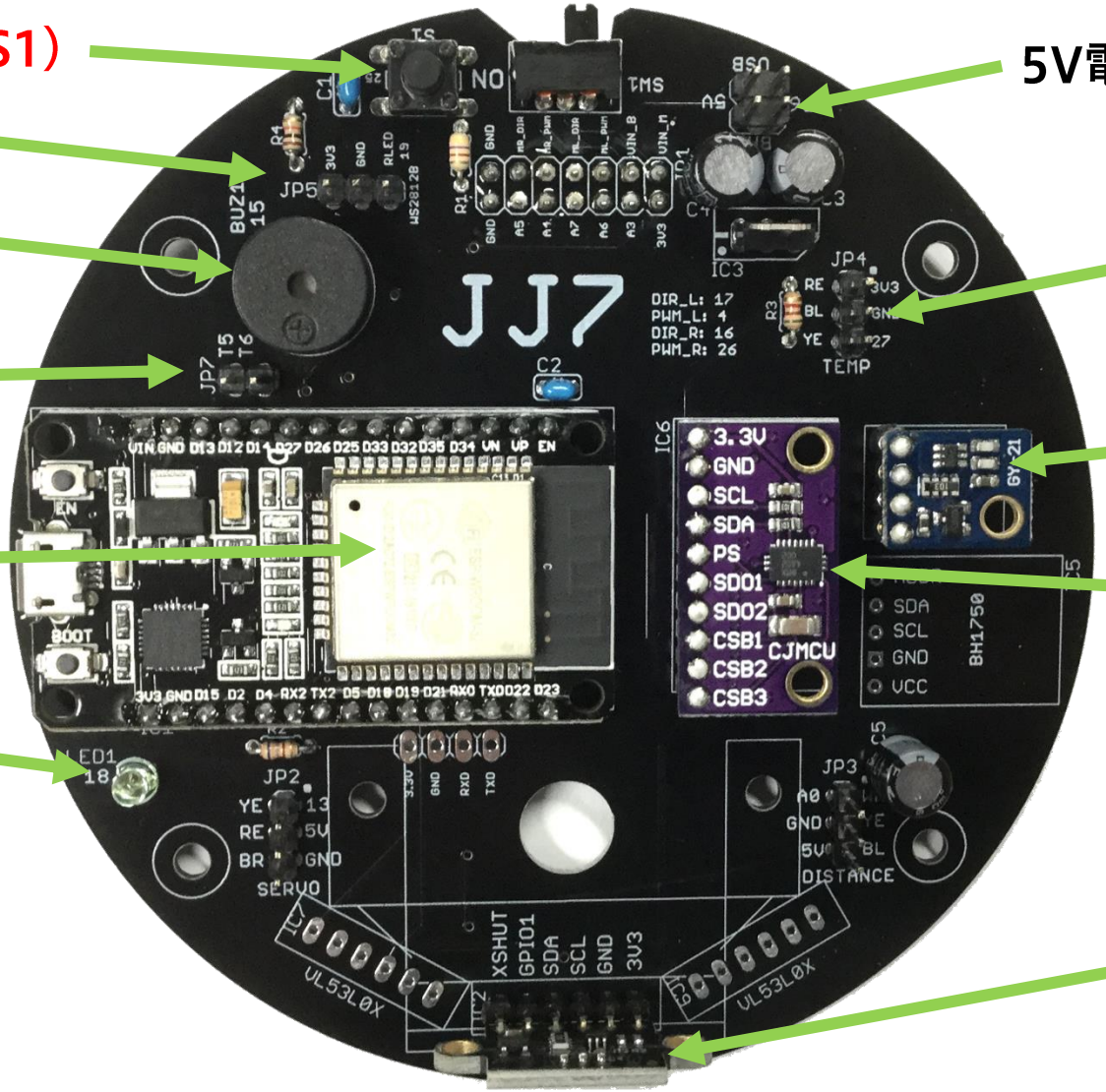
- 方式2・・・インターネット接続(Ex0708)



※ SSID: Service Set Identifier
無線LANを識別するための名前

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

マイコンボード（JJ7）



タクトスイッチ（S1）

LED（FastLED）

圧電ブザー（BUZ1）

タッチセンサ

マイコンESP32

LED1（緑）

5V電源供給（USB または BAT）

防水温度センサ
コネクタ

温湿度センサ

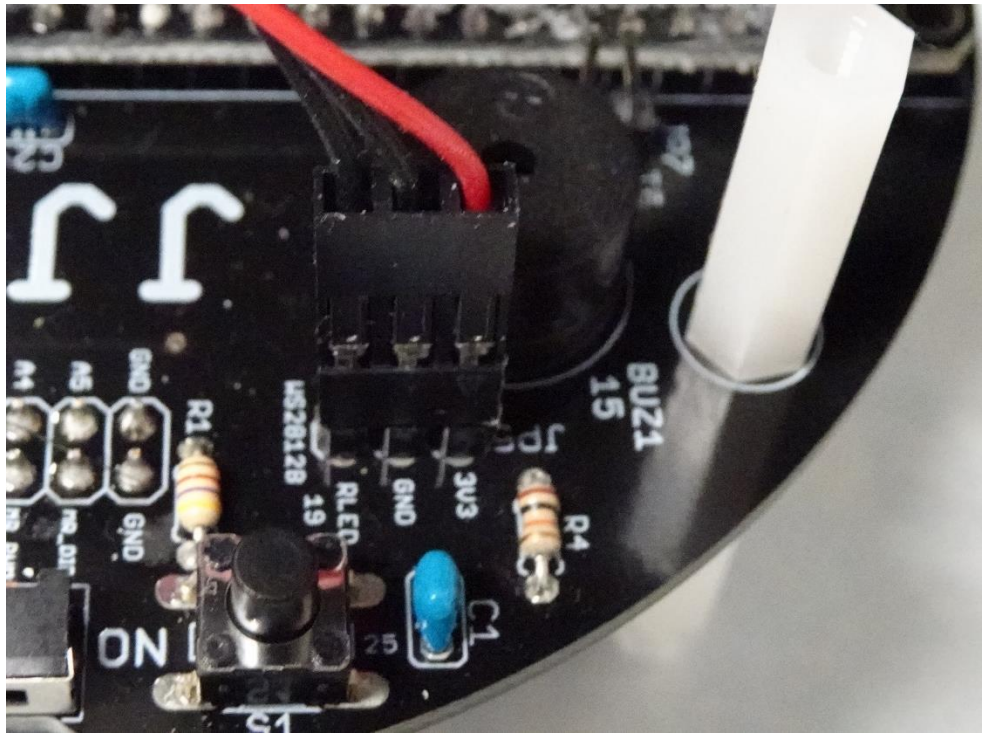
加速・地磁気・
ジャイロセンサ

レーザー距離センサ

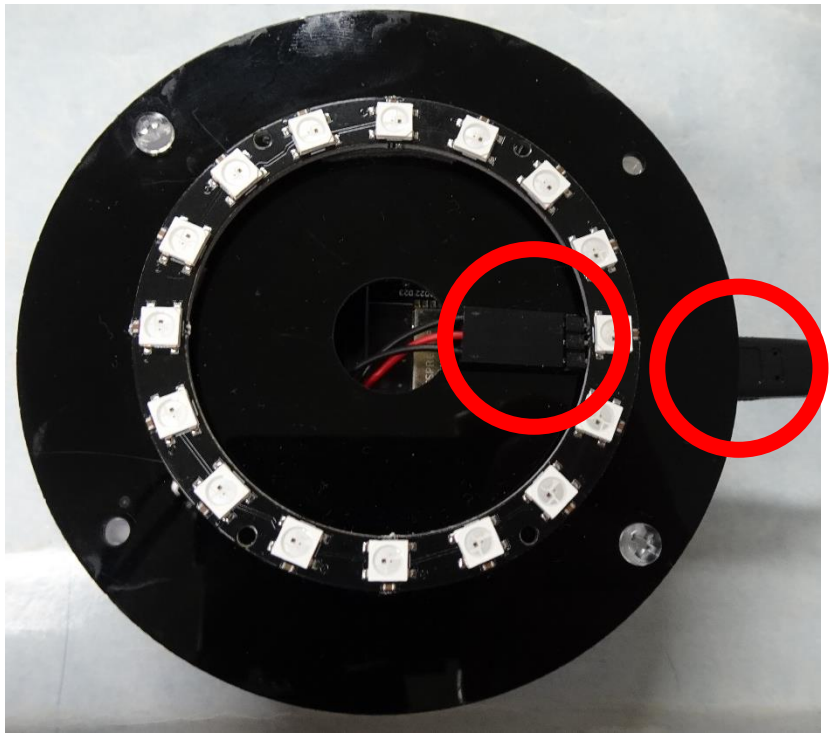
Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

リングLEDの取り付け

JP5（3V3:赤，GND：黒，RLED：黒）



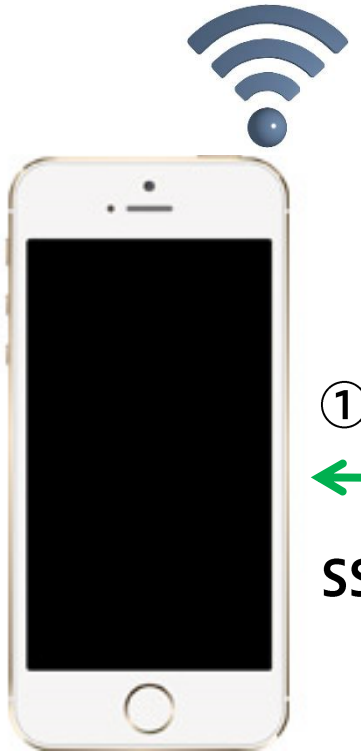
ネジで4カ所取り付け



USBコネクタとLEDコネクタの位置関係

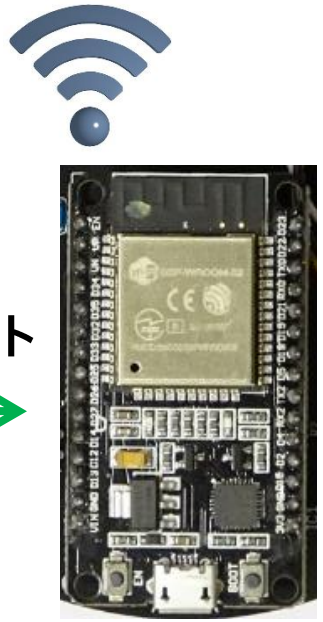
Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

Ex0701：スイッチによるLEDのON/OFF



① WiFiアクセスポイント

SSID: JJ7-nn

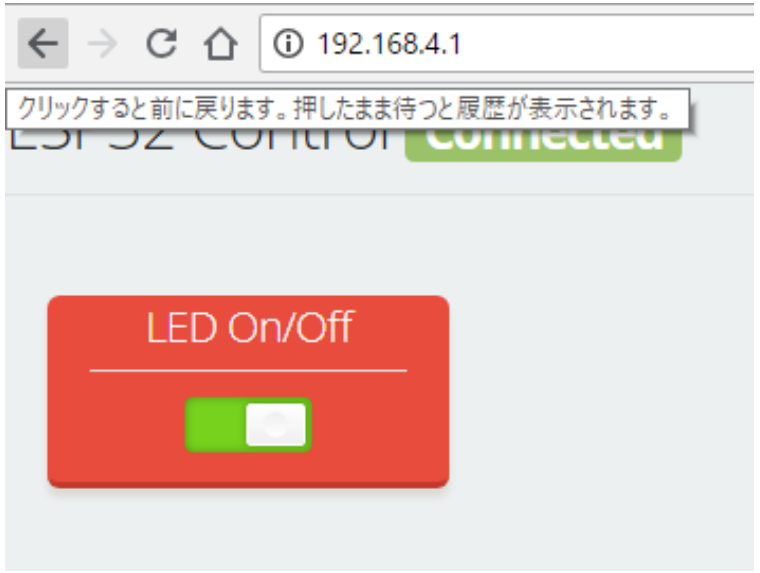


② ブラウザ表示
192.168.4.1

③ LEDのON/OFFボタンを表示

スイッチでLEDをON/OFFする

Pin 2: 青色LED
Pin18: 緑色LED

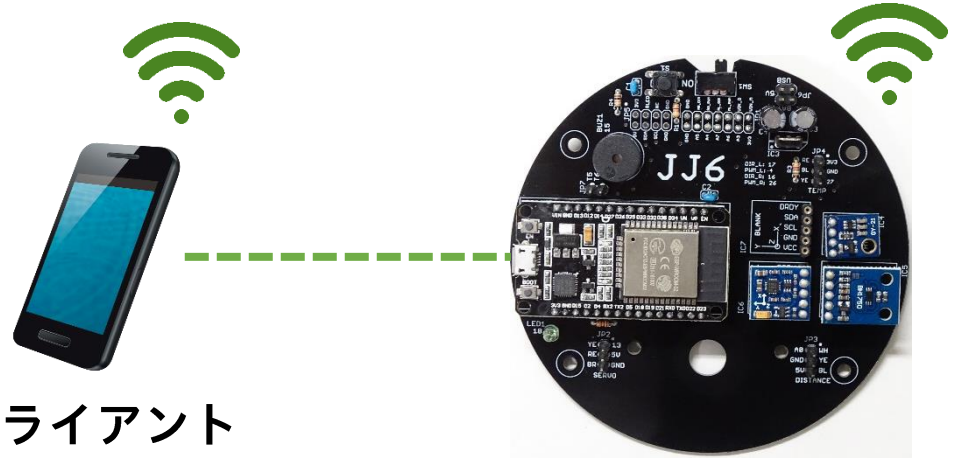


Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

WiFi設定の確認(方式1)

- WiFi設定，方式1に対応するコード

```
#include <WiFi.h>

const char *ssid = “JJ7-";
const char *password = "";
const IPAddress ip(192, 168, 4, 1);
const IPAddress subnet(255, 255, 255, 0);

WiFiServer server(80); // ウェブサーバー80番ポート


setup():
  WiFi.softAP(ssid, password);
  WiFi.softAPConfig(ip, ip, subnet);
  IPAddress myIP = WiFi.softAPIP();
```



クライアント
192.168.4.2

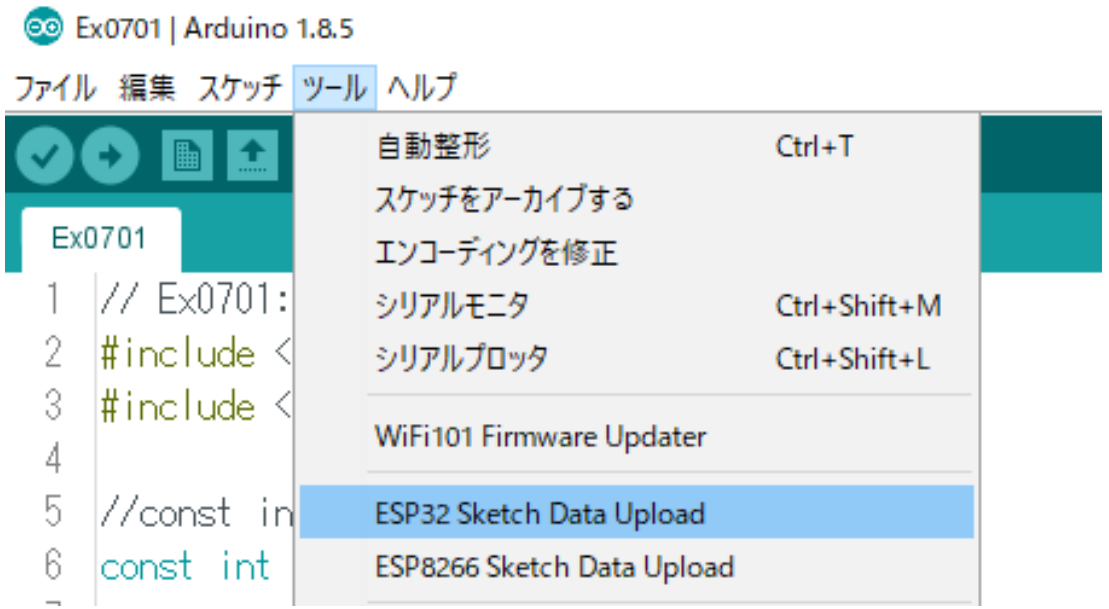
無線AP，ルーター
SSID: JJ7-
Password:
192.168.4.1

ウェブサーバー
192.168.4.1:80

Copyright 2018, Akinori Tsuji, All rights reserved.

Ex0701：スイッチによるLEDのON/OFF

- 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）。
- ssid，password を設定する。
- スケッチの書き込み。
- 無線AP（SSID）に接続する。
  接続時に設定したパスワードを入力する。
- ブラウザに192.168.4.1を入力して
  ウェブサーバーに接続する。
- スイッチをクリックして，LEDのON/OFFを確認する。

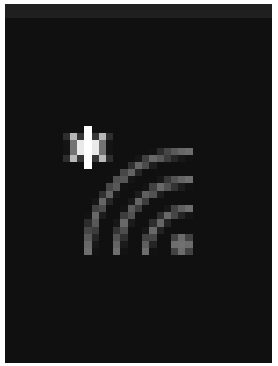


Copyright 2018, Akinori Tsuji, All rights reserved.

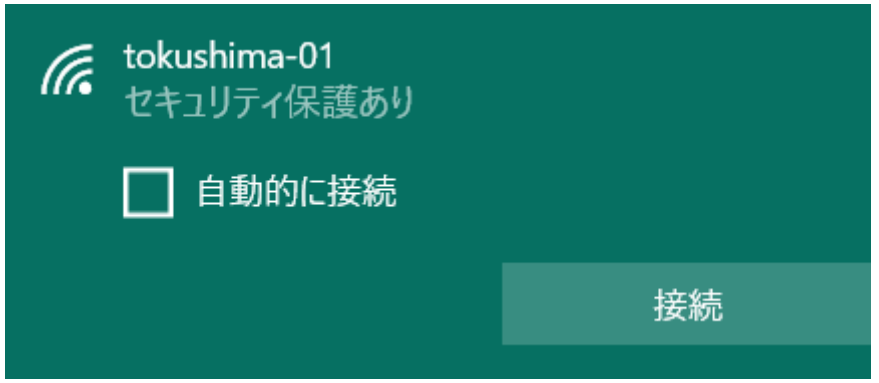
無線APの接続

▶ 無線APの接続｜プログラムを書き込む毎に接続作業が必要

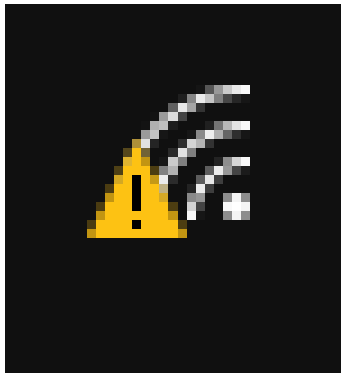
① タスクバーの無線アイコンをクリック



② 各自のSSIDを選択して接続



③ 無線接続が完了



＊ インターネットに接続しないため！が表示

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.

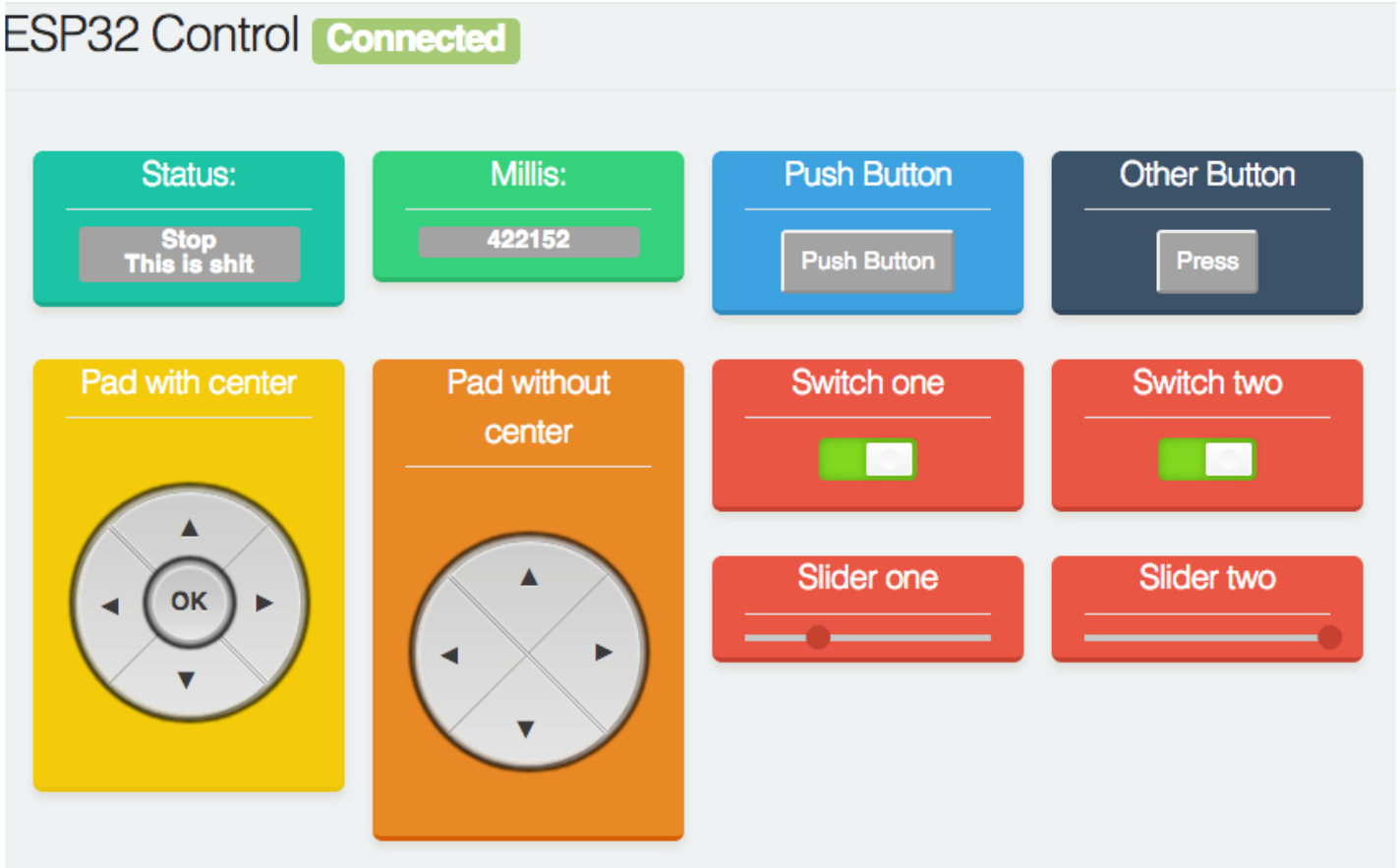
ESPUIライブラリ

- ウェブ上のユーザインタフェース(UI)作成

UIの種類：

- 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。



Copyright 2018, Akinori Tsuji, All rights reserved.

イベントハンドラー

- あるイベントに対して実行される関数

```
void setup() {
  ・・・
  ESPUI.switcher("LED On/Off", true, &Switch, COLOR_NONE);
  ESPUI.begin("ESP32 Control");
}

void Switch(Control sender, int value) {
  switch (value) {
    case S_ACTIVE:
      digitalWrite(LED_PIN, HIGH);
      break;
    case S_INACTIVE:
      digitalWrite(LED_PIN, LOW);
      break;
  }
}
```

- スイッチの操作（イベント）により，Switch関数が呼び出される。
- スイッチの状態（value）が，
  ON（S_ACTIVE）または，
  OFF（S_INACTIVE）
  を判定して，LEDをON/OFFする。

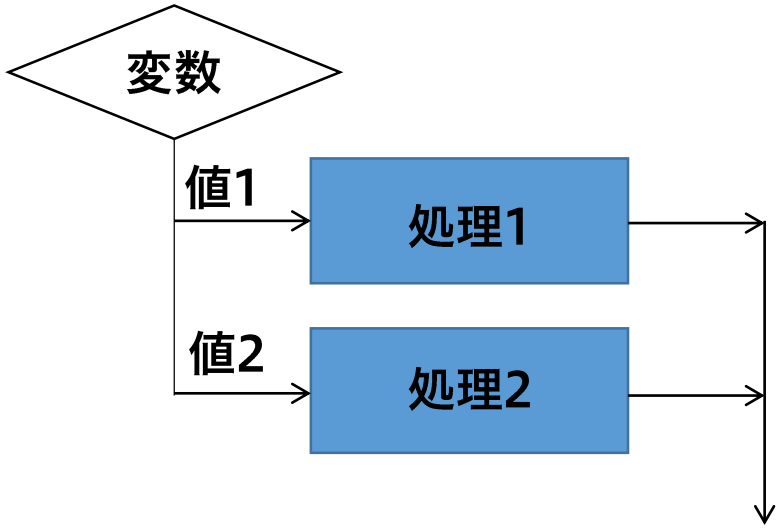
※ S_ACTIVE, S_INACTIVEなど，ESPUIライブラリで定義されている。

Copyright 2018, Akinori Tsuji, All rights reserved.

スイッチ(switch)文

- 条件分岐・・・スイッチ文

スイッチ文のフローチャート



構文

```
switch (変数) {
case 値1 :
   <処理1>
   break;
case 値2:
   <処理2>
    break;
}
```

実際のコード

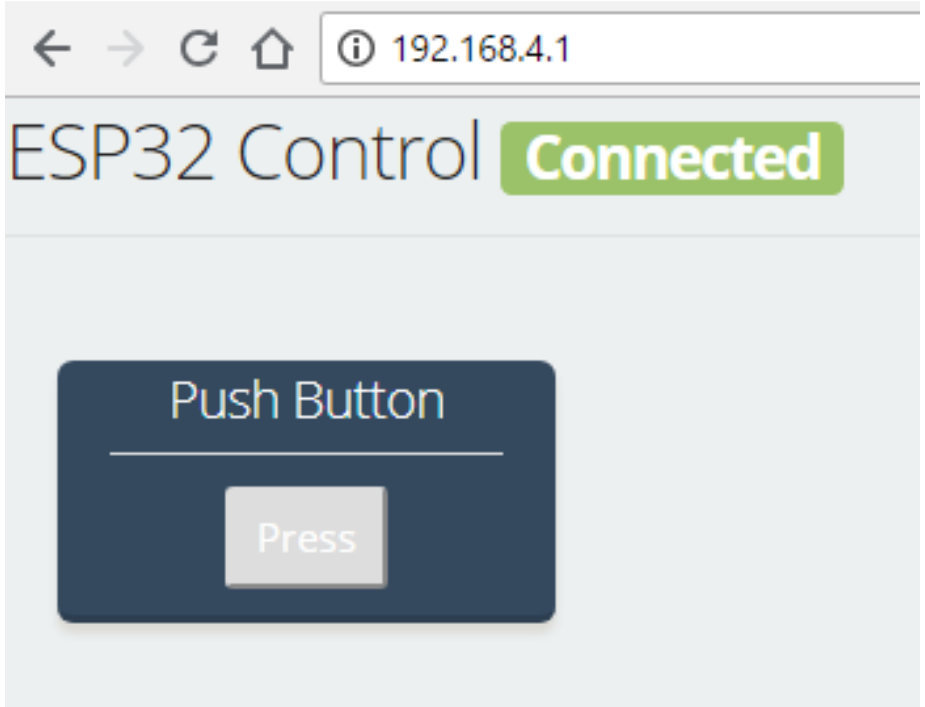
```
switch (value) {
   case S_ACTIVE:
     digitalWrite(LED_PIN, HIGH);
     break;
   case S_INACTIVE:
     digitalWrite(LED_PIN, LOW);
     break;
 }
```


Copyright 2018, Akinori Tsuji, All rights reserved.

Ex0702：押しボタンスイッチによるLEDのON/OFF

- ボタンが押されているときLEDがON。押されていないときLEDがOFF。

```
void setup() {
  ・・・
  ESPUI.button("Push Button", &Button,
                    COLOR_WETASPHALT, "Press");
  ESPUI.begin("ESP32 Control");
}
void Button(Control sender, int type) {
  switch (type) {
    case B_DOWN: // ボタンが押されている
      digitalWrite(LED_PIN, HIGH);
      break;
    case B_UP: // ボタンが押されていない
      digitalWrite(LED_PIN, LOW);
      break;
  }
}
```

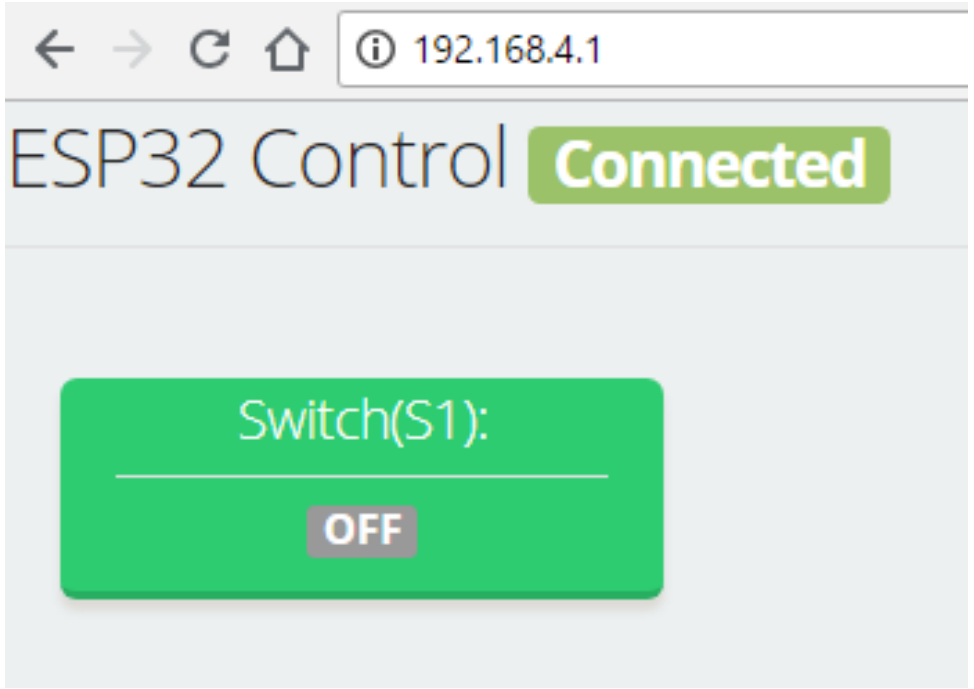


Copyright 2018, Akinori Tsuji, All rights reserved.

Ex0703：ボタンの状態を表示

- ボタンが押されているときラベルを“ON”,
  押されていないとき“OFF”
- 1秒（1000ms）毎にスイッチの状態を確認

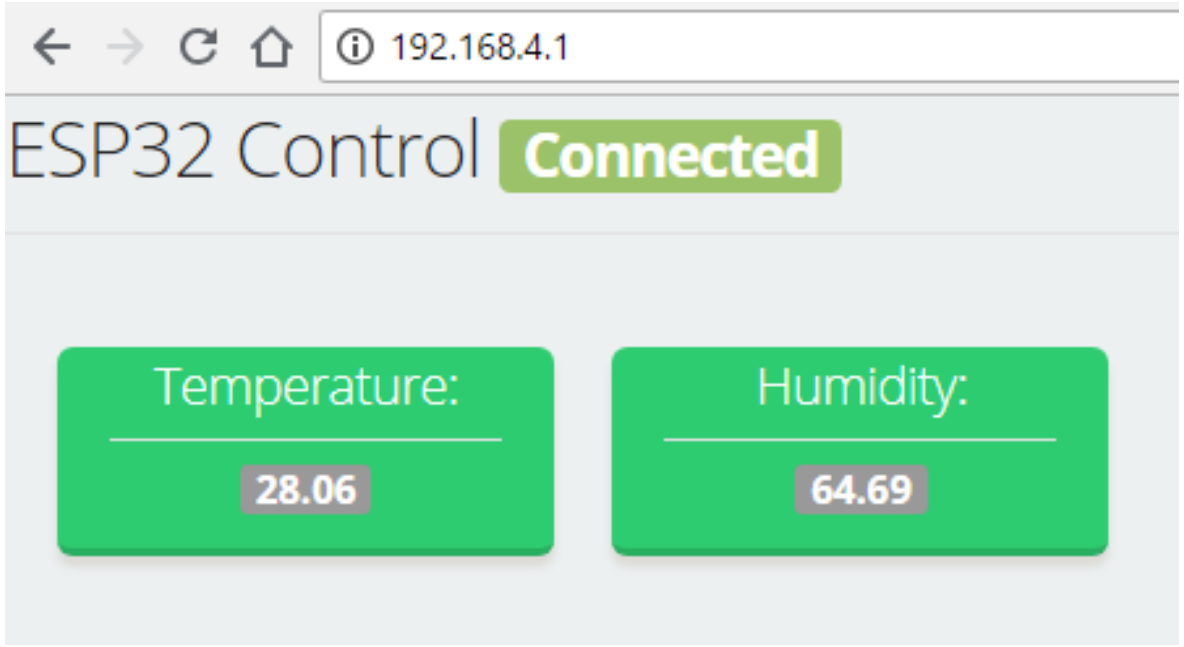
```
unsigned long prevTime = 0;

void loop(void) {
  if (millis() - prevTime > 1000) {
    prevTime = millis();
    if (digitalRead(SW_PIN) == LOW) {
        ESPUI.print("Switch(S1):", "ON");
    } else {
        ESPUI.print("Switch(S1):", "OFF");
    }
}
```



Copyright 2018, Akinori Tsuji, All rights reserved.

Ex0704：温度・湿度(HTU21D)をラベルで表示

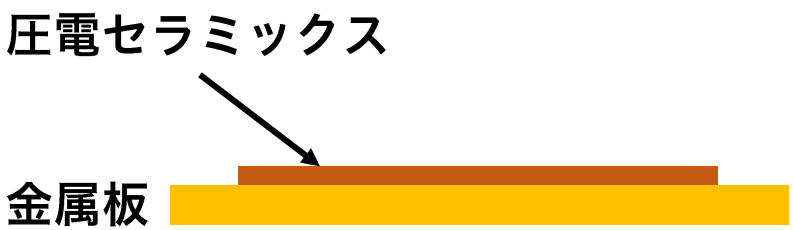
- 温度，湿度，それぞれ5秒毎にデータを更新

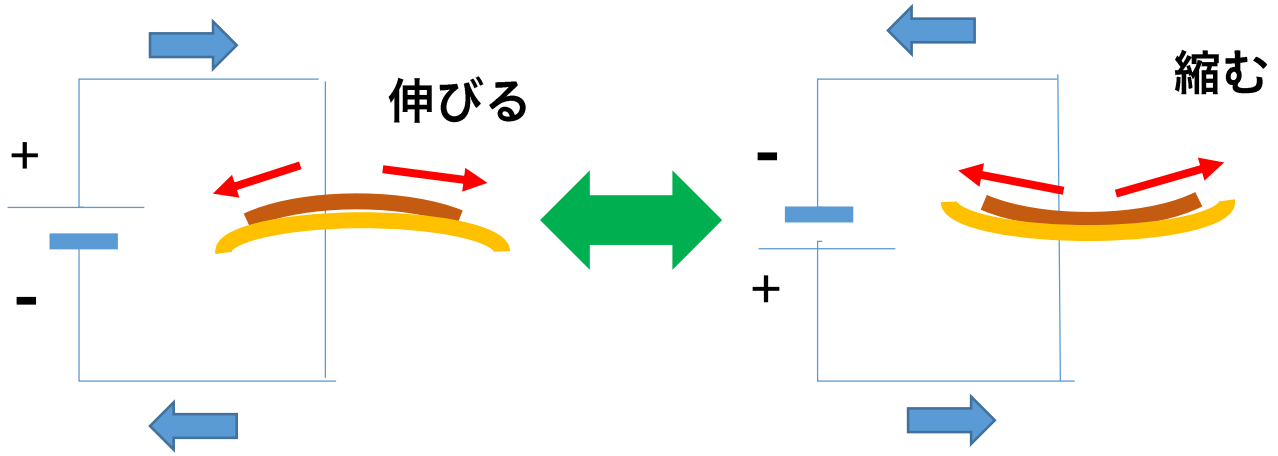


Copyright 2018, Akinori Tsuji, All rights reserved.

圧電スピーカー

- 圧電セラミックス
  - 高純度な酸化チタン・酸化バリウム等を高温度で焼き固めた多結晶体セラミックス
- **圧電正効果**：外部から力を加えると電圧が発生
  - 振動や音を電圧に変換
- **圧電逆効果**：電圧をかけると極性により伸縮
  - 電圧の変化を振動や音に変換





交互に繰り返すと振動して音が発生

Copyright 2018, Akinori Tsuji, All rights reserved.

周期と周波数

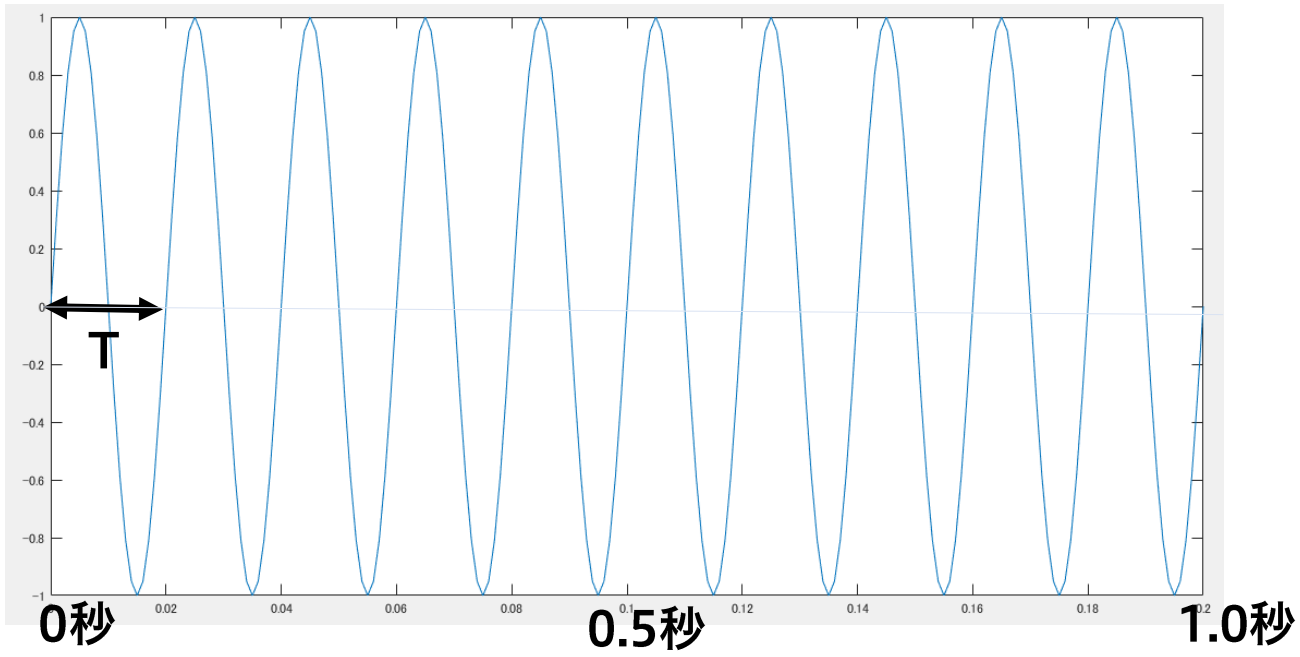
- 一定の周期で振動する波
  - 周期　T [s]・・・・・繰り返される波1つの時間
  - 周波数 f [Hz]・・・1秒間に繰り返される波の数
  - 周期と周波数の関係：f＝1 / T

例）電源の周波数：西日本 60Hz, 東日本 50Hz

1秒間に10回の波

周期 T= 0.1 [s]

周波数 f = 10 [Hz]



Copyright 2018, Akinori Tsuji, All rights reserved.

Ex0705：タイマーでブザーを鳴らす

- タイマーの設定
  - ピン番号　15番
  - チャネル 0
  - 精度　　13ビット
  - 周波数　5kHz

ledcWriteTone(チャネル, 周波数)
指定のチャネルに周波数の波形を生成する関数

```
const int SPEAKER_PIN = 15; // ブザー：15番ピン
#define LEDC_CH0       0   // タイマーのチャネル0 (0から15)
#define LEDC_TIMER_13BIT 13  // タイマーの精度13ビット
#define LEDC_BASE_FREQ  5000 // タイマーの周波数 5kHz(5000)
```

setup() ：

```
  ledcSetup(LEDC_CH0, LEDC_BASE_FREQ, LEDC_TIMER_13BIT);
  ledcAttachPin(SPEAKER_PIN, LEDC_CH0);
```

loop()：

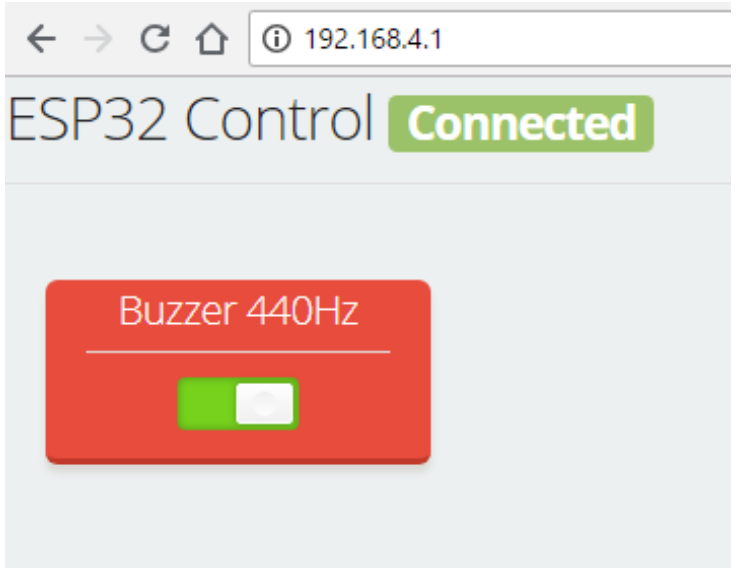
```
 ledcWriteTone(LEDC_CH0, 440); // トーンの開始 440Hz
 delay(300); // 300ms間，トーンを持続
 ledcWriteTone(LEDC_CH0, 0); // トーンの停止
 delay(1000); // 1秒間をおく
```


Copyright 2018, Akinori Tsuji, All rights reserved.

Ex0706：ボタンが押されたときブザーが鳴る

- ボタンが押されたことを検知して周波数440Hzでブザーを鳴らす
  - 5kHzのタイマー(CH0)を使用

```
ledcSetup(LEDC_CH0, LEDC_BASE_FREQ, LEDC_TIMER_13BIT);
ledcAttachPin(SPEAKER_PIN, LEDC_CH0);

ESPUI.switcher("Buzzer 440Hz", true, &Switch, COLOR_NONE);

void Switch(Control sender, int value) {
  switch (value) {
    case S_ACTIVE:
      ledcWriteTone(LEDC_CH0, 440); // トーンの開始 440Hz
      break;
    case S_INACTIVE:
      ledcWriteTone(LEDC_CH0, 0); // トーンの停止
      break;
  }
}
```



ledcWriteTone(チャネル, 周波数)
指定のチャネルに周波数の波形を生成する関数

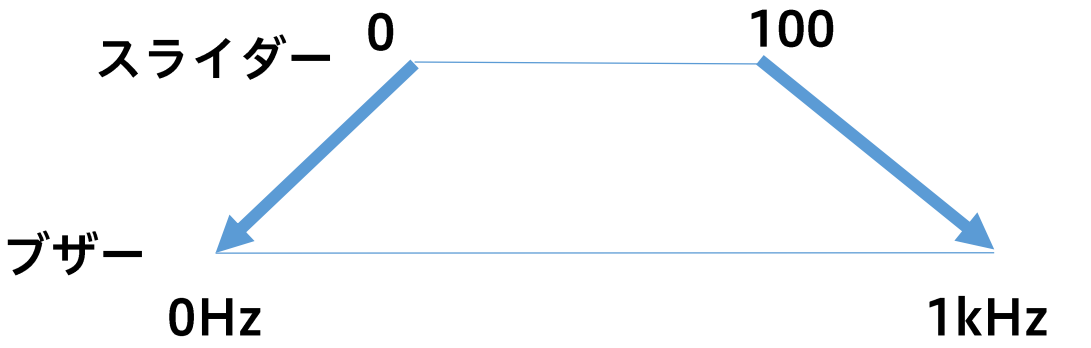
Copyright 2018, Akinori Tsuji, All rights reserved.

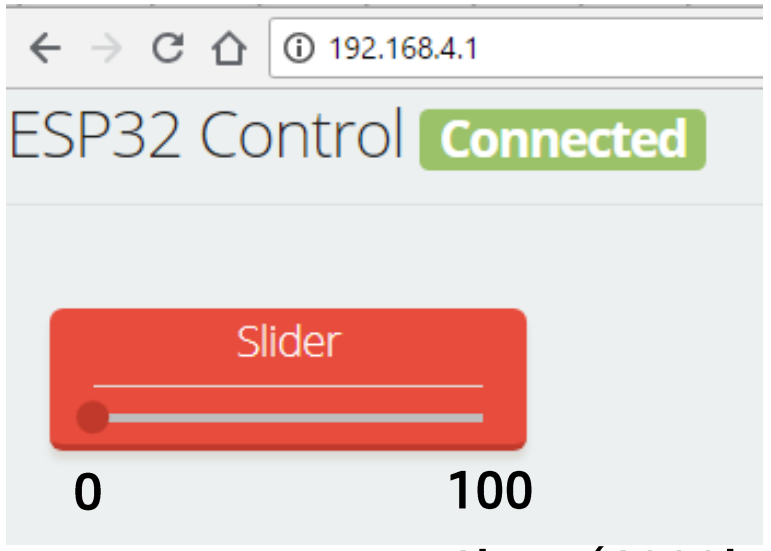
Ex0707: スライダーによる音の変化

- スライダーを動かすことで音（トーン）が0Hz～1kHzの範囲で変化

```
ESPUI.slider("Slider", &slider, COLOR_ALIZARIN, "0");

void slider(Control sender, int type) {
  Serial.println(sender.value);
  int n = map(sender.value.toInt(), 0, 100, 0, 1000);
  ledcWriteTone(LEDC_CH0, n); // 0Hz～1000Hz
}
```

map関数：ある数値の範囲を別の数値の範囲に置き換える。





Copyright 2018, Akinori Tsuji, All rights reserved.

考えてみよう

- Ex0701｜青色LEDに変更
- Ex0702｜青色LEDに変更
- Ex0703｜スイッチを押している間LEDを点灯，離したとき消灯を追加
- Ex0704｜温湿度の更新時間を変更
- Ex0706｜ボタンを押したときにメロディが流れる
- Ex0707｜音の変化する範囲を変更

Copyright 2016-2019, Akinori Tsuji, Tokushima University, All rights reserved.