# Practical 1

Controlling the Light Emitting Diode (LED) with a push button.

Introduction: Push-button is a very simple mechanism which is used to control electronic signal either by blocking it or allowing it to pass. This happens when mechanical pressure is applied to connect two points of the switch together. Push buttons or switches connect two points in a circuit when pressed. When the push-button is released, there is no connection between the two legs of the push-button. Here it turns on the built-in LED on pin 11 when the button is pressed. The LED stays ON as long as the button is being pressed.
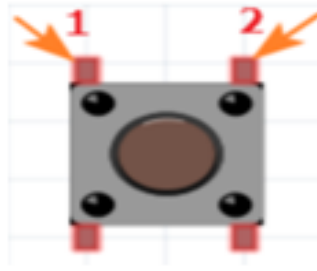
LED Specifications
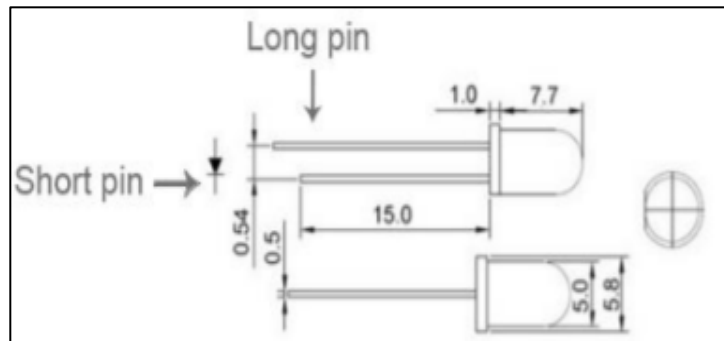
Pin definition

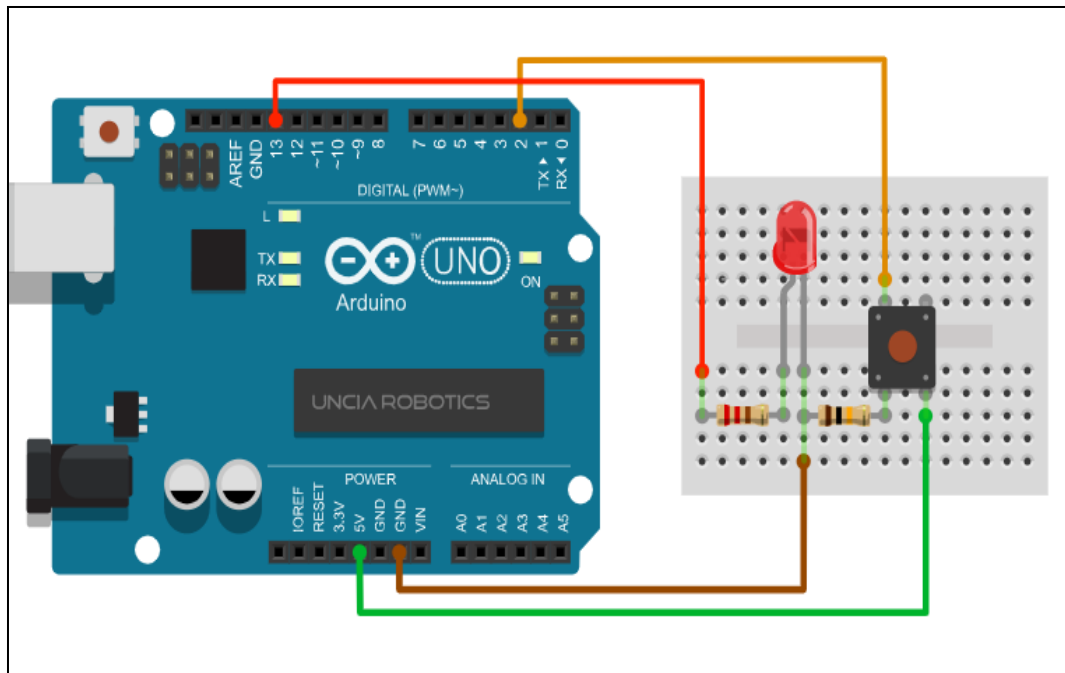| Long pin	+5V
Short pin	GND

Push Button

Specifications:

Size	6 x 6 x 5mm
Temperature	-30 ~ +70 Centigrade

Hardware Required:

Component Name	Quantity
Arduino UNO	1
LED	1
Push Button	1
220Ω resistor	1
10KΩ resistor	1
USB Cable	1

Breadboard	1
Jumper wires	Several

Connection Diagram:

Steps of working

1. Insert the push button into your breadboard and connect it to the digital pin 7(D7) which act as INPUT.
2. Insert the LED into the breadboard. Attach the positive leg (the longer leg) to digital pin 11 of the Arduino Uno, and the negative leg via the 220-ohm resistor to GND. The pin D11 is taken as OUTPUT.
3. The 10kΩ resistor used as PULL-UP resistor and 220 Ω resistors is used to limit the current through the LED.
4. Upload the code as given below.

5. Press the push-button to control the ON state of LED.

The Sketch

> This sketch works by setting pin D7 as for the push button as INPUT and pin 11 as an OUTPUT to power the LED.
> The initial state of the button is set to OFF.
> After that the run a loop that continually reads the state from the pushbutton and sends that value as voltage to the LED. The LED will be ON accordingly.

/**************Pressing Button LED*****/**

```
const int buttonPin = 7; // choose the pin for the pushbutton
const int ledPin = 11;  // choose the pin for a LED
int buttonState = 0;   // variable for reading the pushbutton pin status
void setup()
{
 pinMode(ledPin, OUTPUT); // declare LED as output
 pinMode(buttonPin, INPUT);   // declare pushbutton as input
}
void loop()
{
 buttonState = digitalRead(button Pin); // read input value
 if (buttonState == HIGH)
{      // check if the input is HIGH (button pressed)
 digitalWrite(ledPin, HIGH); // turn LED ON
```

```
   }
  else
  {
  digitalWrite(ledPin, LOW);  // turn LED OFF}}
```

Observation Table:

Sr no.	Push button State	LED State
1		
2		

Precautions:

1. The pushbutton is square so it is important to set it appropriately on breadboard.

2. While making the connections make sure to use a pull-down resistor because directly connecting two points of a switch to the circuit will leave the input pin in floating condition and circuit may not work according to the program.

3. It is very important to set pinMode() as OUTPUT first before using digitalWrite() function on that pin.

4. If you do not set the pinMode() to OUTPUT, and connect an LED to a pin, when calling digitalWrite(HIGH), the LED may appear dim.