

Arduinoをはじめよう

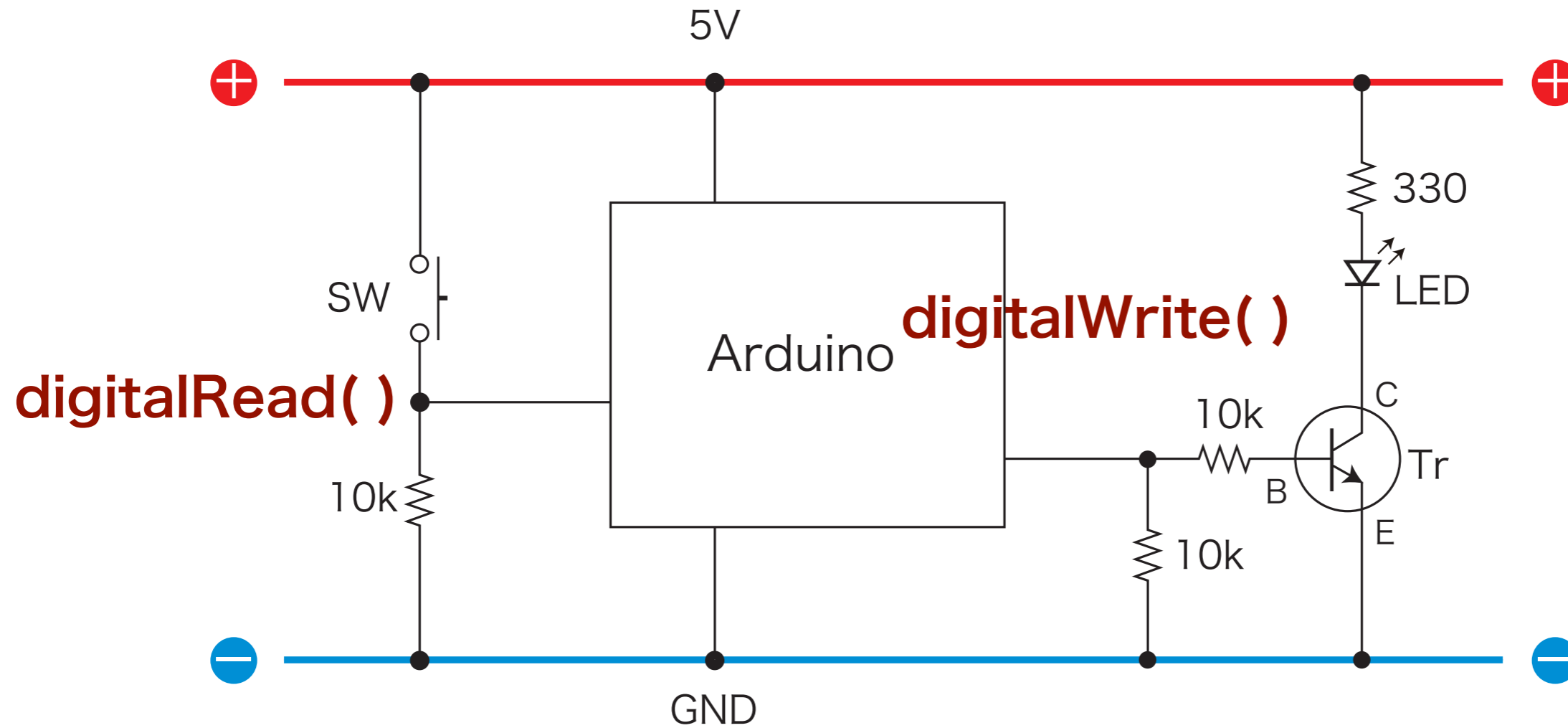
digitalReadとスイッチ

川上 博

2013/11/21

今日の内容

第3回： 2種類のスイッチとそれらを使ったプログラム

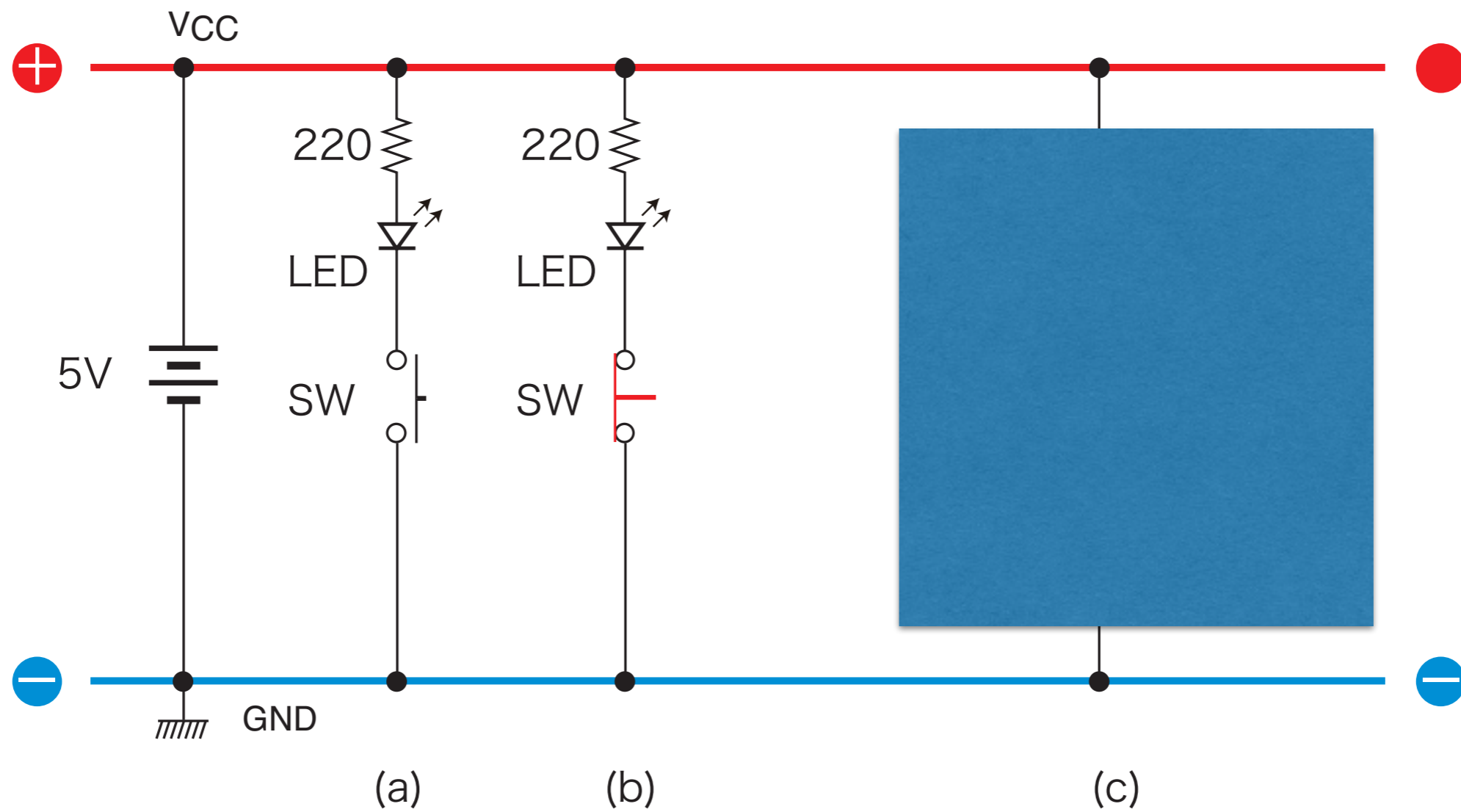


クイズ

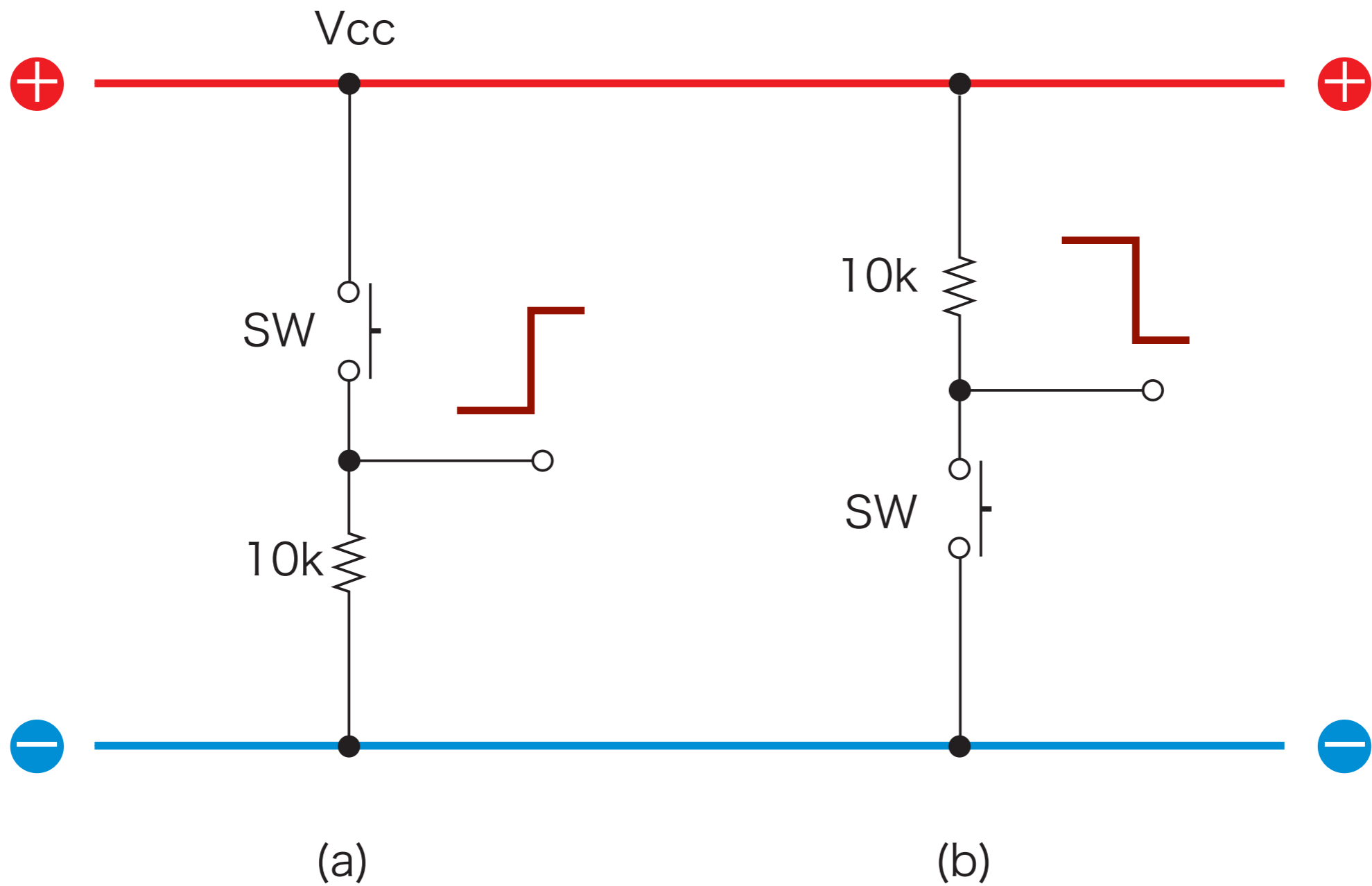
「スイッチを押したらLEDが点灯する回路」
をブレッドボードに作りなさい

「もともと点灯していたLEDがスイッチを押
したら消える回路」を作りなさい

クイズ：答え

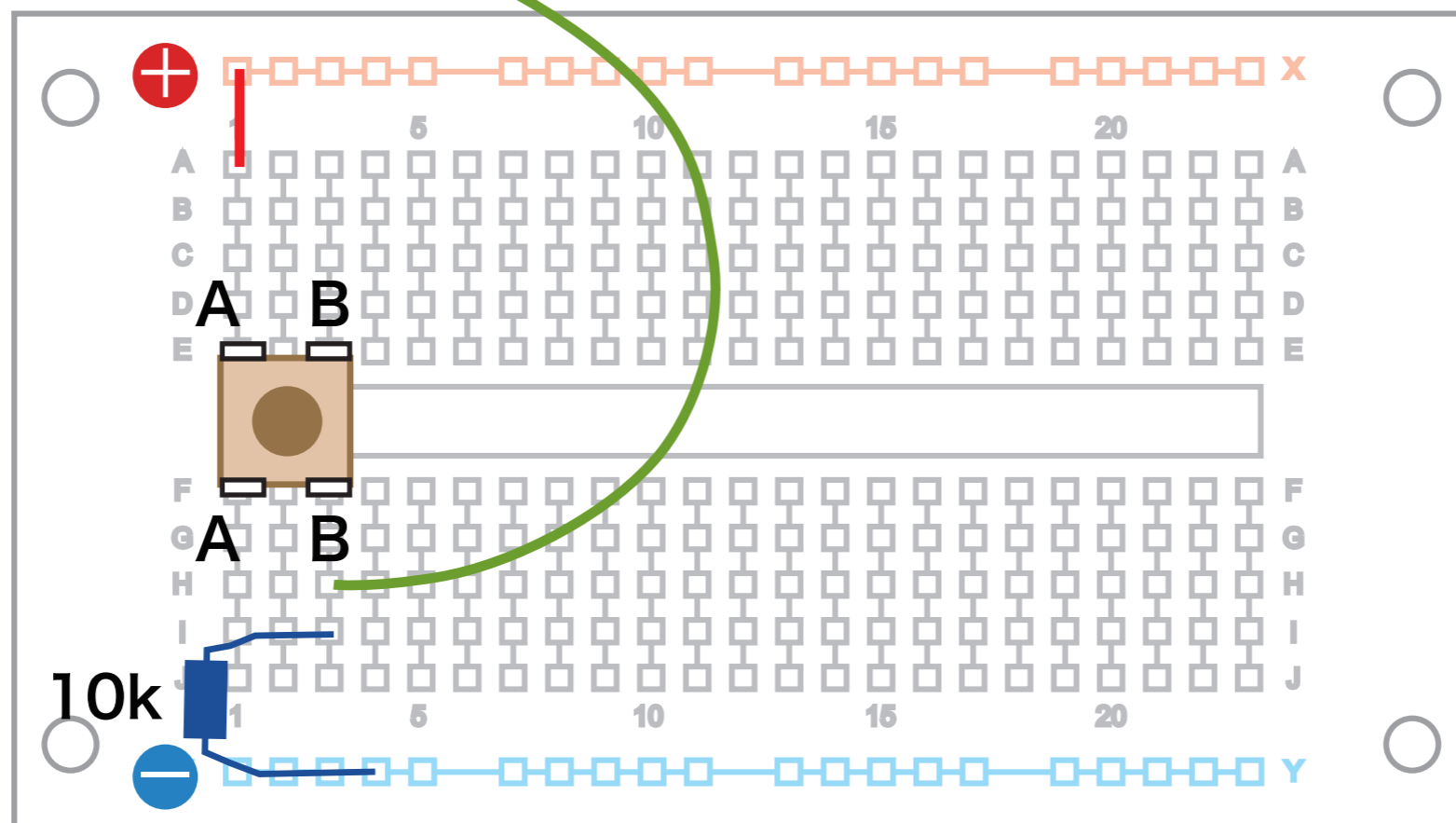


active high vs active low

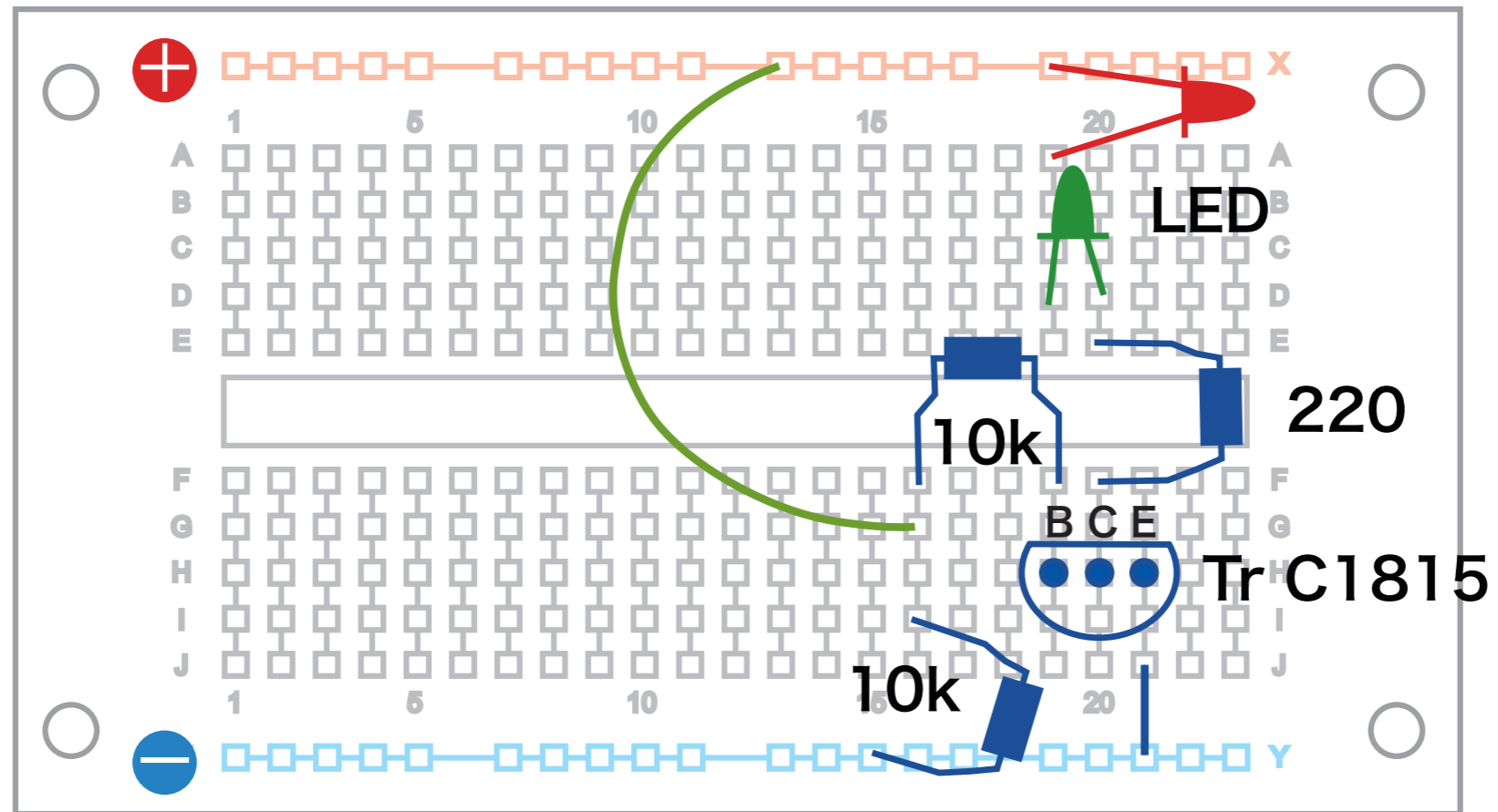


タクトスイッチの接続法：active high

Arduinoのデジタルピンへ

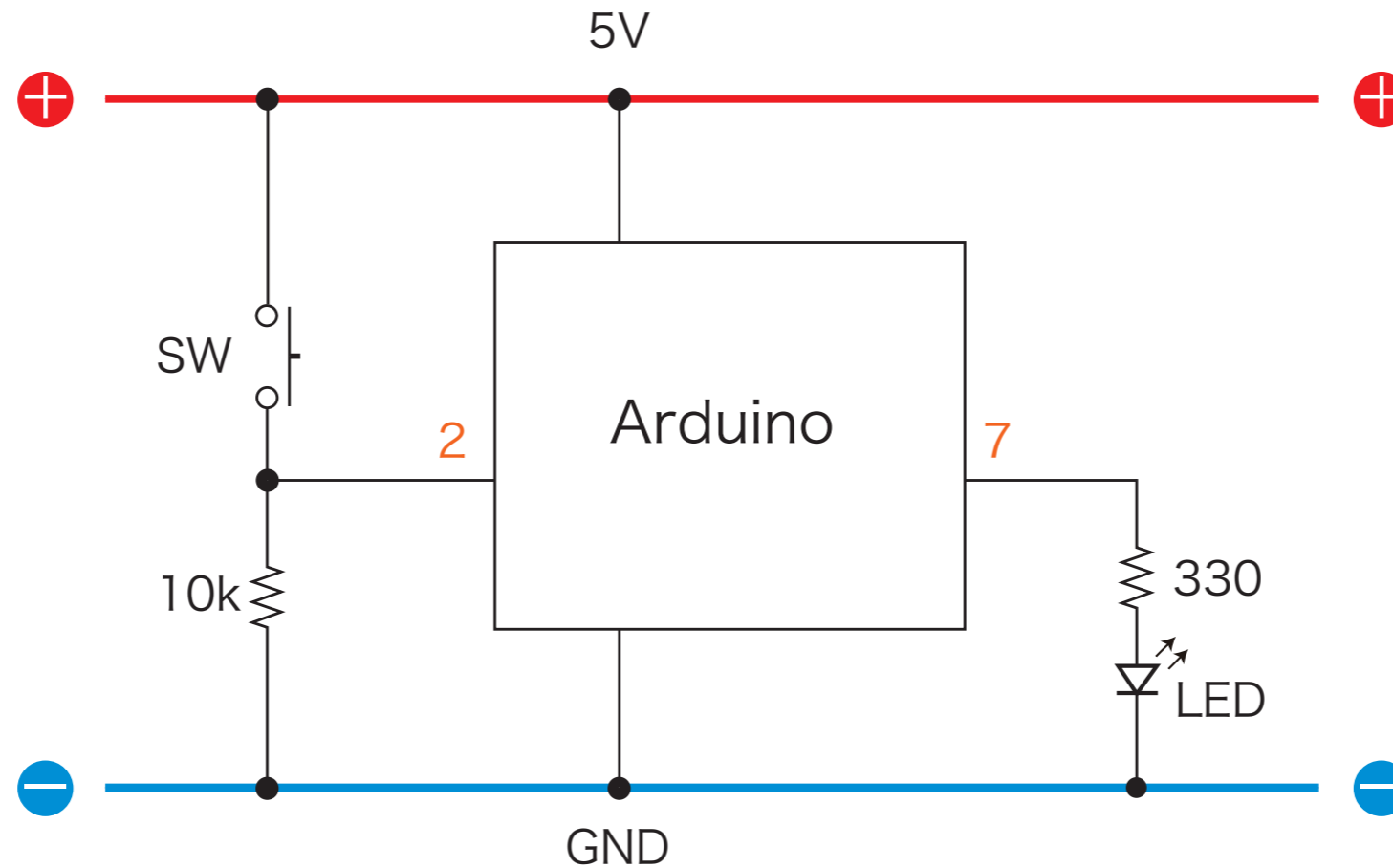


トランジスタ・スイッチを使った点灯



最初に使う回路

最初に使う回路



スイッチを押したらLEDが点灯するプログラムを書く

スイッチを押したらLEDが消灯するプログラムを書く

sketchの例

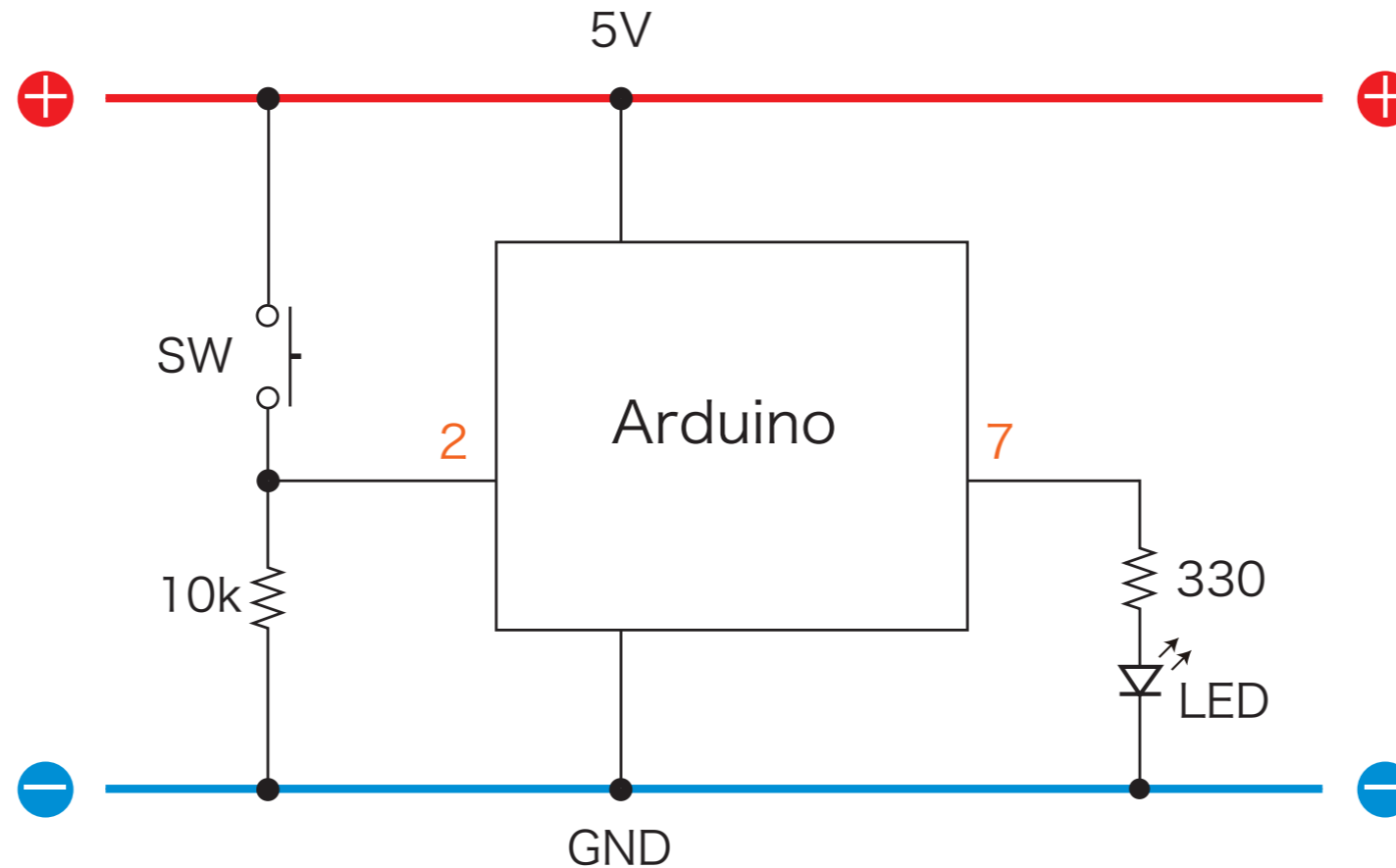
```
// Filename:SW01

const int led=7;
const int sw=2;
int state=LOW;

void setup(){
    pinMode(led, OUTPUT);
    pinMode(sw, INPUT);
}

void loop(){
    state=digitalRead(sw);
    digitalWrite(led, state);
}
```

toggle動作のプログラムを書く



最初にスイッチを押したらLEDが点灯し、
次に押すと消える動作を繰り返すプログラムを書く

どこかおかしいsketchの例

```
// Filename:SW02

const int led=7;
const int sw=2;
int currentState, previousState=LOW;
int ledState=LOW;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(sw, INPUT);
}

void loop(){
  currentState=digitalRead(sw);
  if(currentState == HIGH) {
    ledState = !ledState; // toggle action
    delay(20);
  }
  digitalWrite(led, state);
}
```

```
// Filename:SW03

const int led=7;
const int sw=2;
int currentState, previousState=LOW;
int ledState=LOW;

void setup(){
    pinMode(led, OUTPUT);
    pinMode(sw, INPUT);
}

void loop(){
    currentState=digitalRead(sw);
    if(currentState != previousState){//edge detector
        previousState=currentState;
        if(currentState == HIGH) {
            ledState = !ledState; // toggle action
            digitalWrite(led, ledState);
        }
    }
}
```

高度なテクニック：割り込み

```
// Filename:SW04

const int led=7;
const int sw=2;
volatile int state = LOW;

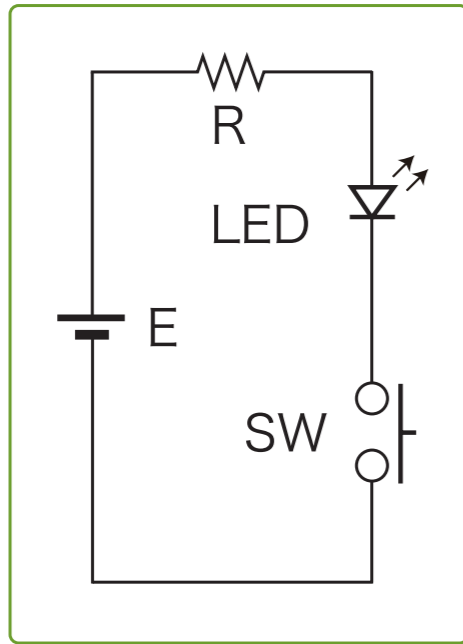
void setup(){
    pinMode(led, OUTPUT);
    digitalWrite(sw, HIGH);
    attachInterrupt(0, blink, RISING);
}

void loop(){
    digitalWrite(led, state);
}

void blink(){
    state = !state;
}
```

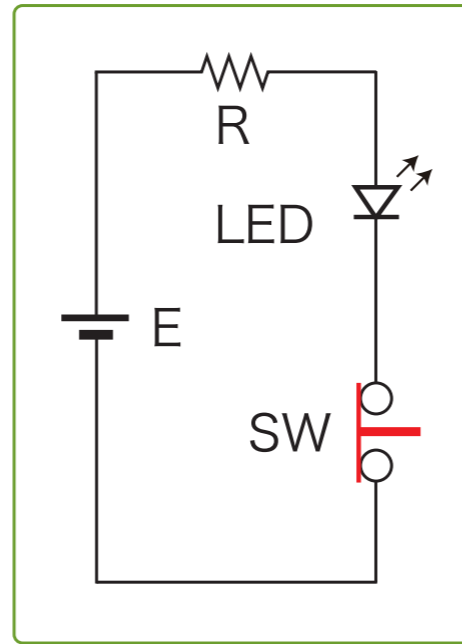
2つのスイッチを使う

スイッチを使ったLEDの点灯回路



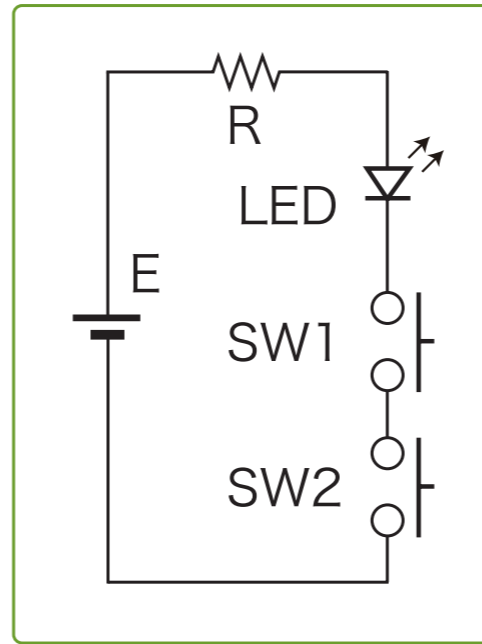
(a)

SW	LED
ON	点灯
OFF	消灯



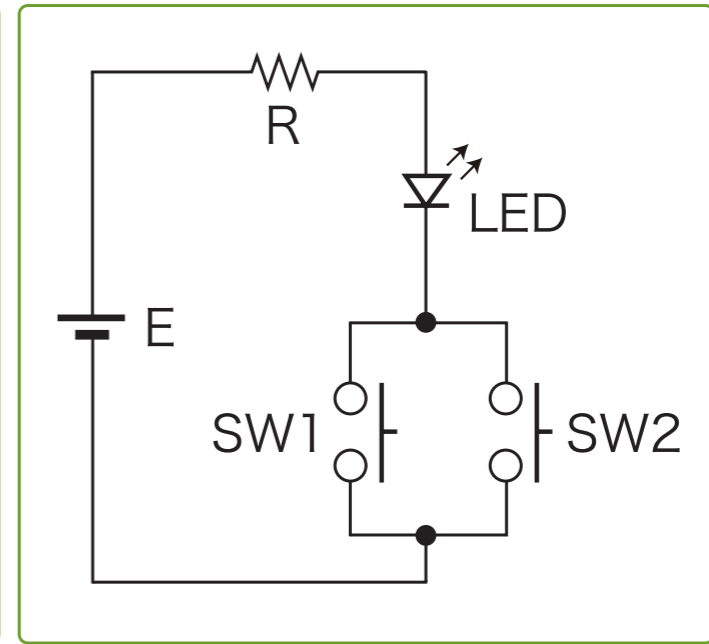
(b)

SW	LED
ON	消灯
OFF	点灯



(c)

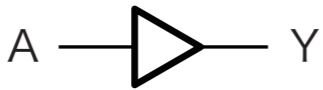

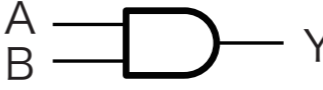
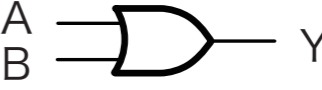
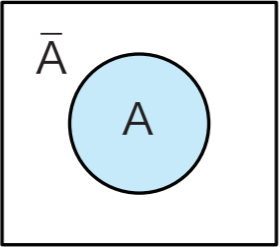
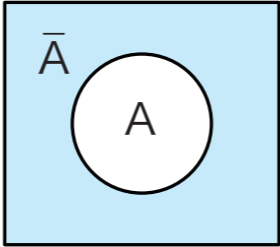
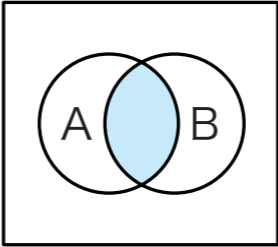
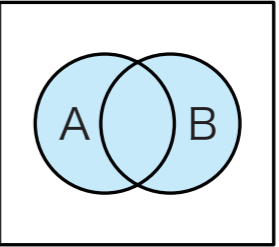
SW1	SW2	LED
OFF	OFF	
OFF	ON	
ON	OFF	
ON	ON	







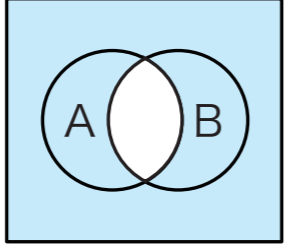
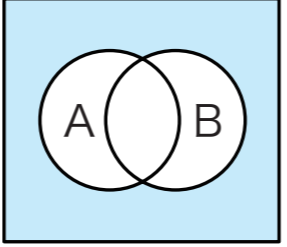
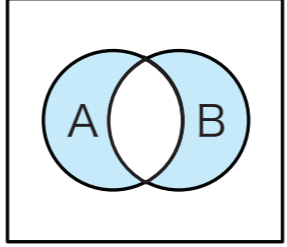
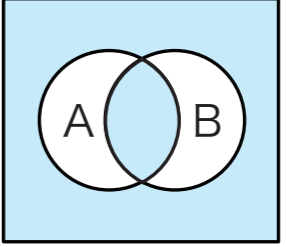
(d)

SW1	SW2	LED
OFF	OFF	
OFF	ON	
ON	OFF	
ON	ON	

基本的な論理演算 1

演算	Buffer (論理肯定)	NOT (論理否定)	AND (論理積)	OR (論理和)																																										
式	$Y = A$	$Y = \bar{A}$	$Y = A \times B$	$Y = A + B$																																										
論理 回路図																																														
真理 値表	<table border="1" data-bbox="653 983 900 1201"> <thead> <tr> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	Y	0	0	1	1	<table border="1" data-bbox="1133 983 1380 1201"> <thead> <tr> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	Y	0	1	1	0	<table border="1" data-bbox="1599 983 1846 1304"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1" data-bbox="2066 983 2313 1304"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1
A	Y																																													
0	0																																													
1	1																																													
A	Y																																													
0	1																																													
1	0																																													
A	B	Y																																												
0	0	0																																												
0	1	0																																												
1	0	0																																												
1	1	1																																												
A	B	Y																																												
0	0	0																																												
0	1	1																																												
1	0	1																																												
1	1	1																																												
ベン図																																														

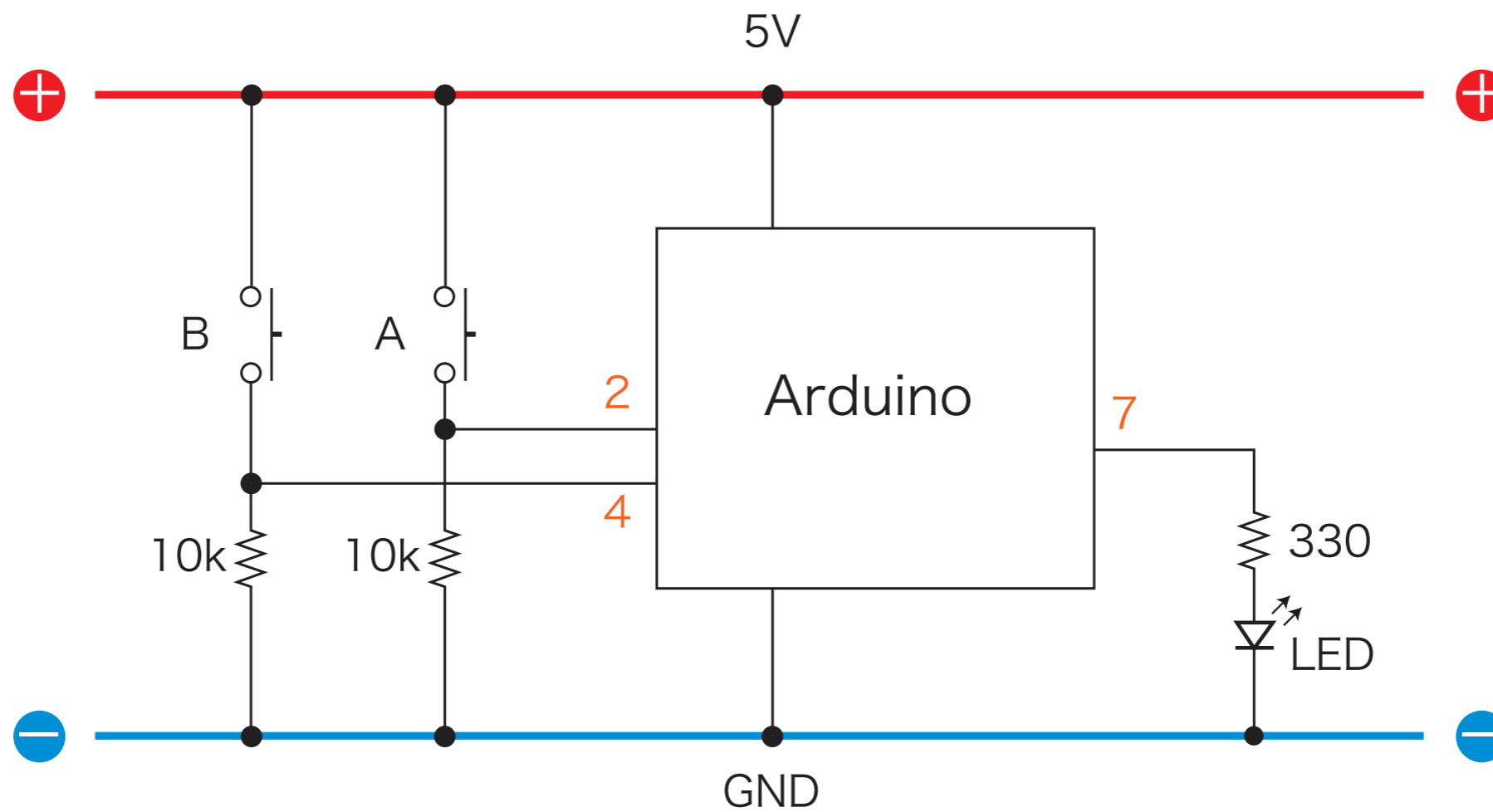
基本的な論理演算 2

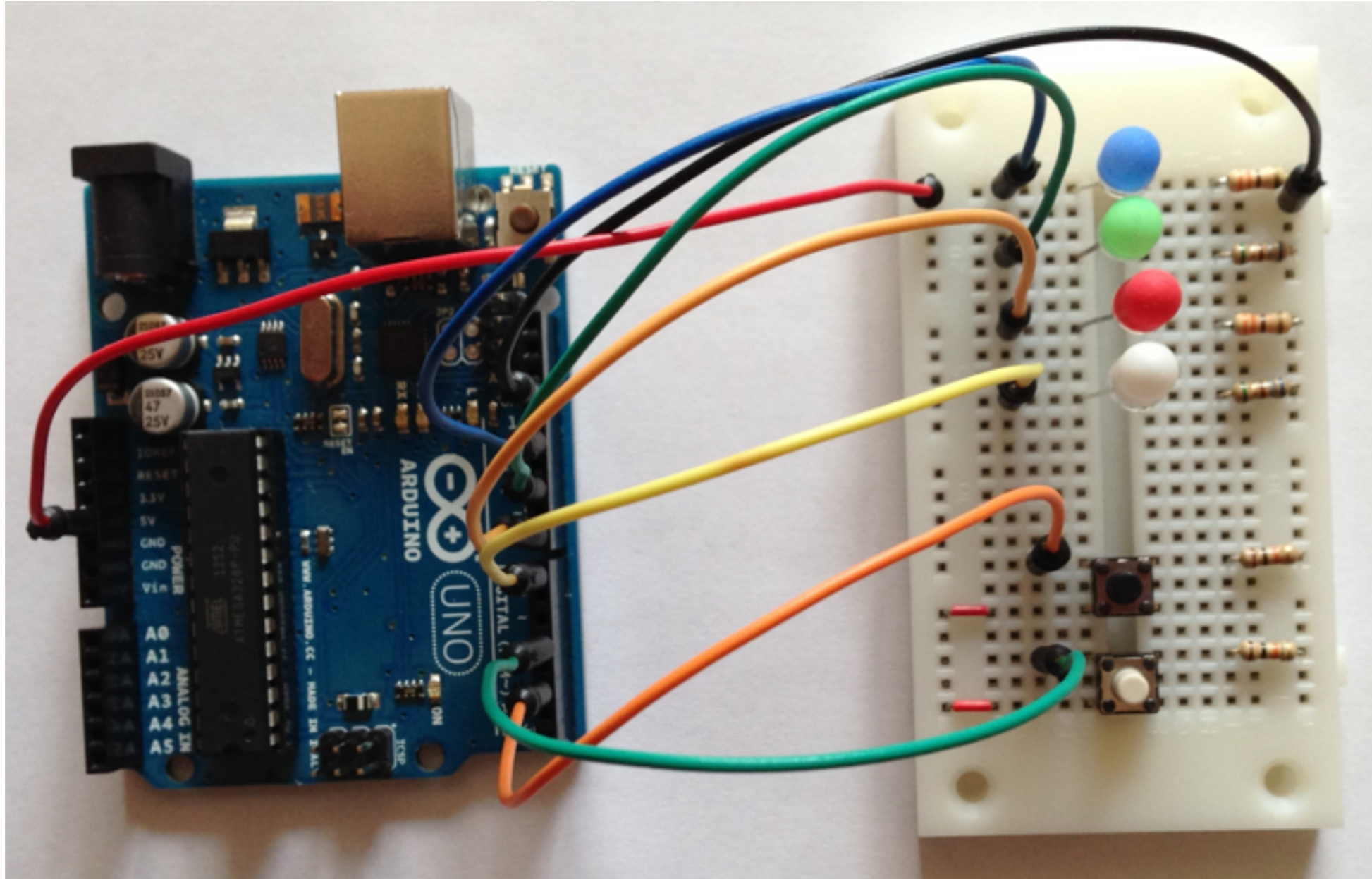
演算	NAND (否定論理積)	NOR (否定論理和)	EX-OR (排他的論理和)	EX-NOR (一致：否定排他和)																																																												
式	$Y = \overline{A \times B}$	$Y = \overline{A + B}$	$Y = A \oplus B$	$Y = \overline{A \oplus B}$																																																												
論理回路図																																																																
真理値表	<table border="1" data-bbox="672 1001 919 1318"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1" data-bbox="1138 1001 1385 1318"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1" data-bbox="1605 1001 1852 1318"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1" data-bbox="2058 1001 2304 1318"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																																																														
0	0	1																																																														
0	1	1																																																														
1	0	1																																																														
1	1	0																																																														
A	B	Y																																																														
0	0	1																																																														
0	1	0																																																														
1	0	0																																																														
1	1	0																																																														
A	B	Y																																																														
0	0	0																																																														
0	1	1																																																														
1	0	1																																																														
1	1	0																																																														
A	B	Y																																																														
0	0	1																																																														
0	1	0																																																														
1	0	0																																																														
1	1	1																																																														
ベン図																																																																

普通の代数と論理代数（ブール代数）の違い

普通の代数		論理代数	
$1 + A = A + 1$	$0 \times A = 0$	$1 + A = 1$	$0 \times A = 0$
$0 + A = A$	$1 \times A = A$	$0 + A = A$	$1 \times A = A$
$A + A = 2A$	$A \times A = A^2$	$A + A = A$	$A \times A = A$
$A + (-A) = 0$	$A \times (-A) = -A^2$	$A + \bar{A} = 1$	$A \times \bar{A} = 0$
$-(-A) = A$		$\overline{\bar{A}} = A$	
$A + B = B + A$	$A \times B = B \times A$	$A + B = B + A$	$A \times B = B \times A$
		ド・モルガンの定理	
		$\overline{A + B} = \bar{A} \times \bar{B}$	$\overline{A \times B} = \bar{A} + \bar{B}$

2つのスイッチを使う：論理プログラム





Cの論理(Boole)演算

(1) OR operation: `||`
`if(A || B){`
 `analogWrite(ledPin, HIGH);`
`}`

(2) AND operation: `&&`
`if(A && B){`
 `analogWrite(ledPin, HIGH);`
`}`

(3) NOT operation: `!`
`if(!A){`
 `analogWrite(ledPin, HIGH);`
`}`

注意：よく似た演算に**ビット**演算なるものがある。

| (OR), & (AND), ~ (NOT), ^ (XOR), <<, >> (shift)

論理和で点灯させる

```
int led = 7;
int swA = 2, swB=4;
int stateA = 0, stateB=0;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(swA, INPUT);
  pinMode(swB, INPUT);
}

void loop(){
  stateA = digitalRead(swA);
  stateB = digitalRead(swB);
  if(stateA || stateB){
    digitalWrite(led, HIGH);
  }else{
    digitalWrite(led, LOW);
  }
}
```

論理積で点灯させる

```
int led = 7;
int swA = 2, swB=4;
int stateA = 0, stateB=0;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(swA, INPUT);
  pinMode(swB, INPUT);
}

void loop(){
  stateA = digitalRead(swA);
  stateB = digitalRead(swB);
  if(stateA && stateB){
    digitalWrite(led, HIGH);
  }else{
    digitalWrite(led, LOW);
  }
}
```


排他的論理和(EX-OR)で点灯させる

```
int led = 7;
int swA = 2, swB=4;
int stateA = 0, stateB=0;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(swA, INPUT);
  pinMode(swB, INPUT);
}

void loop(){
  stateA = digitalRead(swA);
  stateB = digitalRead(swB);
  if((!stateA && stateB) || (stateA && !stateB)){
    digitalWrite(led, HIGH);
  }else{
    digitalWrite(led, LOW);
  }
}
```

お疲れさまでした