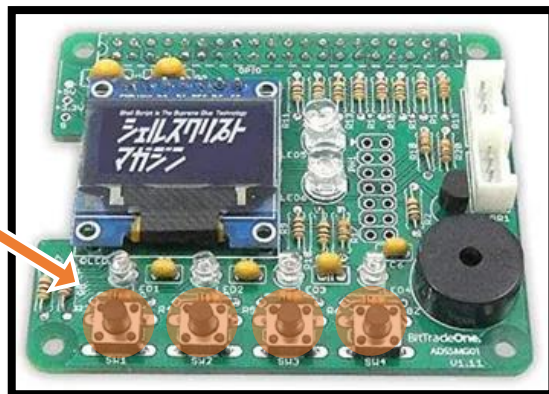


SWITCH (スイッチ) とは

SWITCH (スイッチ) とはボタンを押す、つまみを捻るなどといった操作により電子回路上の通電を開放・遮断する電子部品のこと。

入門ボードにはタクトスイッチが4個搭載されており、ボタンを押す事によって接続されている GPIO ピンがグラウンドに繋がり、電圧を **LOW** に落とすことができる。



デジタル入力とは

スイッチの状態を読み取る方法として、今回はデジタル入力を使用する。

デジタル入力とは、電子回路上に流れる電圧の値を読み取り、0 と 1 の二値に変換する信号のこと。

スイッチを押した場合、GPIO ピンに掛かる電圧が **LOW** となり、**0 が入力された**と読み取り、

スイッチを離した場合、GPIO ピンに掛かる電圧が **HIGH** となり、**1 が入力された**と読み取る。

スイッチの GPIO 番号

4つのスイッチに接続されている GPIO 番号は以下の通りです。

例えば、SW1 が押されているか判定したい場合は7番の電圧が **LOW (0)** になっているかを読み取り、離しているか判定したい場合は7番の電圧が **HIGH (1)** になっているかを読み取ります。

表5.1.1 Raspberry Piの拡張コネクタ

備考	機能	ピン名	GPIO番号	ピン番号	ピン番号	GPIO番号	ピン名	機能	備考
(7番ピン合計)		3.3V		1	2	5V			
プルアップ抵抗付	I ² C1 (SDA)	GPIO2	8	3	4	5V			
	I ² C1 (SCL)	GPIO3	9	5	6	GND			
	GPCLK0	GPIO4	7	7	8	15	GPIO14	UART0 (TXD)	起動時にシリアルコンソールとして使用
	GND			9	10	16	GPIO15	UART0 (RXD)	起動時にシリアルコンソールとして使用
	SPI1 (CS1)	GPIO17	0	11	12	1	GPIO18	SPI1 (CS0), PWM0	その他の機能: PCM_CLK
		GPIO27	2	13	14	GND			
		GPIO22	3	15	16	4	GPIO23		
(17番ピン合計)		3.3V		17	18	5	GPIO24		
	SPI0 (MOSI)	GPIO10	12	19	20	GND			
	SPI0 (MISO)	GPIO9	13	21	22	6	GPIO25		
	SPI0 (SCLK)	GPIO11	14	23	24	10	GPIO8	SPI0 (CS0)	
	GND			25	26	11	GPIO7	SPI0 (CS1)	
	ID_SD			27	28		ID_SC		拡張基板 (Hat) に搭載のEEPROM用信号
	GPIO21			29	30	GND			
	GPIO6			31	32	26	GPIO12	PWM0	
	PWM1	GPIO13	23	33	34	GND			
	PCM_FS	SPI1 (MISO), PWM1	GPIO19	24	36	27	GPIO16	SPI1 (CS2)	
		GPIO26	25	37	38	28	GPIO20	SPI1 (MOSI)	その他の機能: PCM_DIN
	GND			39	40	29	GPIO21	SPI1 (SCLK)	その他の機能: PCM_DOUT

※ Raspberry Pi タイプB は26ピンまでになります。

拡張コネクタのピンアサイン

練習（TrySwitch.java）

以下のサンプルプログラムを記述して実行し、SW1 を押している間は LED1 が点灯し、SW1 を離している間は LED1 が消灯するか確認しなさい。

```
/*
 * TrySwitch.java
 * Date   : 2022/01/01
 * Author : IE1A 99 K.Murakami
 */

// GPIO ピンを利用するために必要なクラスを読み込む
import com.pi4j.wiringpi.Gpio;
import com.pi4j.wiringpi.GpioUtil;

public class TrySwitch {
    // Thread.sleep メソッドで発生する割り込み例外を throws する
    public static void main (String[] args) throws InterruptedException {

        System.out.println("プログラム開始");

        // デジタル出力信号を定数化
        final int HIGH = 1;
        final int LOW = 0;
        // デジタル入力信号を定数化
        final int ON = 0;
        final int OFF = 1;

        // LED のピン番号を宣言
        final int LED1 = 15;
        // SW のピン番号を宣言
        final int SW1 = 7;

        // GPIO を初期化
        Gpio.wiringPiSetup();
        System.out.println("GPIO 初期化完了");

        // 各 LED を出力に設定
```

```

    Gpio.pinMode(LED1, Gpio.OUTPUT);
    // 各 SW を入力に設定
    Gpio.pinMode(SW1, Gpio.INPUT);
    System.out.println("GPIO 入出力設定完了");

    // プログラムを終了させない為に無限ループする
    while(true) {
        // SW1 が ON になっている場合
        if(Gpio.digitalRead(SW1) == ON) {
            Gpio.digitalWrite(LED1, HIGH); //LED1 点灯
        }else{
            Gpio.digitalWrite(LED1, LOW); //LED1 消灯
        }
        // 100 ミリ秒待機する
        Thread.sleep(100);
    }
}

```

この命令 (**if 文**) により SW1 が押されているか判定している

else は**それ以外**という命令で、if で記述した SW1 が ON になっている場合**以外 (OFF の場合)** に処理が行われる

この待機する命令は**チャタリング**というスイッチの ON/OFF が誤認識される現象を防ぐために記述している
これはプログラムの命令が高速で実行されているが故に起こる現象の為、while 文の最後に少しの待機命令を記述するとチャタリングを除去して誤作動を防ぐことができる

課題 1 (SwitchLit.java)

前述の練習を**別名保存**し、SW2 が押されている場合に LED2 を、SW3 が押されている場合に LED3 を、SW4 が押されている場合に LED4 を点灯する処理を追加したプログラムを作成しなさい。



押したスイッチの上にある LED が点灯し、離したら消灯させて 4 つ全て連動させる

課題 2 (SwitchAll.java)

4 つのスイッチ**全てが押されている間**のみ、0.5 秒間隔で 4 つの LED が点滅を繰り返すプログラムを作成しなさい。スイッチが押されているかの判定 if 文では**論理演算子**を使用すること。



スイッチを 4 つ同時に押している間のみ点滅



チャタリングの補足

タクトスイッチを押した時の電圧をオシロスコープで確認すると以下のような波形が確認できる。

