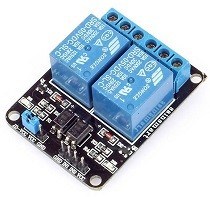
**Giới thiệu Mô-đun rơle (re-lay)**

Rơle là một công tắc hoạt động bằng điện dân dụng. Có nghĩa là nó có thể được bật hoặc tắt, cho phép dòng điện đi qua hay không.

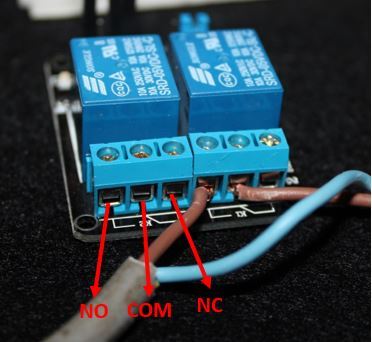
Điều khiển rơle bằng Arduino cũng đơn giản như điều khiển đầu ra như đèn LED.Một loại mô-đun role trong hình dưới đây.



Mô-đun này có hai kênh (những hình khối màu xanh). Có những điểm giống/ khác với một, bốn và tám kênh.

## ****Kết nối điện áp chính****

Liên quan đến điện áp nguồn, rơle có 3 kết nối:

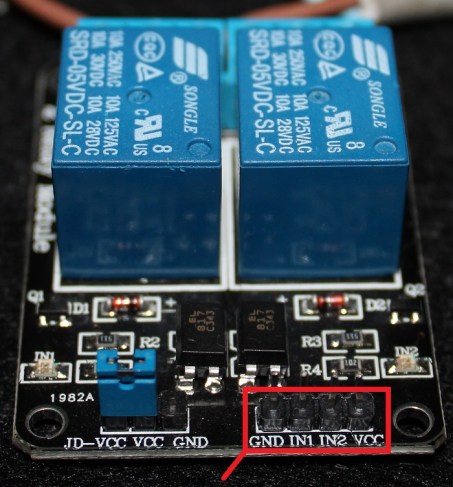


* **COM**: chân chung
* **NO** (Thường mở): không có tiếp xúc giữa chân chung và chân thường mở. Vì vậy, khi bạn kích hoạt rơle, nó kết nối với chân COM và nguồn cung cấp được cung cấp cho tải
* **NC**(Thường đóng): có tiếp xúc giữa chân chung và chân thường đóng. Luôn có kết nối giữa các chân COM và NC, ngay cả khi tắt rơle. Khi bạn kích hoạt rơle, mạch được mở và không có nguồn cung cấp cho tải.

Nếu bạn muốn điều khiển đèn chẳng hạn, tốt hơn là sử dụng mạch mở thông thường, bởi vì chúng tôi chỉ muốn thỉnh thoảng đèn sáng.

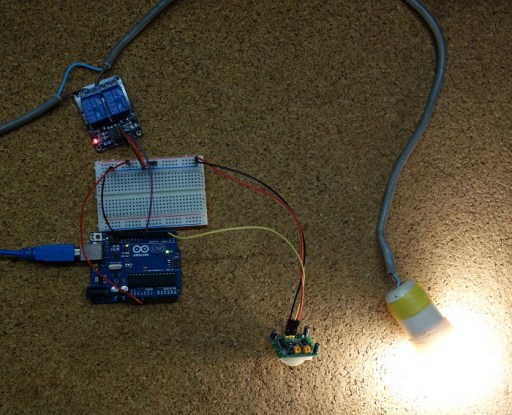
# ****Nối dây****

Các kết nối giữa mô-đun role và Arduino thực sự đơn giản:



* **GND**: nối đất
* **IN1**: điều khiển rơle đầu tiên (nó sẽ được kết nối với chân kỹ thuật số Arduino)
* **IN2**: điều khiển rơle thứ hai (cần được kết nối với chân kỹ thuật số Arduino nếu bạn đang sử dụng rơle thứ hai này. Nếu không, bạn không cần nối nó)
* **VCC**: đi đến 5V

# ****Ví dụ: Điều khiển đèn bằng Mô-đun Rơle và Cảm biến chuyển động PIR****



Trong ví dụ này, bạn sẽ tạo ra đèn sáng trong 10 giây mỗi khi phát hiện chuyển động.

Chuyển động sẽ được phát hiện bằng cảm biến chuyển động PIR.

***Cảnh báo an toàn***

Trước khi tiếp tục với dự án này, tôi muốn cho bạn biết rằng bạn đang xử lý điện áp 200V nên cần hết sức cẩn thận.

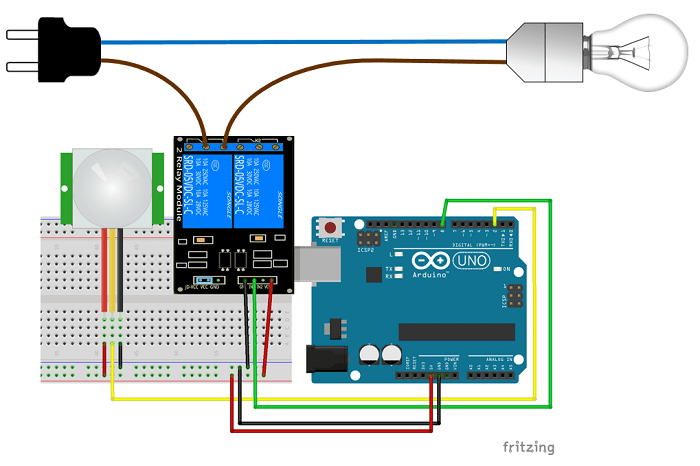
***Linh kiện cần thiết***

Đây là những linh kiện cần thiết cho ví dụ này:

* Mô-đun rơle
* Arduino UNO
* Cảm biến chuyển động PIR
* Bộ dây đèn

***Sơ đồ***

Lắp ráp tất cả các linh kiện như trong sơ đồ dưới đây.



Lưu ý: không chạm vào bất kỳ dây nào được kết nối với điện áp 220V. Hãy chắc chắn rằng bạn đã thắt chặt tất cả các ốc vít.

# Code

// Relay pin is controlled with D8. The active wire is connected to Normally Closed and common

int relay = 8;

volatile byte relayState = LOW;

// PIR Motion Sensor is connected to D2.

int PIRInterrupt = 2;

// Timer Variables

long lastDebounceTime = 0;

long debounceDelay = 10000;

void setup() {

  // Pin for relay module set as output

  pinMode(relay, OUTPUT);

  digitalWrite(relay, HIGH);

  // PIR motion sensor set as an input

  pinMode(PIRInterrupt, INPUT);

  // Triggers detectMotion function on rising mode to turn the relay on, if the condition is met

  attachInterrupt(digitalPinToInterrupt(PIRInterrupt), detectMotion, RISING);

  // Serial communication for debugging purposes

  Serial.begin(9600);

}

void loop() {

  // If 10 seconds have passed, the relay is turned off

  if((millis() - lastDebounceTime) > debounceDelay && relayState == HIGH){

    digitalWrite(relay, HIGH);

    relayState = LOW;

    Serial.println("OFF");

  }

  delay(50);

}

void detectMotion() {

  Serial.println("Motion");

  if(relayState == LOW){

    digitalWrite(relay, LOW);

  }

  relayState = HIGH;

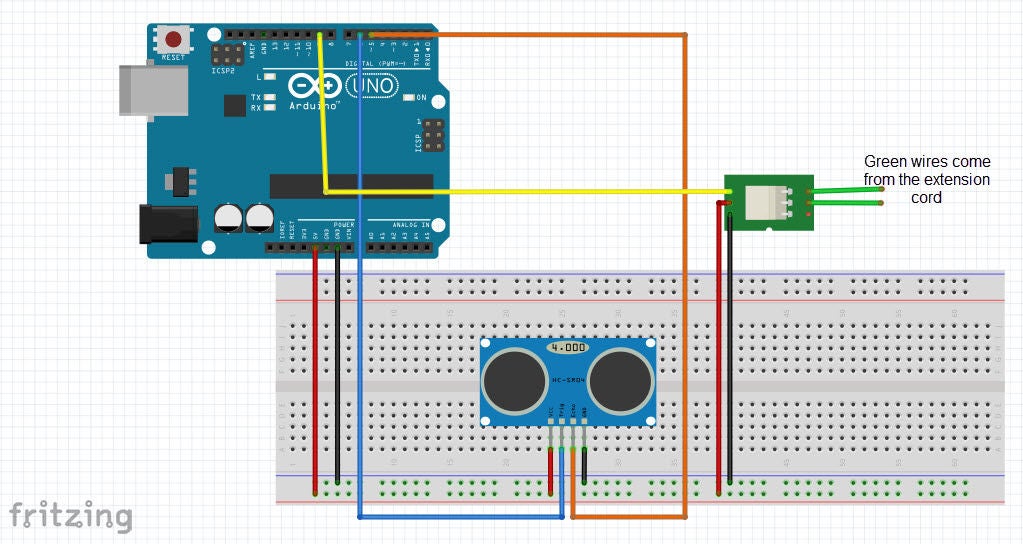
  Serial.println("ON");

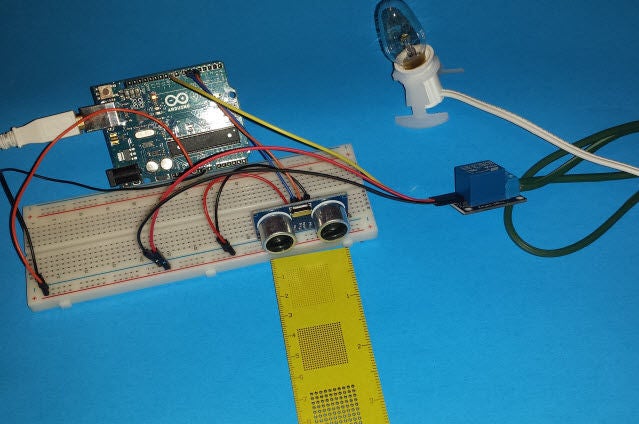
  lastDebounceTime = millis();

}

# ****Ví dụ: Điều khiển đèn bằng Mô-đun Rơle và Cảm biến khoảng cách**** HC-SR04 Sensor

# Nếu khoảng cách đến vật cản nhỏ hơn 50 CM thì đèn sẽ sáng





**CODE**

// Use the HC-SR04 to detect movement after 13 CM to turn on a light for 15 seconds

#define trigPin 6 //Define the HC-SE04 triger on pin 6 on the arduino

#define echoPin 5 //Define the HC-SE04 echo on pin 5 on the arduino

#define bulb 9 //Define the relay signal on pin 9 on the arduino

void setup()

{

Serial.begin (9600); //Start the serial monitor

pinMode(trigPin, OUTPUT); //set the trigpin to output

pinMode(echoPin, INPUT); //set the echopin to input

pinMode (bulb, OUTPUT); //set the bulb on pin 9 to output

}

void loop()

{

int duration, distance; //Define two intregers duration and distance to be used to save data

digitalWrite(trigPin, HIGH); //write a digital high to the trigpin to send out the pulse

delayMicroseconds(500); //wait half a millisecond

digitalWrite(trigPin, LOW); //turn off the trigpin

duration = pulseIn(echoPin, HIGH); //measure the time using pulsein when the echo receives a signal set it to high

distance = (duration/2) / 29.1; //distance is the duration divided by 2 becasue the signal traveled from the trigpin then back to the echo pin, then devide by 29.1 to convert to centimeters

if (distance < 50) //if the distance is less than 50 CM

{

Light(); //execute the Light subroutine below

}

Serial.print(distance); //Dispaly the distance on the serial monitor

Serial.println(" CM"); //in centimeters

delay(500); //delay half a second

}

void Light() //Start the Light subroutine

{

digitalWrite(bulb, HIGH); //turn on the light

delay (15000); //wait 15 seconds

digitalWrite(bulb, LOW); //turn off the light

}