

# GIAO TIẾP CẢM BIẾN - VI ĐIỀU KHIỂN - THIẾT BỊ CHẤP HÀNH

## 1. Thực hành cho vi điều khiển thu thập dữ liệu từ cảm biến nhiệt độ, độ ẩm và hiển thị kết quả ra màn hình

### 1.1. Cảm biến nhiệt độ, độ ẩm (DHT)

- Cảm biến độ ẩm và nhiệt độ DHT11 là cảm biến rất thông dụng và chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (dùng 1 chân digital để truyền dữ liệu. Bộ tiền xử lý tín hiệu tích hợp trong cảm biến có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.
- Cảm biến DHT11 được tích hợp trong một mạch duy nhất. Chỉ cần nối dây nguồn (VCC), dây đất (GND) và dây tín hiệu (Data, Signal) là xong.



### 1.2. Thông số kỹ thuật của DHT11:

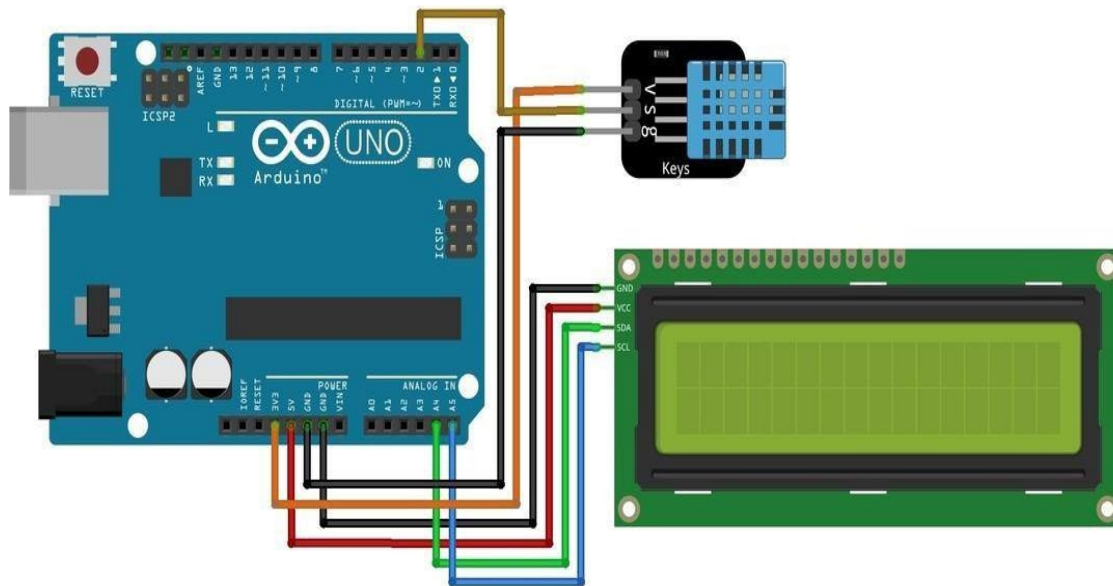
- Điện áp hoạt động: 3-5.5V DC
- Ngưỡng độ ẩm: 20 - 90%
- Sai số độ ẩm:  $\pm 5\%$  RH
- Ngưỡng nhiệt độ: 0 - 55°C
- Sai số nhiệt độ:  $\pm 2^\circ\text{C}$

Ngoài ra còn có cảm biến nhiệt độ, ẩm dòng DHT21, DHT 22 giúp cho việc sai số khi đo đáng kể hơn, phạm vi đo cũng mở rộng thêm

### 1.3. Chuẩn bị phần cứng:

- Cảm biến nhiệt độ, độ ẩm DHT11
- Board Arduino Uno
- Bread Board
- Màn hình LCD 16x2 giao tiếp I2C

### 1.4. Sơ đồ đấu nối:



Arduino	LCD
5V	VCC
GND	GND
SDA/A4	SDA
SCL/A5	SCL

Arduino	DHT11
5V	VCC
GND	GND
D7	Signal (Data)

## 2.5. Code tham khảo:

- Thư viện DHT11: “DHT sensor library” của Adafruit, link tham khảo code hoặc có thể tải qua: <https://github.com/adafruit/DHT-sensor-library>

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
#include "DHT.h" //khai báo thư viện DHT
```

```
const int DHTPIN = 7; //khai báo chân dữ liệu DHT
const int DHTTYPE = DHT22; //khai báo kiểu DHT, có 3 loại DHT11,
DHT21, DHT22 tùy kết quả có thể thay loại
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
  Serial.begin(9600);
  dht.begin(); // Khởi động cảm biến
  lcd.init(); // Khởi tạo LCD
  lcd.backlight(); // Bật đèn LCD
}
```

```
void loop() {
```

```

float h = dht.readHumidity(); //Đọc độ ẩm
float t = dht.readTemperature(); //Đọc nhiệt độ
lcd.setCursor(0,0); // Đặt con trỏ hàng 1 cột 1
lcd.print("Do Am:"); // In ra chữ Độ ẩm
lcd.print(h); // In ra giá trị Độ ẩm
lcd.setCursor(0,1); // Đặt con trỏ hàng 2 cột 1
lcd.print("Nhiệt Độ:"); // In ra chữ Nhiệt độ
lcd.print(t); // In ra giá trị Nhiệt độ
delay(1000); //Đợi 1 giây
}

```

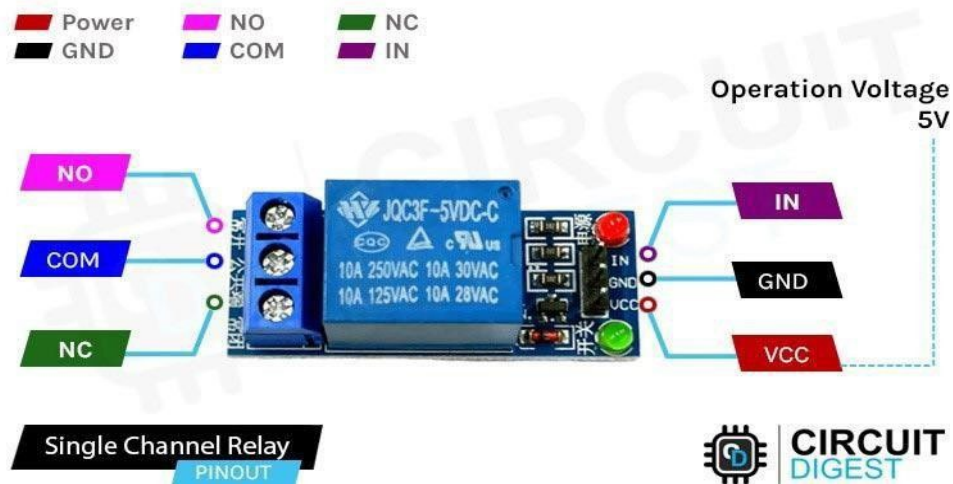
## 2.6. Kết quả:



## 2. Thực hành cho vi điều khiển điều khiển bóng đèn qua relay

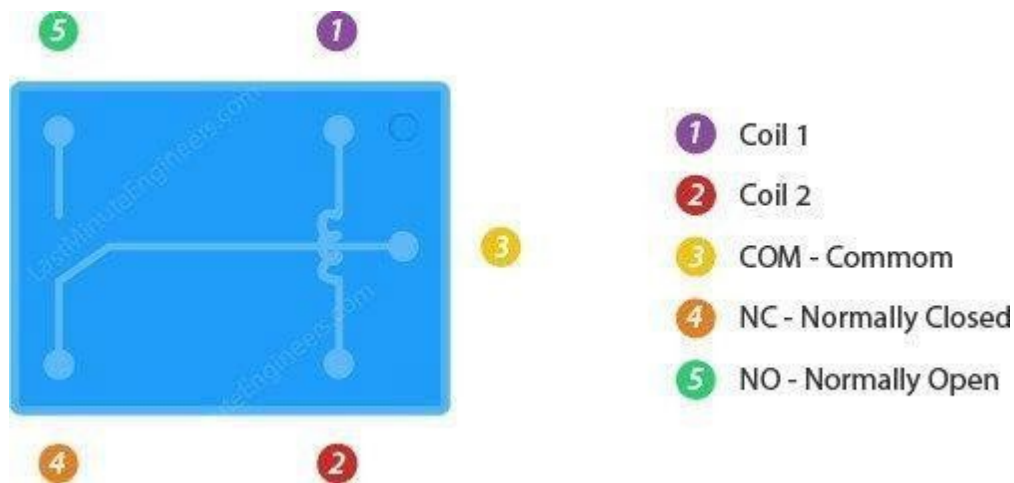
### 2.1. Relay (Rơ-le)

- Relay là một thiết bị điện tử dùng để điều khiển mạch điện bằng cách sử dụng một tín hiệu điều khiển nhỏ. Nó hoạt động dựa trên nguyên lý của cuộn dây điện dẫn điện và cơ cấu chuyển mạch.
- Relay có hai trạng thái chính: trạng thái thường đóng (NC) và trạng thái thường mở (NO). Khi một tín hiệu điều khiển được đưa vào cuộn dây điện, relay sẽ chuyển từ trạng thái thường mở (NO) sang trạng thái thường đóng (NC) giúp cho thiết bị được hoạt động.

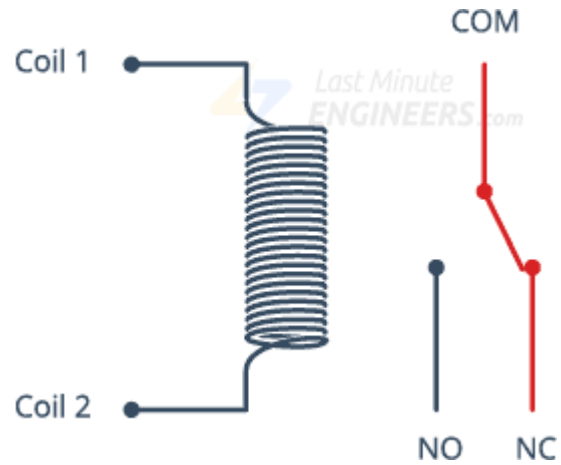


## 2.2. Nguyên lý hoạt động

Khi một tín hiệu điều khiển được cấp vào cuộn dây điện, dòng điện chạy qua cuộn dây sẽ tạo ra một trường từ xung quanh cuộn dây. Trường từ này sẽ tác động lên cơ cấu chuyển mạch trong relay, làm cho nó chuyển từ trạng thái thường mở (NO) sang trạng thái thường đóng (NC).



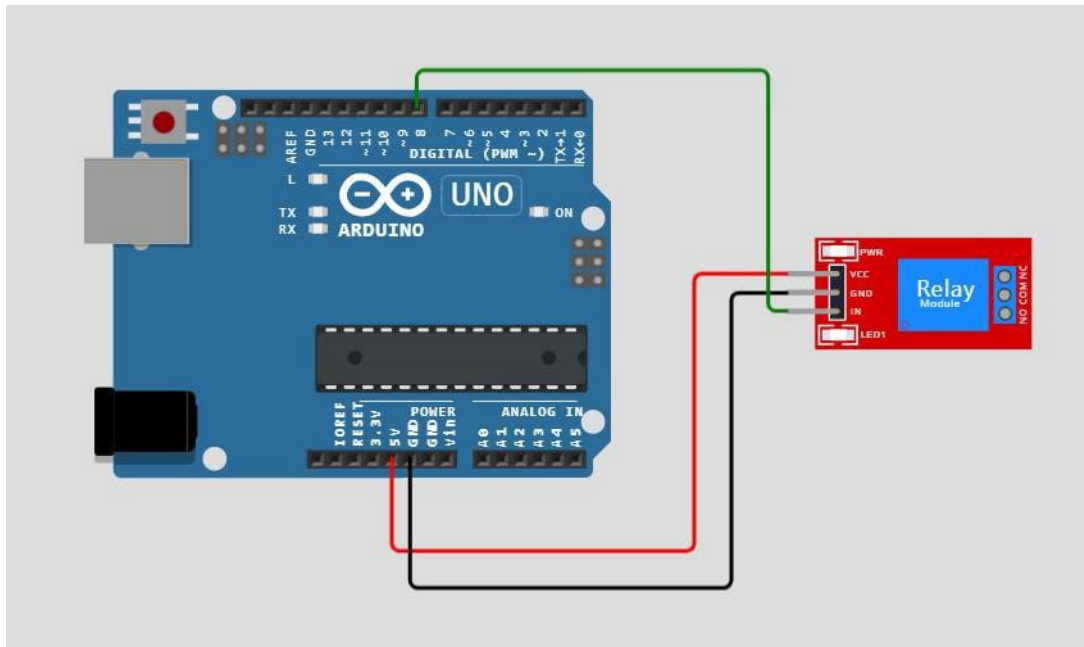
- Khi rơ le ở trạng thái thường mở (NO) thì các tiếp điểm trong mạch relay không được kết nối với nhau. Khi tín hiệu điều khiển được cấp, relay chuyển sang trạng thái thường đóng (NC)



### 2.3. Chuẩn bị phần cứng:

- 1 Relay
- 1 nguồn ngoài 12V
- Bóng đèn 12V
- Arduino Uno

### 2.4. Sơ đồ đấu nối:



Arduino	Relay
5V	VCC
GND	GND
D8	IN

## 2.5. Code tham khảo:

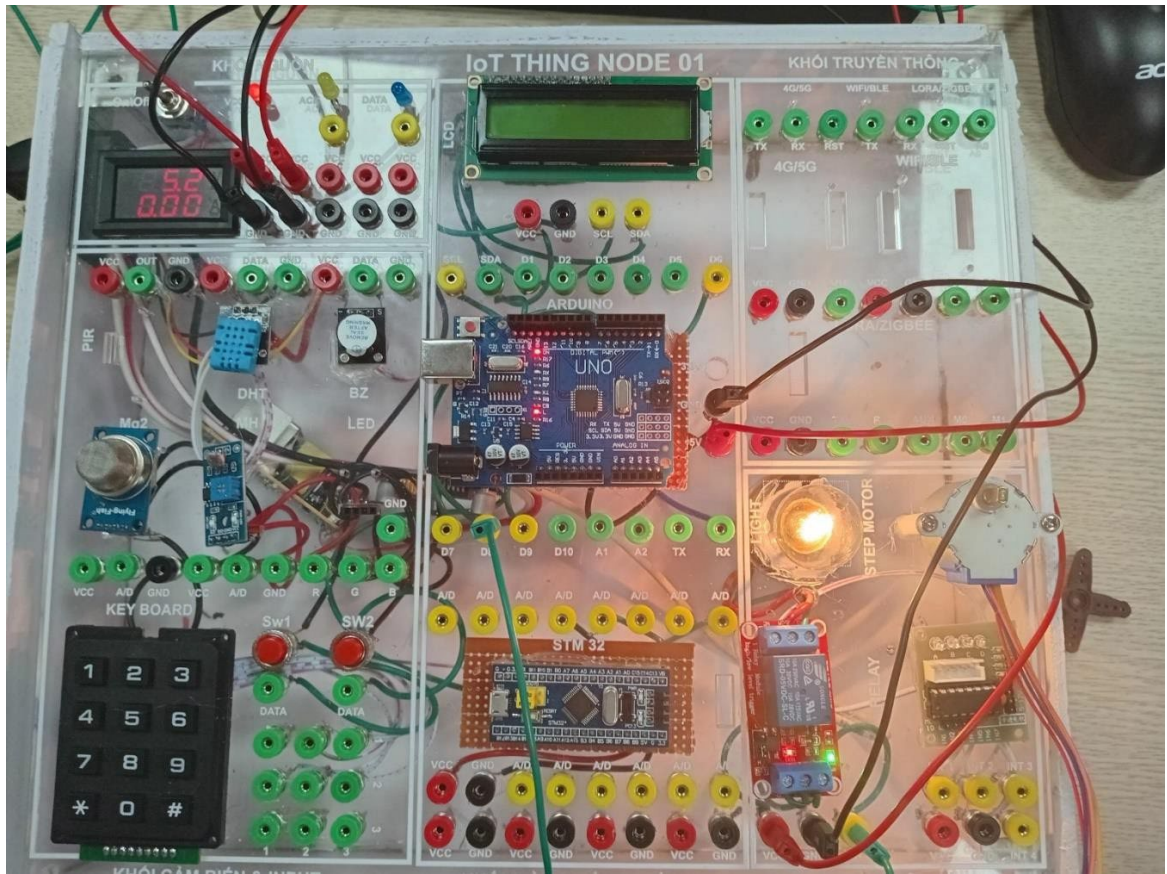
```

int Relay = 8;
void setup() {
  pinMode(Relay, OUTPUT);
  digitalWrite(Relay, HIGH);
}
void loop() {
  digitalWrite(Relay,
LOW); delay(1000);
  digitalWrite(Relay,
HIGH); delay(1000);
}

```

## 2.6. Kết quả:





### 3. Thực hành cho vi điều khiển điều khiển động cơ bước

#### 3.1. Động cơ bước (Step Motor)

- Động cơ bước là một loại động cơ mà ở đó bạn sẽ có thể quy định chính xác số góc quay và động cơ bước sẽ phải quay. Động cơ bước có thể quay bao nhiêu độ tùy ý và mỗi lần quay nó sẽ quay được 1 step, 1 step ở đây là bao nhiêu còn phụ thuộc vào động cơ bước của bạn.

Động cơ bước 28BYJ-48

- Động cơ bước sử dụng trong bài toán là động cơ bước 4 pha (thực ra là 2 pha được chia ra làm 2 ở mỗi pha ngay tại vị trí giữa) (gồm 5 dây), 4 trong 5 dây này được kết nối với 2 cuộn dây trong động cơ và 1 dây là dây nguồn chung cho cả 2 cuộn dây. Mỗi bước của động cơ quét 1 góc 5.625 độ, vậy để quay 1 vòng động cơ phải thực hiện 64 bước.





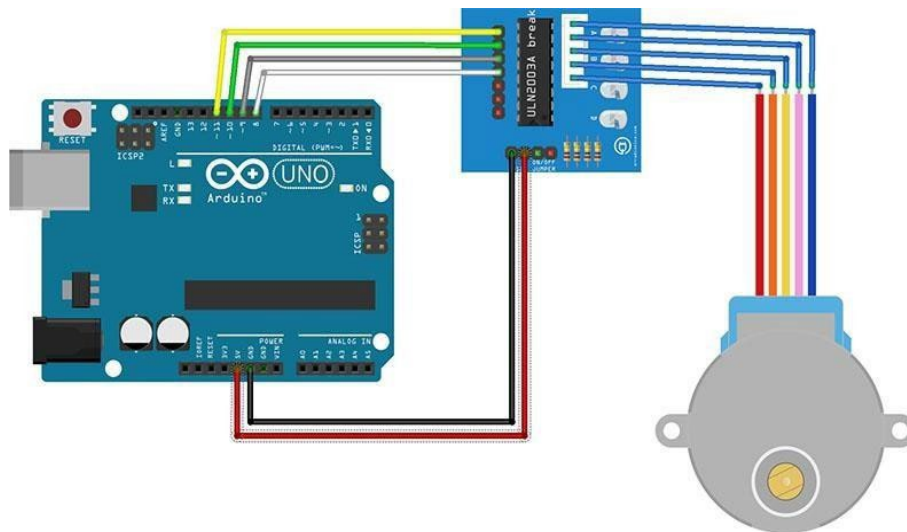
### 3.2. Thông số kỹ thuật

- Điện thế hoạt động: 5V
- Số pha: 4
- Tỷ lệ bánh răng: \*64
- Một bước tương đương:  $5.625^\circ$  (64 bước)
- Tần số: 100Hz
- Điện trở trong:  $50\Omega \pm 7\%$  ( $25^\circ\text{C}$ )

### 3.3. Chuẩn bị phần cứng:

- Arduino Uno
- Động cơ bước 28BYJ-48

### 3.4. Sơ đồ đấu nối:



Arduino	Động cơ
5V	VCC
GND	GND
D6	INT1
D7	INT2
D8	INT3
D9	INT4

### 3.5. Code tham khảo:

- Thư viện cho động cơ bước: “Stepper” của chính hãng Arduino thông tin thư viện chi tiết tại link:  
<https://www.arduino.cc/reference/en/libraries/stepper/>
- Động cơ **BYJ48 Stepper** chỉ có sẵn bộ hộp số và nó cho ta đến \*64 (tỷ lệ

bánh răng) => nó có đến  $64 * 64 = 4096$  bước

```
#include <Stepper.h>
```

```
// Khai báo số bước và chân kết nối đến driver động cơ bước
```

```
const int stepsPerRevolution = 200;
```

```
Stepper stepper(stepsPerRevolution, 6, 7, 8, 9);
```

```
void setup() {
```

```
    // Thiết lập tốc độ quay của động cơ (điều chỉnh theo yêu cầu)
```

```
    stepper.setSpeed(60);
```

```
}
```

```
void loop() {
```

```
    // Quay theo chiều kim đồng hồ 1 vòng
```

```
    stepper.step(stepsPerRevolution);
```

```
    delay(1000); // Dừng 1 giây
```

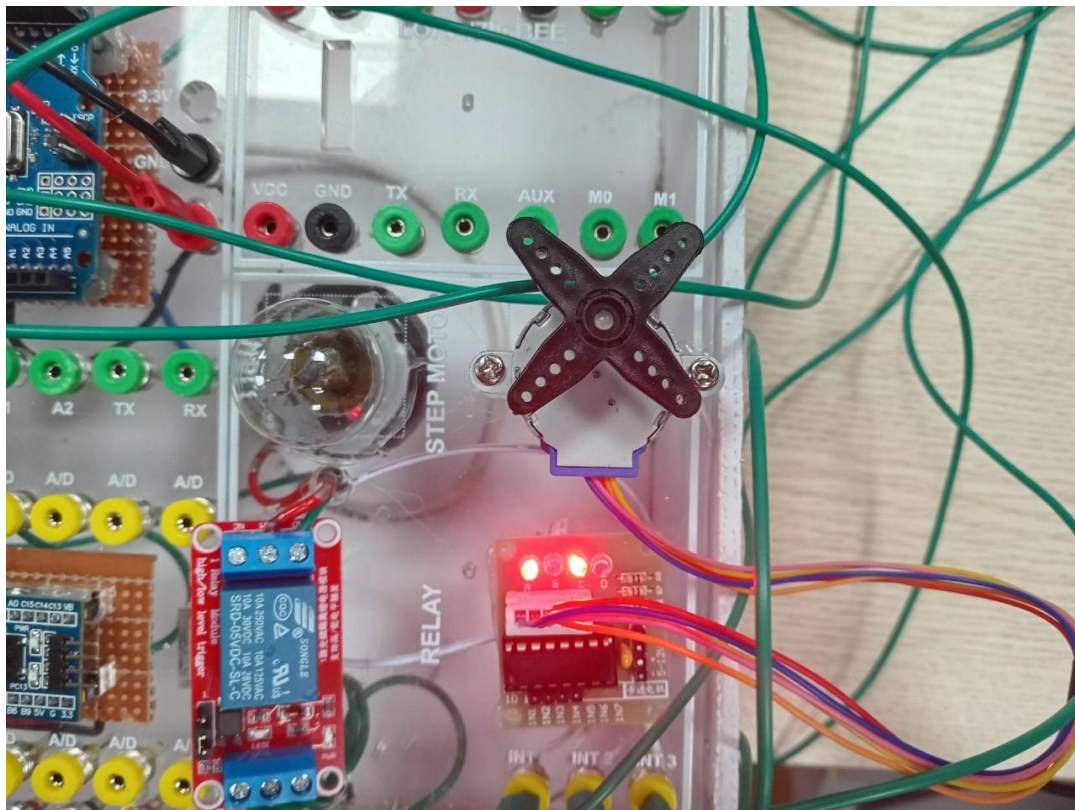
```
    // Quay ngược chiều kim đồng hồ 1 vòng
```

```
    stepper.step(-stepsPerRevolution);
```

```
    delay(1000); // Dừng 1 giây
```

```
}
```

### 3.5. Kết quả:



## 4. Cảm biến khí gas

### 4.1. Định nghĩa:

- Cảm biến khí Gas MQ2 là một trong những cảm biến được sử dụng rộng rãi nhất trong các dòng cảm biến MQ. Nó là một cảm biến MOS (Metal Oxide Semiconductor). Cảm biến oxit kim loại hay còn được gọi là (**Điện trở hóa trị**) vì cảm biến dựa trên sự thay đổi điện trở của cảm biến khi tiếp xúc với khí.
- Cảm biến khí gas arduino hoạt động trên 5V DC và tiêu thụ khoảng 800mW. Nó có thể phát hiện nồng độ **LPG, Khí, Rượu, Propane, Hydrogen, Methane** và **Carbon Monoxide** từ 200 đến 10000 ppm
- Lưu ý: **Cảm biến MQ2** phát hiện nhiều loại khí, nhưng không thể xác định chúng đó là loại khí nào! hầu hết các cảm biến khí trên thị trường hiện nay đều hoạt động theo cách này. Do đó, nó phù hợp để đo hoặc phát hiện nồng độ khí.



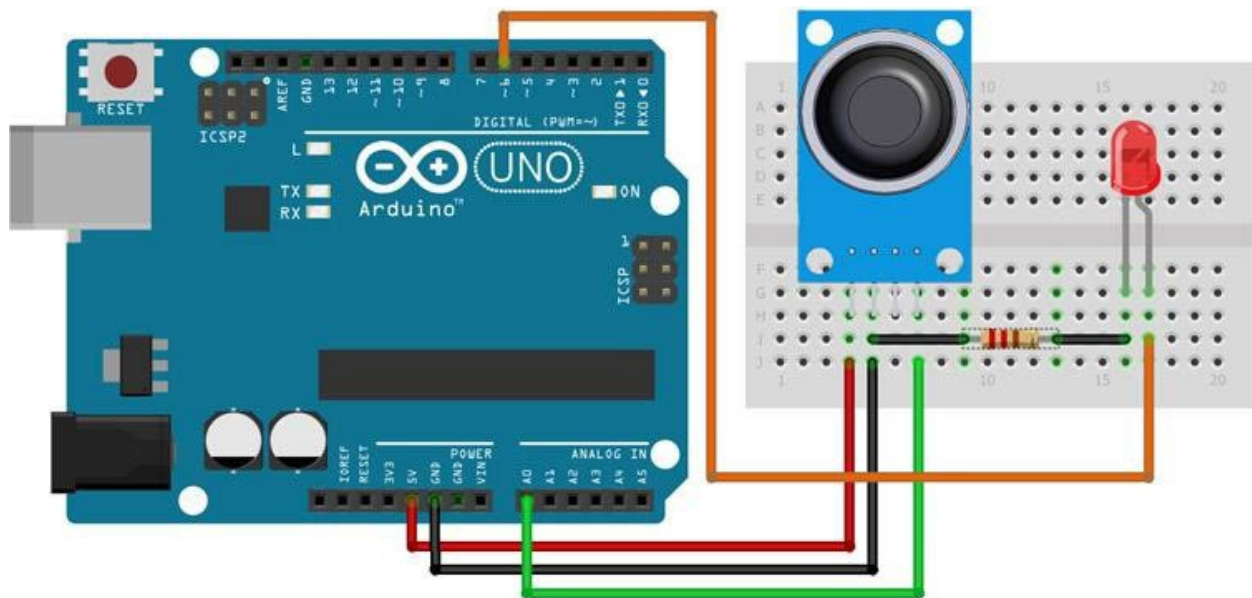
### 4.2. Nguyên lý hoạt động:

- Cảm biến khí Gas MQ2 hoạt động dựa trên nguyên lý phản ứng hóa học khi tiếp xúc các khí trong môi trường. Nguyên tắc hoạt động của nó là khi các khí trong môi trường như khí CO, khí LPG, khí methane, khí Hydro, khói... tiếp xúc với phần tử bên trong cảm biến, làm cho các electron được giải phóng vào Thiếc Dioxide cho phép dòng điện chạy qua cảm biến một cách tự do.
- Khi được làm nóng, phần tử cảm biến sẽ tạo ra phản ứng hóa học với các khí tiếp xúc và làm thay đổi điện trở của phần tử cảm biến. Cảm biến MQ2 đo lường các biến đổi điện trở này và chuyển đổi chúng thành tín hiệu điện analog hoặc Digital.

### 4.3. Thực hành: Sử dụng cảm biến khí gas để báo cháy:

Phần cứng chuẩn bị:

- 1 Cảm biến khí gas MQ02
- 1 Còi báo nếu không có thì có thể thay bằng đèn led để báo hiệu
- Arduino Uno



Chú ý: Không cần phải thêm trở cho cảm biến Gas

Arduino	MQ02 (Gas)
5V	VCC
GND	GND
A0	A0

#### 4.5. Code:

```
#define led 6

void setup() {
  Serial.begin(9600); //Mở cổng Serial để giao tiếp | tham khảo Serial
  pinMode(led,OUTPUT);
}

void loop() {
  int value = analogRead(A0); //đọc giá trị điện áp ở chân A0 - chân cảm biến
  value luôn nằm trong khoảng 0-1023
  Serial.println(value); //xuất ra giá trị vừa đọc
  int threshold = 500 // Set một ngưỡng nào cho khí gas
  if(value > threshold) // nếu vượt quá ngưỡng
    digitalWrite(led,HIGH); // bật đèn cảnh báo
```

```
else // nếu vẫn đang bình thường  
    digitalWrite(led,LOW); //Đèn tắt  
}
```

- Với còi thì cũng y hệt như đèn led, nó cũng có hai chân âm dương và code giống hệt led.