**Đếm số lần nhấn một button - ButtonStateChange**

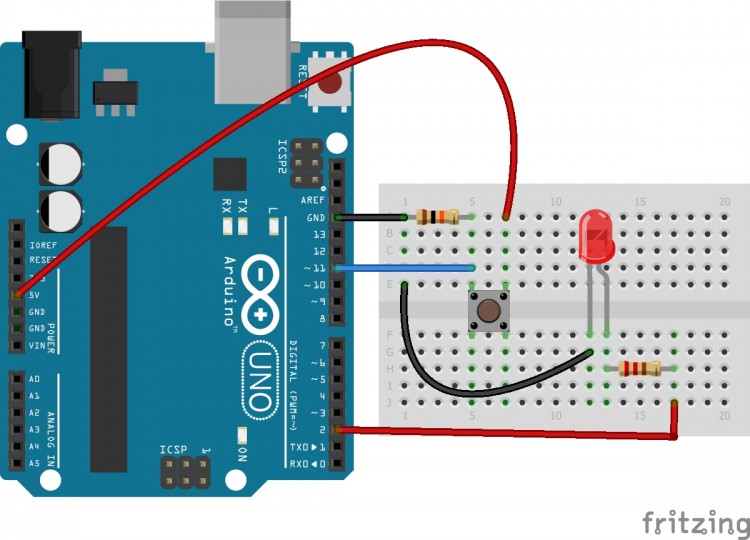
**Nội dung chính, cần nắm**

Nếu bạn đã đọc qua: [Xác định trạng thái của một nút nhấn (button)](http://arduino.vn/bai-viet/80-bai-3-xac-dinh-trang-thai-cua-mot-nut-nhan-button) thì chắc hẳn bạn đã biết cách sử dụng một button. Nhưng đôi khi bạn muốn button của bạn đặc biệt hơn một tí, chẳng hạn như là nhấn vài ba lần thì mới thực hiện chức năng của nó. Để làm được điều này, bạn cần biết được lúc nào button được nhấn và lúc nào button được thả ra, và đếm số lần. Hôm nay, tôi sẽ hướng dẫn bạn giải quyết vấn đề này. Ở nước ngoài, người ta gọi vấn đề này là **state change detection** hoặc **edge detection**[.](http://arduino.vn/bai-viet/157-bai-10-dem-so-lan-nhan-mot-button-buttonstatechange)

**Phần cứng**

* Arduino Uno
* Breadboard
* Dây cắm breadboard
* 1 điện trở 220Ohm
* 1 điện trở 10Kohm
* 1 đèn LED siêu sáng
* 1 button( nút nhấn)

**Lắp mạch**

[](http://k3.arduino.vn/img/2014/06/09/0/527_8121-1402280929-0-led-button-bai-8-bb.jpg)

**Lập trình và giải thích**

1. // dùng const đặt trước khi khai báo biến sẽ coi biến này là một hằng số
2. // Bạn có thể đọc được giá trị biến hoặc thực hiện các phép toán
3. // Nhưng không thay đổi được giá trị của hằng số này.
4. const int buttonPin = 11;// hằng số buttonPin mang giá trị là chân digital được nối với button
5. const int ledPin = 2; // hằng số ledPin mang giá trị là chân digital đươc nối với led
7. // Các biến này có thể thay đổi giá trị được
8. int buttonPushCounter = 0; // số lần button được nhấn
9. int buttonState = 0; // trạng thái hiện tại của button
10. int lastButtonState = 0; // trạng thái trước đó của button
12. void setup() {
13. pinMode(buttonPin, INPUT); // Cài đặt button là INPUT
14. pinMode(ledPin, OUTPUT); // Cài đặt đèn LED là OUTPUT
16. Serial.begin(9600); //Bật cổng Serial ở baudrate 9600
17. }

20. void loop() {
21. // đọc giá trị hiện tại của button
22. buttonState = digitalRead(buttonPin);
24. // so sánh với giá trị trước đó
25. if (buttonState != lastButtonState) {
26. if (buttonState == HIGH) {
27. // Nếu trạng thái bây giờ là button đang được nhấn
28. // thì hiển nhiên trước đó là button chưa được nhấn (điều kiện trên)
29. // chúng ta sẽ tăng số lần nhấn button lên 1
30. buttonPushCounter++;
31. Serial.println("Dang nhan");
32. Serial.print("So lan nhan button la: ");
33. Serial.println(buttonPushCounter);
34. }
35. else {
36. // Nếu trạng thái bây giờ là button đang được THẢ
37. // thì hiển nhiên trước đó là button đang được nhấn (điều kiện trên)
38. // Chúng ta sẽ thông báo là button đang được thả và không làm gì cả
39. Serial.println("off");
40. }
41. }
42. // lưu lại trạng thái button cho lần kiểm tra tiếp theo
43. lastButtonState = buttonState;
45. // Đã đếm được số lần nhấn button, bây giờ sẽ là phần sau bao nhiêu
46. // lần thì button sẽ làm đèn sáng
47. // Trong ngôn ngữ lập trình Arduino, chúng ta có thêm một phép toán mới
48. // đó là phép chia lấy dư (khác với các phép +, -, \*, / được học trong trường
49. // phép này sẽ trả về phần dư của một phép chia.
50. // Ví dụ: 6 % 4 = 2 (% là toán tử) (vì 6 chia 4 = 1 dư 2). Xem thêm tại: http://arduino.vn/reference/modulo
51. // Áp dụng:
52. // Chẳng hạn, bạn làm button này cứ sau 4 lần bấm sẽ làm đèn led sáng vì bạn làm như sau:
53. // Mới upload code thì đèn sáng do buttonPushCounter = 0. 0 % 4 = 0
54. // Sau đó cứ mỗi lần nhấn nút thì buttonPushCounter được tăng lên.
55. //... 1 % 4 = 1 --> tắt
56. //... 2 % 4 = 2 --> tắt
57. //... 3 % 4 = 3 --> tắt
58. //... 4 % 4 = 0 --> bật
59. //... 5 % 4 = 1 --> tắt
60. //...
61. if (buttonPushCounter % 4 == 0) {
62. digitalWrite(ledPin, HIGH);
63. Serial.println("Da bat den");
64. } else {
65. digitalWrite(ledPin, LOW);
66. }
68. }