

プログラミング基礎演習 / 説明資料 / arduino / SW

目標：

PC と Arduino の接続して、タクトスイッチの情報を読み取ります。

準備：

- 1) Arduino 開発ソフトの「Arduino IDE」を用意します。

※ 演習室の PC には、すでにインストールされているため不要です。

<https://www.arduino.cc/en/software/>

- Arduino IDE 2.3.7 . . . 高機能版、本格開発
- Legacy IDE 1.8.19 . . . 軽量・安定版、教育向け ★こちらを使用する

- 2) Arduino ボード、USB ケーブル、ブレッドボード、電子部品等を用意する

- 3) サンプルコードをダウンロードする

arduino_sample/

| | |
|-------------|--------------------------------|
| — led_1/ | · · · Arduino 内臓の LED を使ったサンプル |
| — led_2/ | · · · LED ライトを使ったサンプル |
| — sw/ | · · · タクトスイッチを使ったサンプル |
| — joystick/ | · · · ジョイスティックを使ったサンプル |

★今回使用

開発手順：

Step1 : ブレッドボードに回路を組む

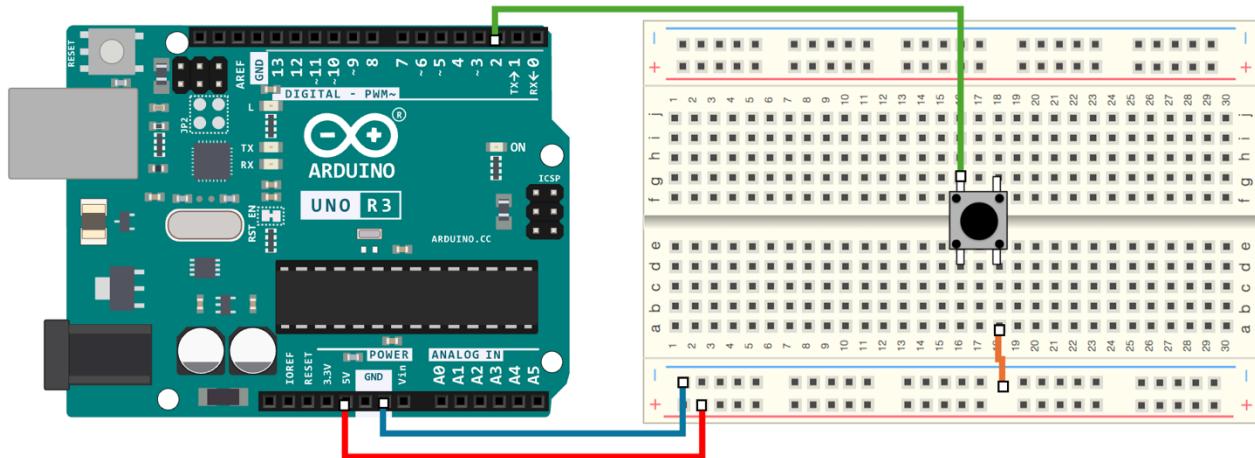
Step2 : Arduino IDE でプログラムを書く（スケッチを書く）

Step3 : Arduino IDE でデータの受信を確認する

Step4 : Web アプリと連携する

Step1：ブレッドボードに回路を組む

ブレッドボードにタクトスイッチの回路を組み、Arduino と配線します。



Step2：Arduino IDE でプログラムを書く

以下のコードを記述してください。コメントは割愛しています。

コードの解説は、配布ファイルに直接記述していますので、確認してください。

```
const int SW1_PIN = 2;
const unsigned long INTERVAL_MS = 100;
unsigned long lastSendTime = 0;

void setup() {
    pinMode(SW1_PIN, INPUT_PULLUP);
    Serial.begin(115200);
}

void loop() {
    unsigned long now = millis();
    if (now - lastSendTime >= INTERVAL_MS) {
        lastSendTime = now;
        sendSwitchState();
    }
}

void sendSwitchState() {
    int raw1 = digitalRead(SW1_PIN);
    int sw1 = (raw1 == LOW) ? 0 : 1;
```

```

Serial.print("sw1:");
Serial.println(sw1);
}

```

コードが記述できたら、「コンパイル（レ）」「書き込み（→）」を行ってください。

Step3 : Arduino IDE でデータの受信を確認する

(1) メニュー > シリアルモニタを開いてください。



(2) シリアルモニタのボーレートを"115200 bps"にします。



結果：

下から上にデータが流れます。ボタンを押すと「sw1:0」と表示されたら成功です。



Step4 : Web アプリと連携する

Web アプリとの連携サンプルを動かします。

※ Arduino IDE のシリアルモニタは必ず閉じてください。

(1) Arduino と Web アプリを接続する

index.html ファイルを開いて、「Connect」ボタンを押して下さい。

未接続 → 接続中 になったら接続成功です。

Switch Monitor

Connect 未接続

スイッチ状態 : OFF

(2) スイッチの動作を確認する

Arduino と接続しているスイッチを押すと、ブラウザのスイッチ状態が「ON」になれば成功です。

解説 :

(1) Arduino との通信用に使う JavaScript ファイルです。

このファイルは変更しないでください。また、app.js より先に（上に）記述してください。

index.html

```
<script src="js/arduino-io.js" defer></script>
```

(2) Arduino と接続する関数とイベントリスナーです。

app.js / handleArduinoConnect()

```
async function handleArduinoConnect() {  
    ... 略... コードの解説は直接ファイルを確認してください  
}
```

```
connectBtn.addEventListener('click', handleArduinoConnect);
```

async function と await とは？

async function とは 時間がかかる処理を含む関数であることを示す目印です。

この中では、処理が終わるのを「待つ」ことができます。

await とは この処理が終わるまで、次に進まないで待つという意味です。

(3) Arduino からのデータを取得する関数とイベントリスナーです。

```
app.js / handleArduinoData(e)  
async function handleArduinoData(e) {  
    ... 略... コードの解説は直接ファイルを確認してください  
}  
window.addEventListener('arduino-data', handleArduinoData);
```

Arduino から送信されたデータは、 arduino-io.js を介して受信されます。

取得したデータは、 data = e.detail; として格納されています。

Arduino の送信データが 「sw1:値」 の場合は、 data.sw1 で取得できます。