

第3講 スwitchの ON/OFF を取得する

Digital Input を用いてスイッチの ON/OFF を取得する。

1 前回の復習

前回やった Digital Output の復習をしましょう。ですが、前回やったことをもう一度やるだけではつまらないので、ついでに Processing の新しい命令を覚えましょう。

mousePressed

mousePressed 変数はマウスが押されているか押されていないかによって、それぞれ true と false に値が変わります。これを用いると、「マウスがクリックされたときに何かをする」という動作が実現できます。

```
if (mousePressed) {  
    // マウスが押されているときの処理  
}  
else {  
    // マウスが押されていないときの処理  
}
```

では、下のプログラムを参考にして前回の内容を思い出しながら、マウスをクリックしたら LED が点灯する、というプログラムを書いてみましょう。

回路は前回のままで OK です。

```
import processing.serial.*;  
import cc.arduino.*;  
  
Arduino arduino;  
int ledPin = 13;  
  
void setup() {  
    arduino = new Arduino(this, Arduino.list()[0], 57600);  
    arduino.pinMode(ledPin, Arduino.OUTPUT);  
}  
  
void draw() {  
    if (mousePressed) {  
        // ここに命令を追加する  
    } else {  
        // ここに命令を追加する  
    }  
}
```

ヒント

```
arduino.digitalWrite(n, Arduino.HIGH); // n 番ピンを High (5V) に  
arduino.digitalWrite(n, Arduino.LOW);  // n 番ピンを Low (5V) に
```

2 Digital Input

ここからが本題です。Arduino の Digital Input を用いてスイッチの ON/OFF を取得してみましょう。

プルアップ/プルダウン抵抗

デジタル回路の場合、入力端子がどこにも接続されていないような状態 (オープン) が起こると、電圧が High(5V) または Low(0V) に定まらず誤動作の原因になります。そのため、マイコンの入力信号が High か Low かを確実に伝えるために、プルアップ/プルダウン抵抗が必要となります。

2.1 プルアップ抵抗

図1を見ると、5V 側に抵抗が付いています。これをプルアップ抵抗と言います。

スイッチが押されていない時は、図1の様に抵抗を通してデジタル入力につながるので 5V になり、arduino では HIGH で読み取られます。

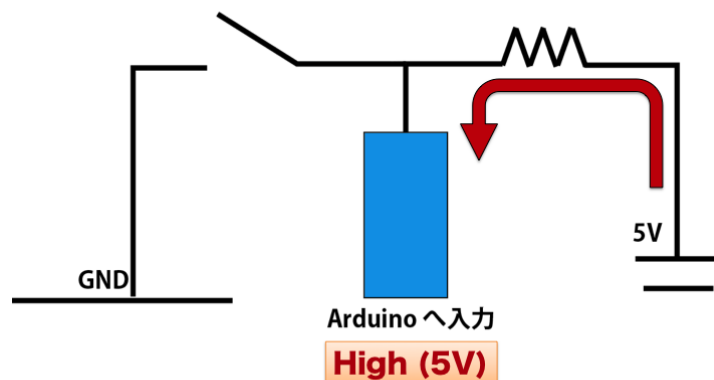


図 1: プルアップ回路 (スイッチが押されていない時)

スイッチが押された場合は、図2の様にスイッチを通して GND とつながるので 0V になります。電流は青い線のように流れ、デジタル入力側には流れないので 0V になり、arduino では LOW で読み取られます。

2.2 プルダウン抵抗

図3を見ると、GND 側に抵抗が付いています。これをプルダウン抵抗と言います。スイッチが押されていない時は、図3のように抵抗を通してデジタル入力は GND とつながるので 0V になり、arduino では LOW で読み取られます。

スイッチが押された場合は、図4のようにスイッチを通して 5V とつながるのでデジタル入力は 5V になります。

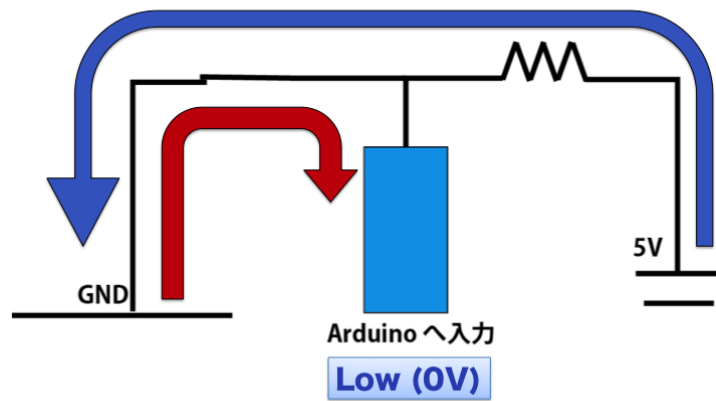


図 2: プルアップ回路 (スイッチが押されていない時)

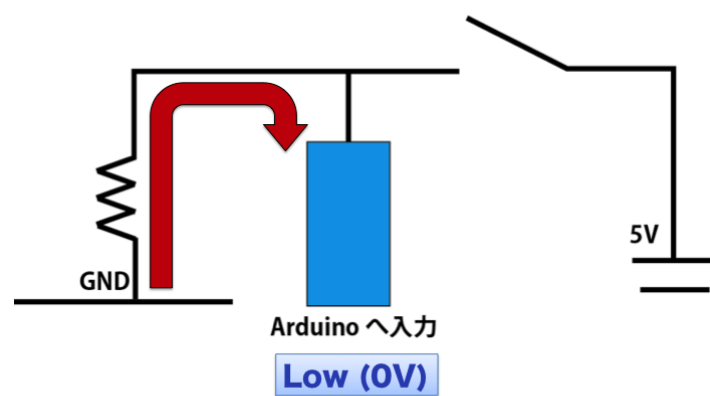


図 3: プルダウン回路 (スイッチが押されていない時)

電流は青い線のように流れます。

arduino では HIGH で読み取られます。

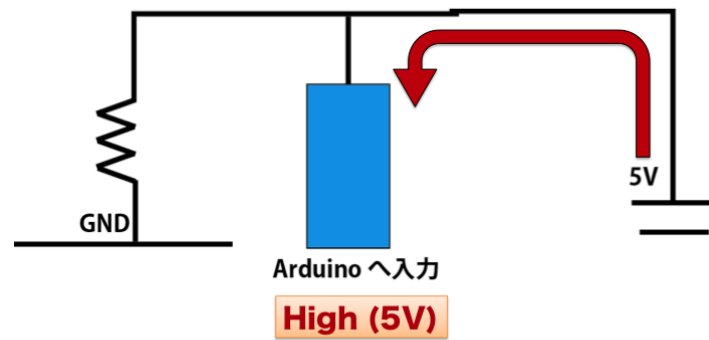


図 4: プルダウン回路 (スイッチが押されている時)

2.2.1 回路を組む

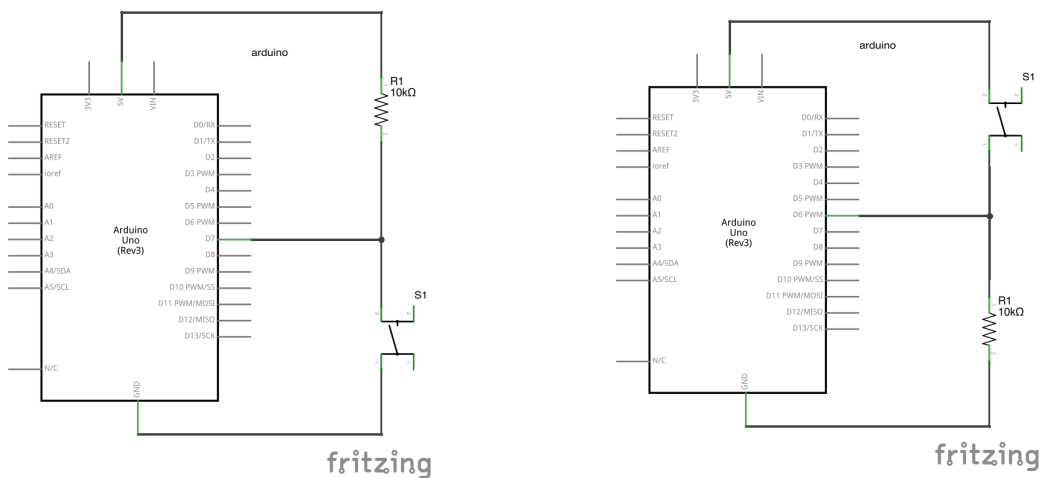


図 5: プルアップ抵抗 (左) とプルダウン抵抗 (右)

2.2.2 プログラムを書く

```
import processing.serial.*;
import cc.arduino.*;

Arduino arduino;
int switchPin = 8; // スイッチを接続したピンの番号

void setup() {
  size(400, 300);
```

```
    arduino = new Arduino(this, Arduino.list()[0], 57600);  
    arduino.pinMode(switchPin, Arduino.INPUT); // ピンモードを Input に  
}  
  
void draw() {  
    // 8番ピンの電圧を取得し、それが HIGH ならば  
    if (arduino.digitalRead(switchPin) == Arduino.HIGH) {  
        background(255, 0, 0); // 背景を赤に  
    } else {  
        background(0, 0, 0);    // そうでなければ (LOW ならば) 背景を黒に  
    }  
}
```

3 Digital Input と Digital Output を組み合わせる

スイッチの入力を Processing で取得し、それに基づいて LED を制御しましょう。前回やった Digital Input と今回やった Digital Output の合わせ技です。

これで入力と出力の両方が実現できるようになります。次回以降の実習でも入力や出力のための部品が変わるだけで基本的な考え方は同じです。

TRY

前回と今回やったことを思い出しながら、回路とプログラムを作成してみましょう。