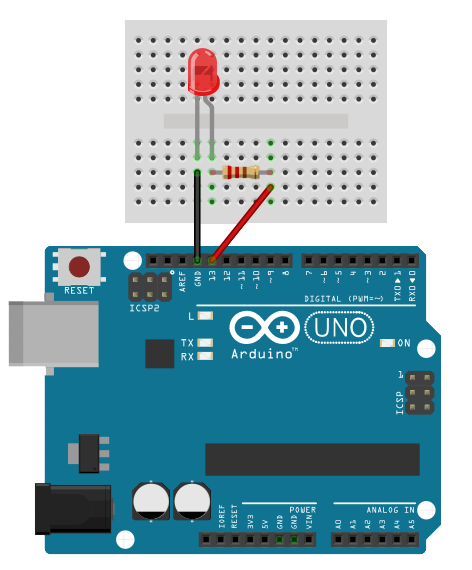
## Cách làm đèn LED nhấp nháy theo yêu cầu

## 1.Phần cứng

* Arduino UNO R3
* 01 LED 5mm
* 01 điện trở 330 Ohm

## 2.Lắp mạch

## Như hình vẽ



**3. Mã lập trình và giải thích**

Trước tiên, cứ mỗi khi dùng một con LED, chúng ta phải ***pinMode OUTPUT*** chân **Digital** mà ta sử dụng cho con đèn LED. Trong bài này, chúng ta sử dụng chân LED là chân digital 13. Nên đoạn code sau cần nằm trong ***voidsetup()***

**pinMode(13, OUTPUT);**

Để bật một con đèn LED, bạn phải ***digitalWrite HIGH*** cho chân số 13 (chân Digtal được kết nối với con LED). Đoạn code này nằm trong ***voidloop()***

**digitalWrite(13,HIGH);**

Dòng lệnh trên sẽ cấp một điện thế là 5V vào chân số Digital 13. Điện thế sẽ đi qua điện trở 330 Ohm rồi đến đèn LED (sẽ làm nó sáng mà không bị cháy, ngoài ra bạn có thể các loại điện trở khác <= 10kOhm). Để tắt một đèn LED, bạn sử dụng hàm:

**digitalWrite(13,LOW);**

Lúc này điện thế tại chân 13 sẽ là 0 V => đèn LED tắt. Và để thấy được trạng thái bật và tắt của đèn LED bạn phải dừng chương trình trong một khoảng thời gian đủ lâu để mắt cảm nhận được (nói vậy thôi, chứ bạn chỉ cần dừng chương trình trong vài miligiây là thấy được rồi). Vì vậy, hàm ***delay*** được tạo ra để làm việc này (Dừng hẳn chương trình bao nhiêu mili giây)!

Sau đây là đoạn code full

1. /\*
2. Blink - Nhấp nháy
3. Đoạn code làm nhấp nháy một đèn LED cho trước
4. \*/
6. // chân digital 13 cần được kết nối với đèn LED
7. // và chân digital 13 này sẽ được đặt tên là 'led'. Biến 'led' này có kiểu dữ liệu là int và có giá trị là 13
8. int led = 13;
10. // Hàm setup chạy một lần duy nhất khi khởi động chương trình
11. void setup() {
12. // đặt 'led' là OUTPUT
13. pinMode(led, OUTPUT);
14. }
16. // Hàm loop chạy mãi mãi sau khi kết thúc hàm setup()
17. void loop() {
18. digitalWrite(led, HIGH); // bật đèn led sáng
19. delay(1000); // dừng chương trình trong 1 giây => thây đèn sáng được 1 giây
20. digitalWrite(led, LOW); // tắt đèn led
21. delay(1000); // dừng chương trình trong 1 giây => thấy đèn tối được 1 giây
22. }