

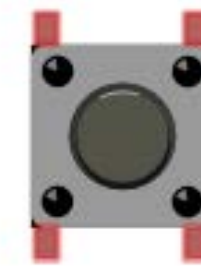
DOCUMENTATION

ACTIVITY

ASSESSMENT

USEFUL LINKS

HOW TO USE A PUSH BUTTON TO TURN ON AN LED WITH ARDUINO



PUSH BUTTON



LED

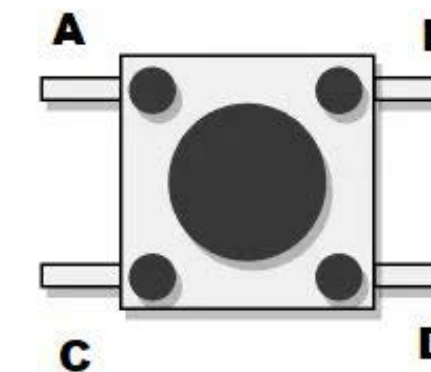
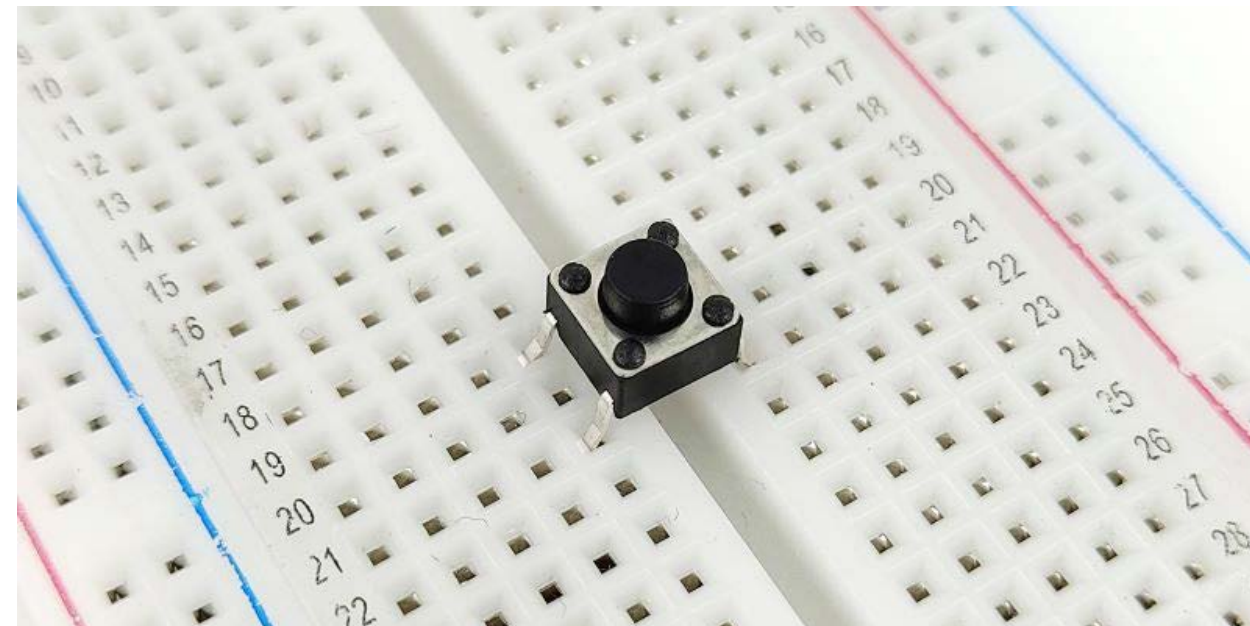
DOCUMENTATION

ACTIVITY

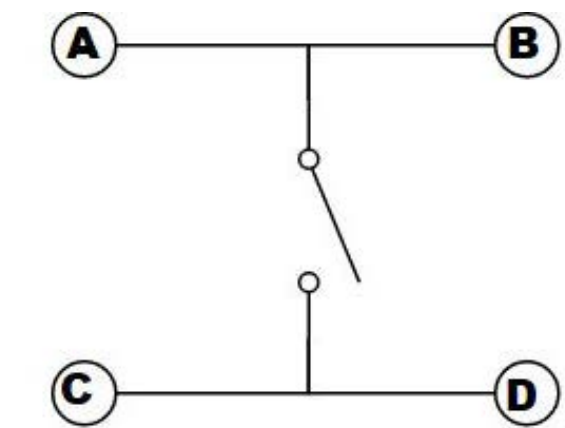
ASSESSMENT

USEFUL LINKS

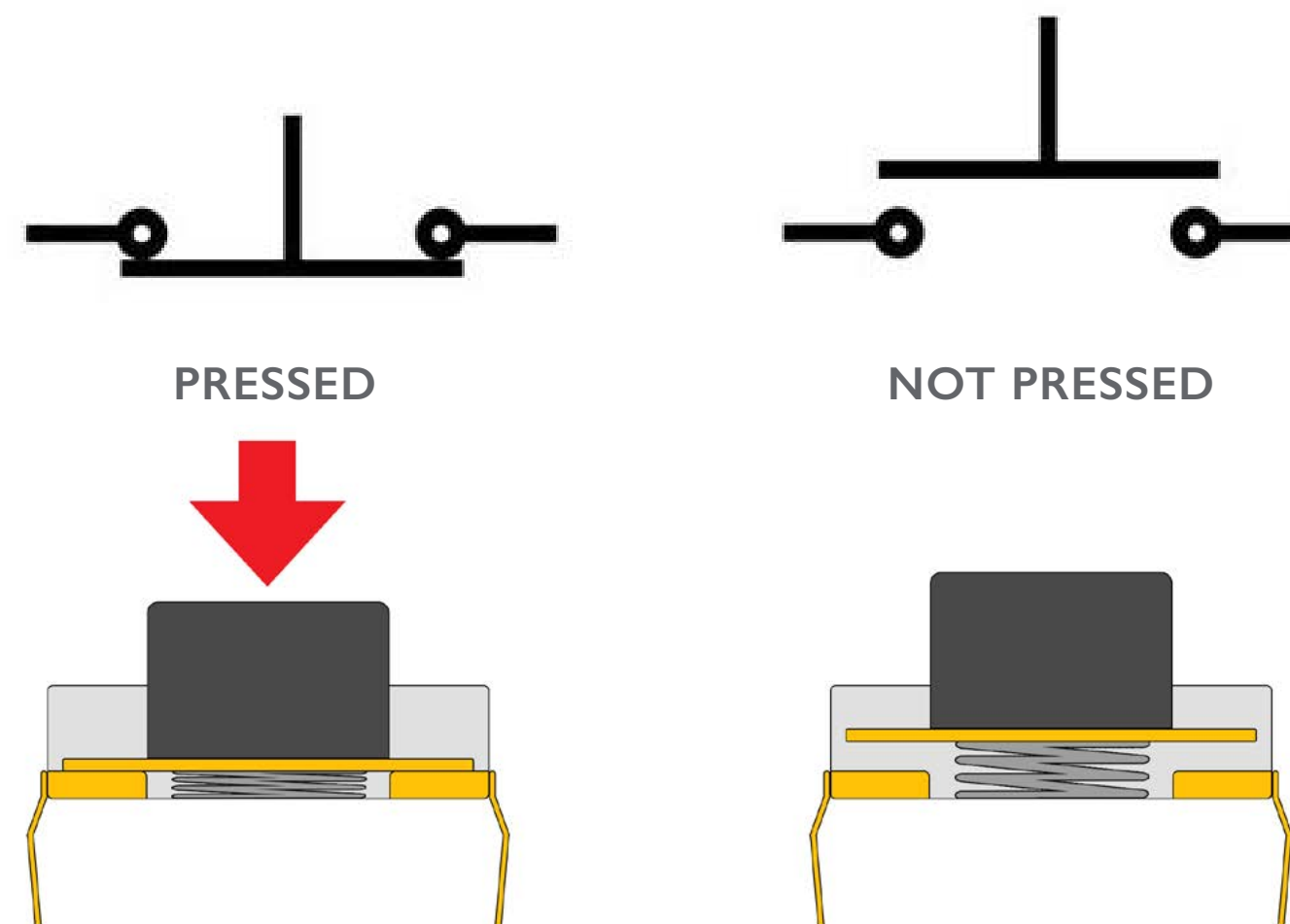
PUSH BUTTON



Channel 1: pins A-B are internally connected



Channel 2: pins C-D are internally connected



HOW DOES IT WORK?

Buttons open and close the flow of electricity in a circuit. When the button is pressed, we connect the button's internal pins (A-B to C-D) creating a bridge that allows the flow of electricity. While the button is not pressed, the connection is broken, therefore, electricity cannot pass through the circuit..

WARNING! The choice of pins does matter. Make sure you are connecting the correct pins.

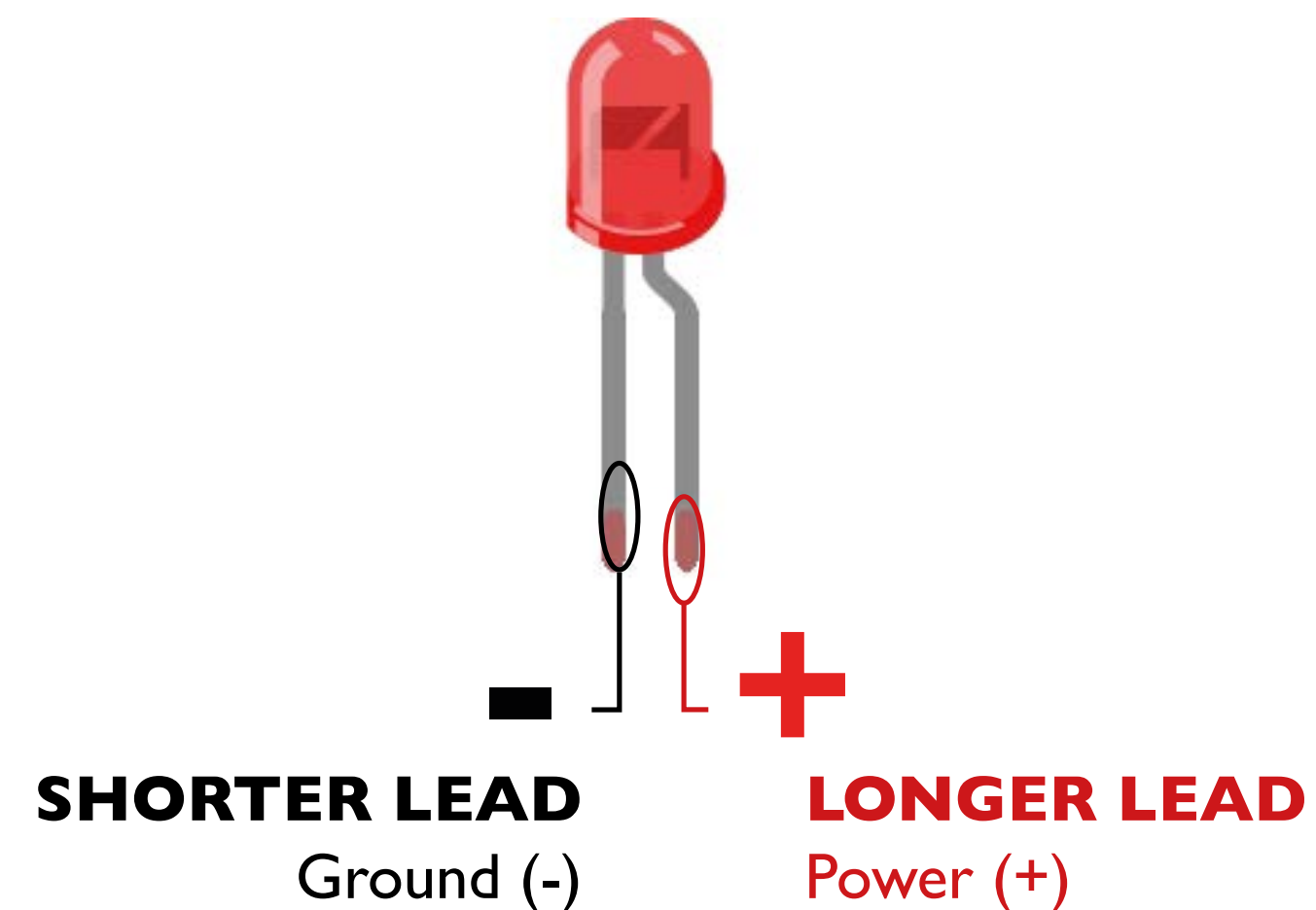
DOCUMENTATION

ACTIVITY

ASSESSMENT

USEFUL LINKS

LED (Light-Emitting Diode)



HOW DOES IT WORK?

LEDs are electronic components (diodes) that emit light when electricity flows through them.

WARNING! Polarity Matters

LEDs are polarized components, meaning electricity flows only in one direction—from positive to negative.

The longer lead must be connected to the positive side of the circuit (e.g., a data pin from the Arduino). You a resistor!

The shorter lead must be connected to ground (GND).

WARNING! Use a Resistor

Arduino pins output 5V—too much for an LED. Use a 220Ω resistor to prevent it from burning out.

Control an LED using a push-button

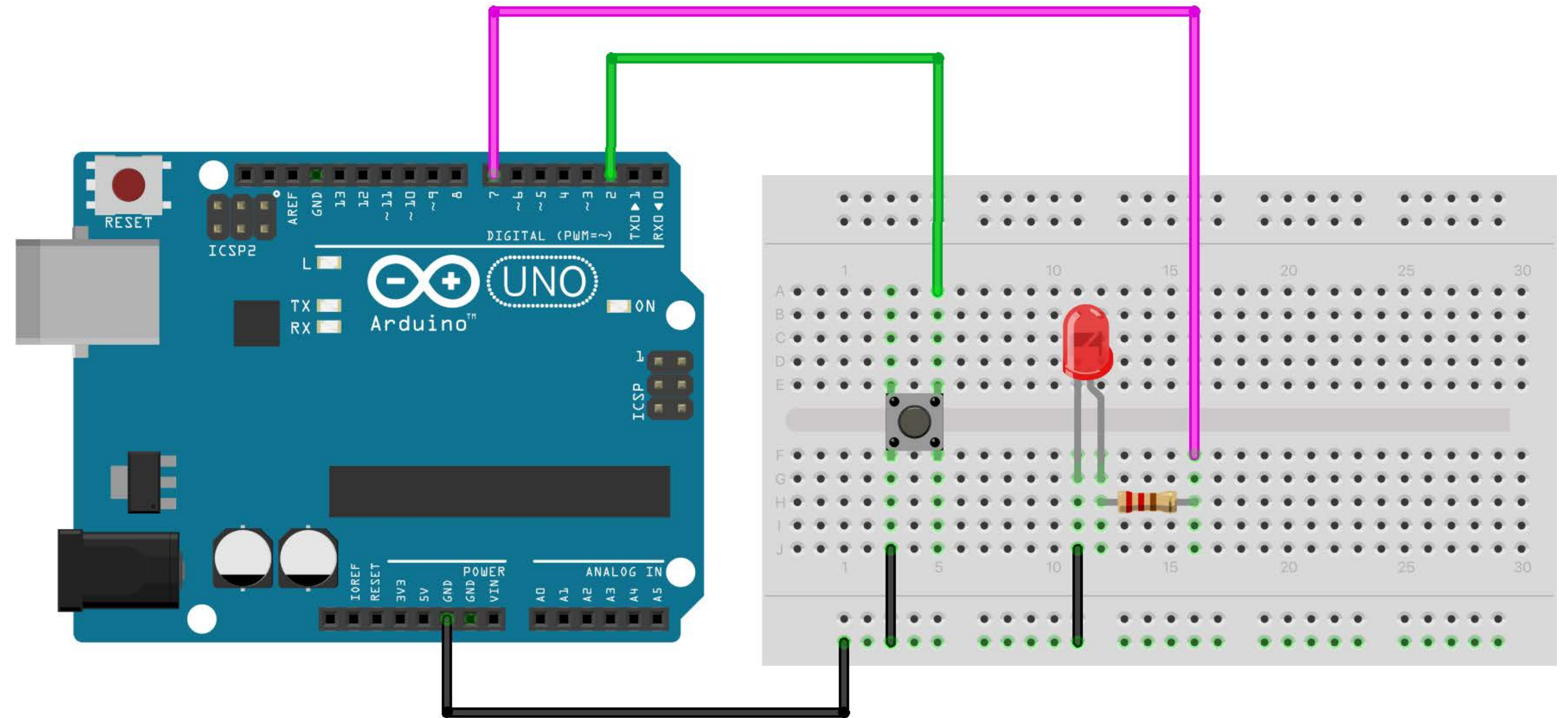
DOCUMENTATION

ACTIVITY

ASSESSMENT

USEFUL LINKS

DIAGRAM



fritzing

CODE

[DOWNLOAD](#)

DOCUMENTATION

ACTIVITY

ASSESSMENT

USEFUL LINKS

SERIAL MONITOR: Let's analyse the data

Initiate
Serial
Communication

Print 'PRESSED'
if the button is being
pressed.

Print 'NOT PRESSED'
if the button is not
being pressed.

CLICK HERE to open
the Serial Monitor

Read data coming from
the Button

SERIAL MONITOR

WHAT DATA IS PRINTED
WHEN YOU PRESS OR
RELEASE THE BUTTON?

?

The screenshot shows the Arduino IDE 2.3.6 interface. The main window displays the code for 'Button_LED.ino'. The code is as follows:

```
1  const int buttonPin = 2;
2  const int ledPin = 7;
3  int buttonState = 0;
4
5  void setup() {
6    Serial.begin(9600);
7    pinMode(ledPin, OUTPUT);
8    pinMode(buttonPin, INPUT_PULLUP);
9  }
10
11 void loop() {
12   buttonState = digitalRead(buttonPin);
13
14   // Check if the button is pressed
15   if (buttonState == LOW) { //(Pressed = LOW due to pull-up)
16
17     digitalWrite(ledPin, HIGH); // Turn the LED on
18     Serial.println("pressed");
19
20   } else {
21     digitalWrite(ledPin, LOW); // Turn the LED off
22     Serial.println("not pressed");
23   }
24 }
25
```

The Serial Monitor window is open at the bottom, showing the 'Output' tab. It has a text input field for messages, a 'New Line' dropdown, and a '9600 baud' dropdown. The Serial Monitor is currently empty.

DOCUMENTATION

ACTIVITY

ASSESSMENT

USEFUL LINKS

CHALLENGE:

1. Build a circuit with two buttons and two LEDs:
 - Button A controls LED A
 - Button B controls LED B
 - Each LED should light only while its button is pressed.
 - Change the messages written in the Serial Monitor
2. Upload a video (max 10 seconds) showing your achievements.

VIDEO UPLOAD

DOCUMENTATION

ACTIVITY

ASSESSMENT

USEFUL LINKS

FURTHER LEARNING

- [Push-Button more tutorials](#)
- [Understanfing Pull-up and Pull-Down](#)
- [How to connect a push button or switch](#)