

REF: C11-PBMOD

Push Button Module with Big Cap



Description

This push button module is a simple, efficient, and versatile component designed for easy integration into your electronics projects. It is compatible with a wide range of popular microcontroller platforms such as Arduino, NodeMCU, Wemos, and Raspberry Pi, making it a perfect addition for both beginners and experienced makers.

Specifications

- Push Button Module
- Material: Plastic
- Cap Color: Red
- Interface: Standard electronic block (GND, VCC, SIG)
- Working voltage: 2.0-5.5VDC
- Working Current: 0.55mA (MAX)
- Output: Digital Level (Press High, Release Low)

Pin connection

Arduino UNO (5V) → Push Button (VCC):

Connect the 5V pin of the Arduino to the VCC pin of the push button. This provides power to the push button.

Arduino UNO (GND) → Push Button (GND):

Connect the GND pin of the Arduino to the GND pin of the push button. This serves as the ground connection.

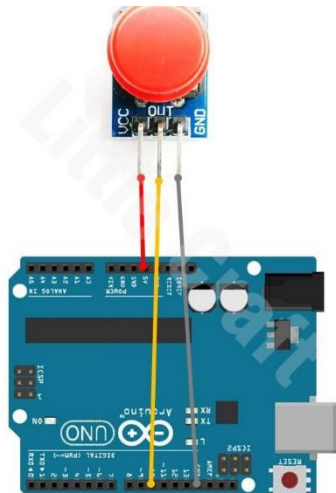
Arduino UNO (D10) → Push Button (SIG):

Connect the digital pin D10 of the Arduino to the SIG (signal) pin of the push button. This pin will be used to read the button's state in the program.

Arduino UNO (PIN)	Push Button (PIN)
5V	VCC
GND	GND
SIG	D10

Circuit diagram

To connect a push button to an Arduino Uno, connect the VCC pin of the push button to the 5V pin on the Arduino for power, and connect the GND pin of the button to the GND pin on the Arduino to establish the ground connection. Then, connect the SIG (signal) pin of the push button to digital pin D10 on the Arduino to read the button's state. If using a button with an internal pull-up resistor, you typically won't need an external resistor. When the button is pressed, the digital pin will read LOW because it connects to ground, and when the button is release



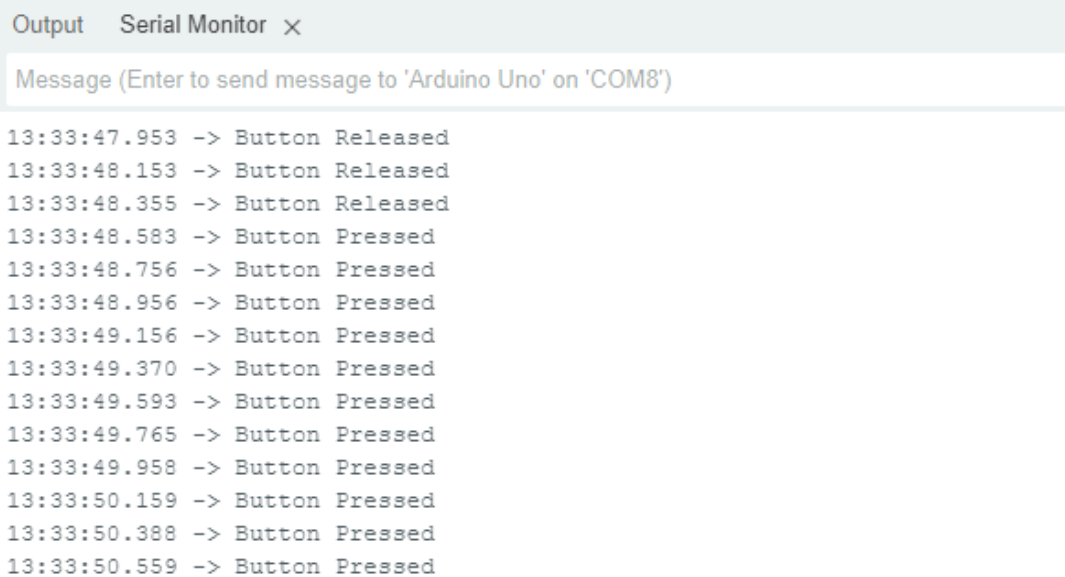
Code push button

codepushbuttonwithbigcap.ino

```
1  const int buttonPin = 10; // Define the pin connected to OUT
2  int buttonState = 0;      // Variable to store button state
3
4  void setup() {
5      Serial.begin(9600); // Start serial communication
6      pinMode(buttonPin, INPUT); // Set button pin as input
7  }
8
9  void loop() {
10     buttonState = digitalRead(buttonPin); // Read button state
11
12     if (buttonState == HIGH) { // If button is pressed
13         Serial.println("Button Pressed");
14     } else {
15         Serial.println("Button Released");
16     }
17
18     delay(200); // Small delay to prevent flickering
19 }
```

Result

Here's what the output looks like on the serial monitor



```
Output  Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM8')
13:33:47.953 -> Button Released
13:33:48.153 -> Button Released
13:33:48.355 -> Button Released
13:33:48.583 -> Button Pressed
13:33:48.756 -> Button Pressed
13:33:48.956 -> Button Pressed
13:33:49.156 -> Button Pressed
13:33:49.370 -> Button Pressed
13:33:49.593 -> Button Pressed
13:33:49.765 -> Button Pressed
13:33:49.958 -> Button Pressed
13:33:50.159 -> Button Pressed
13:33:50.388 -> Button Pressed
13:33:50.559 -> Button Pressed
```