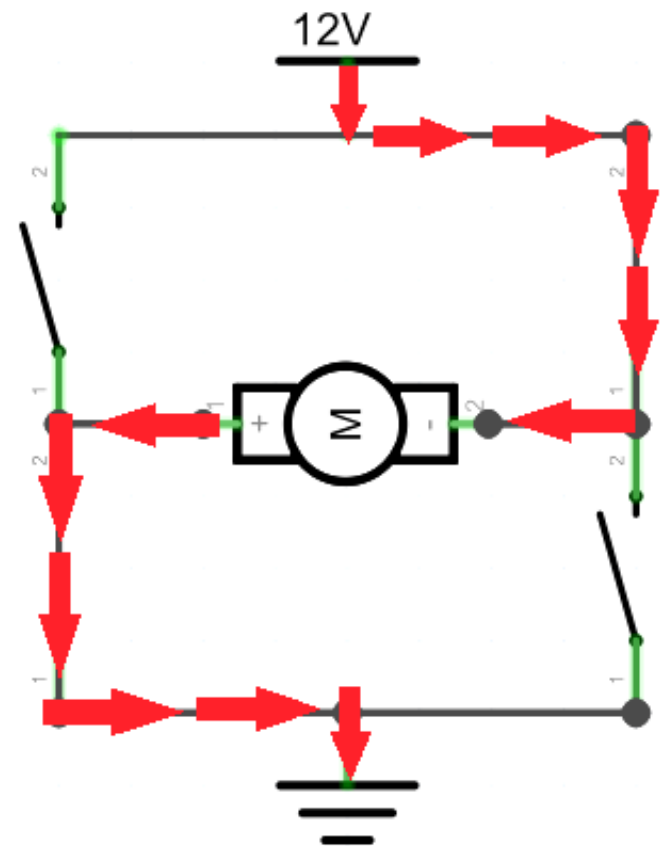
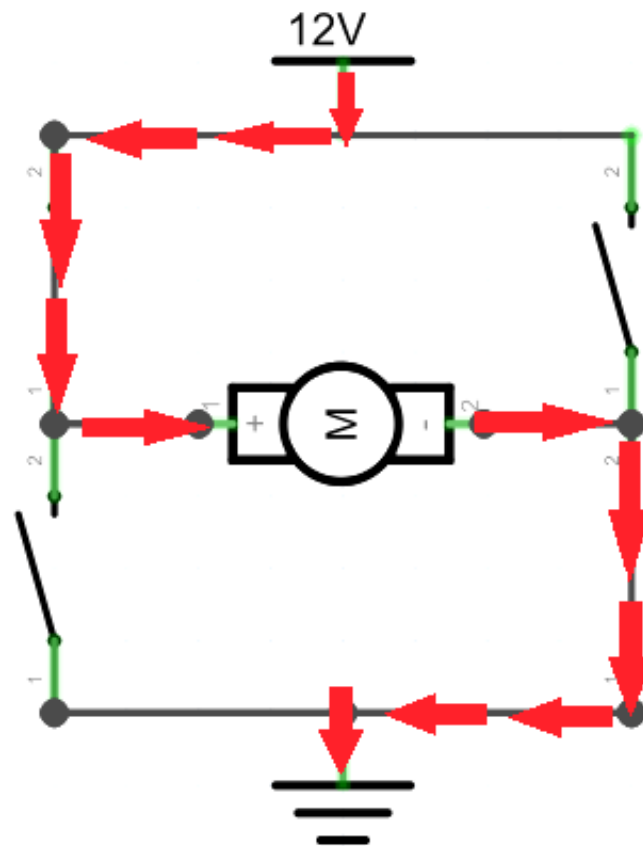
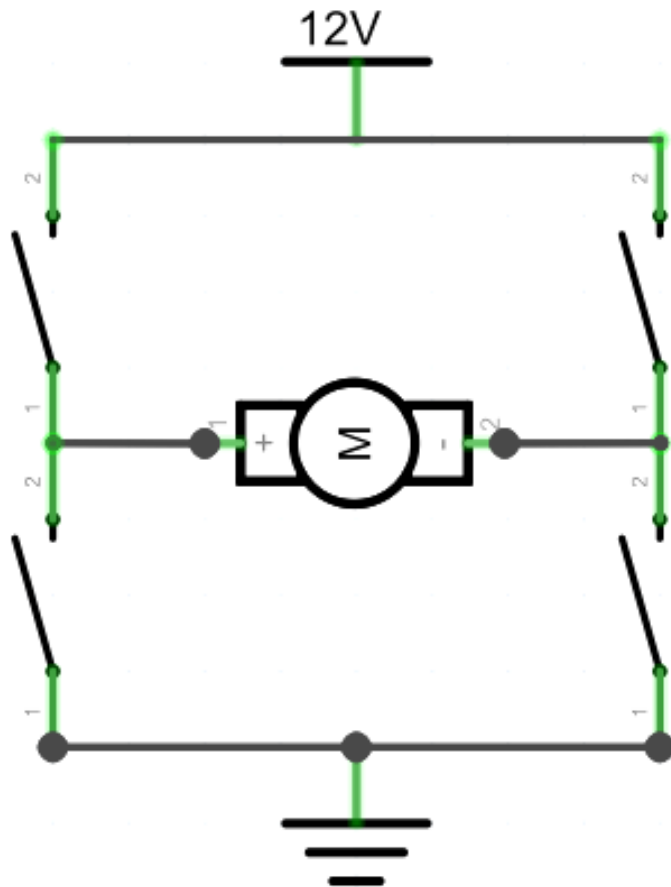




LẬP TRÌNH ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TỬ DÙNG ARDUINO

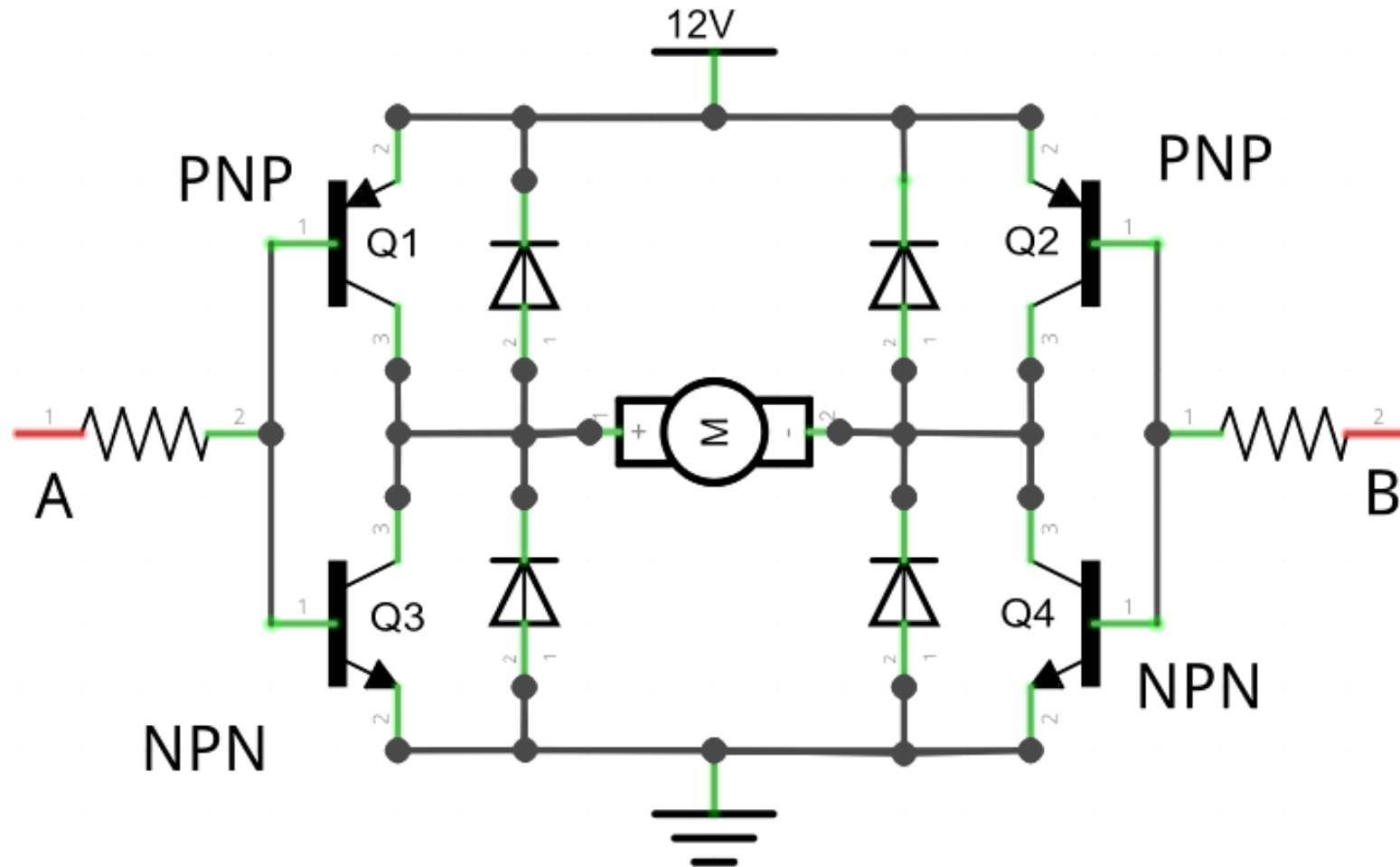
GIỚI THIỆU

Nguyên lý hoạt động Mạch cầu H



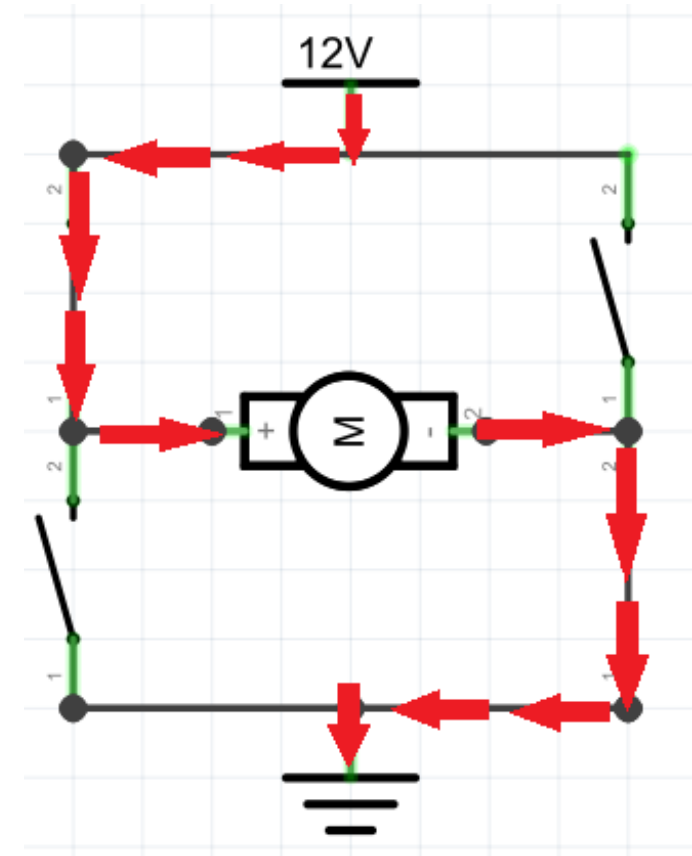
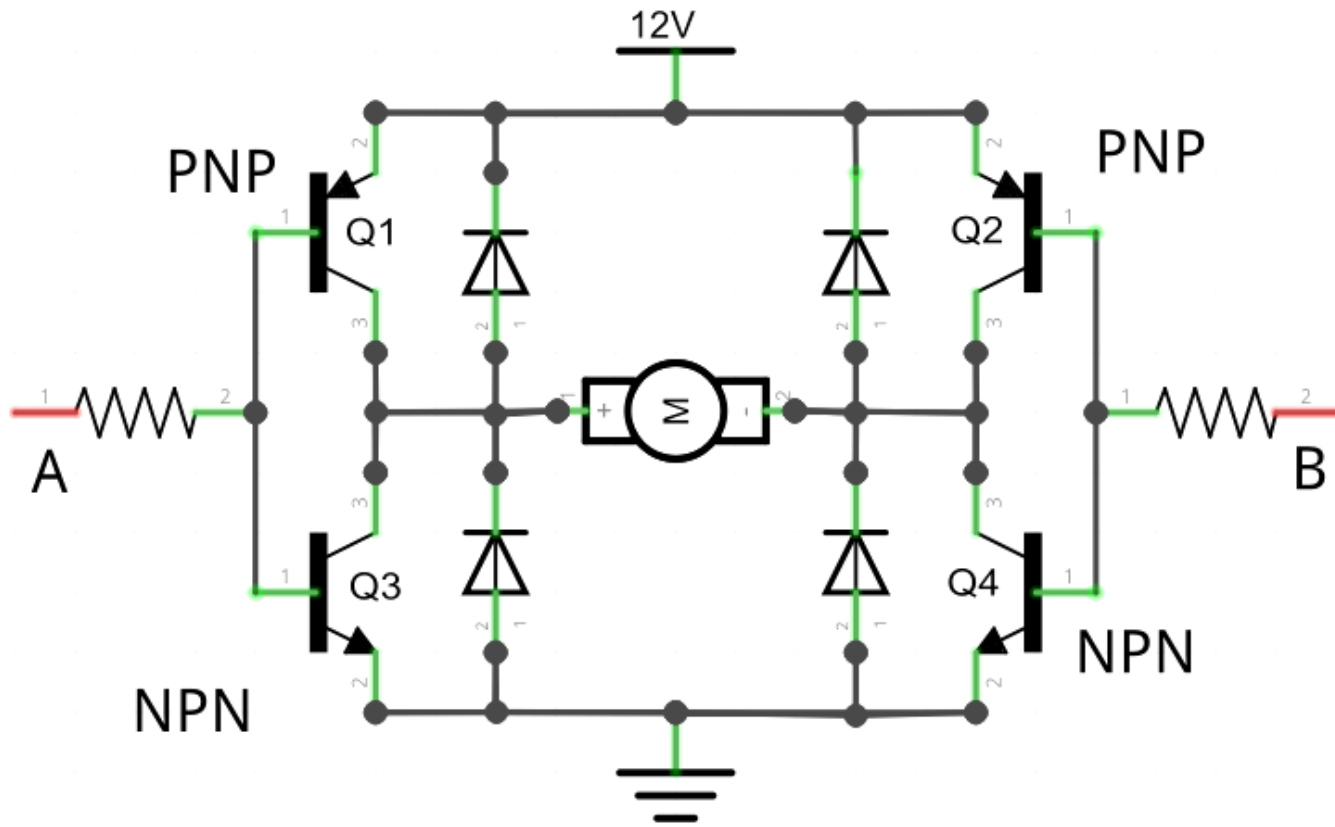
GIỚI THIỆU

Nguyên lý hoạt động Mạch cầu H (transistor BJT)



GIỚI THIỆU

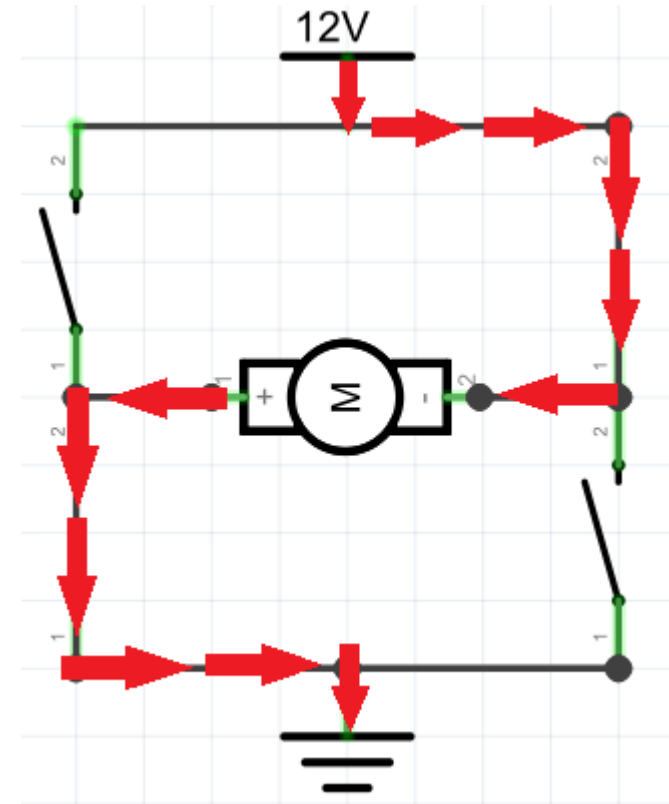
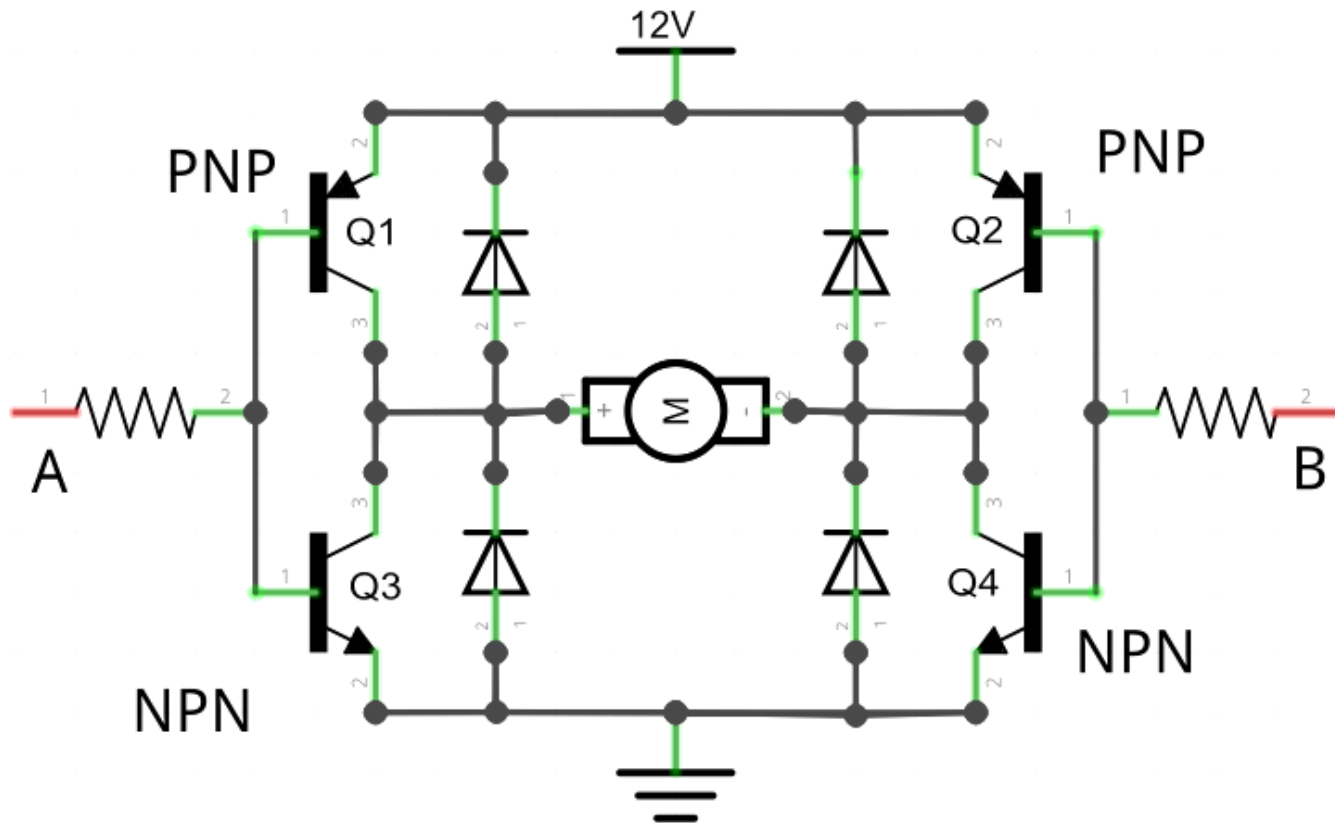
Nguyên lý hoạt động Mạch cầu H (transistor BJT)



A ở mức LOW và B ở mức HIGH

GIỚI THIỆU

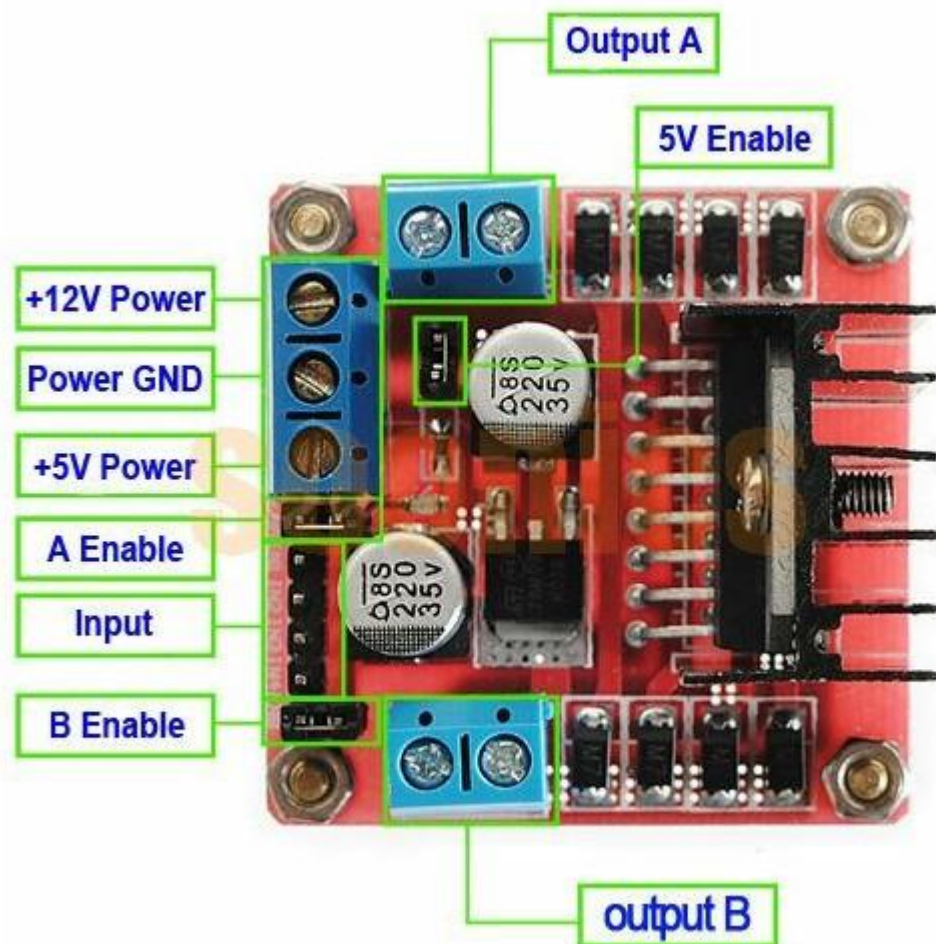
Nguyên lý hoạt động Mạch cầu H (transistor BJT)



A ở mức HIGH và B ở mức LOW

GIỚI THIỆU

Giới thiệu Module điều khiển động cơ L298N



Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển: +5 V ~ +12 V
- Dòng tối đa cho mỗi cầu H là: 2A (\Rightarrow 2A cho mỗi motor)
- Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
- Dòng của tín hiệu điều khiển: 0 ~ 36mA
- Tổn hao công suất: 20W (khi nhiệt độ $T = 75\text{ }^{\circ}\text{C}$)
- Nhiệt độ bảo quản: $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$

Ví dụ 1: Điều khiển động cơ với module L298N

- Động cơ quay thuận trong 5s.
- Động cơ quay nghịch trong 5s.
- Mỗi khi đảo chiều quay của động cơ thì phải dừng 3s.

VÍ DỤ ỨNG DỤNG

Nhập xuất Digital (Digital I/O)

pinMode(): Cấu hình 1 pin quy định hoạt động như là một đầu vào (INPUT) hoặc đầu ra (OUTPUT)

Cú pháp

pinMode(pin, mode)

Thông số

pin: Số của chân digital muốn thiết đặt

mode: INPUT, INPUT_PULLUP hoặc OUTPUT

VÍ DỤ ỨNG DỤNG

Nhập xuất Digital (Digital I/O)

digitalWrite(): Xuất tín hiệu ra các chân digital, có 2 giá trị là HIGH hoặc là LOW

Cú pháp

digitalWrite(pin,value)

Thông số

pin: Số của chân digital mà bạn muốn thiết đặt

value: HIGH hoặc LOW

VÍ DỤ ỨNG DỤNG

Nhập xuất Digital (Digital I/O)

digitalRead(): Đọc tín hiệu điện từ một chân digital (được thiết đặt là INPUT). Trả về 2 giá trị HIGH hoặc LOW.

Cú pháp

digitalRead(pin)

Thông số

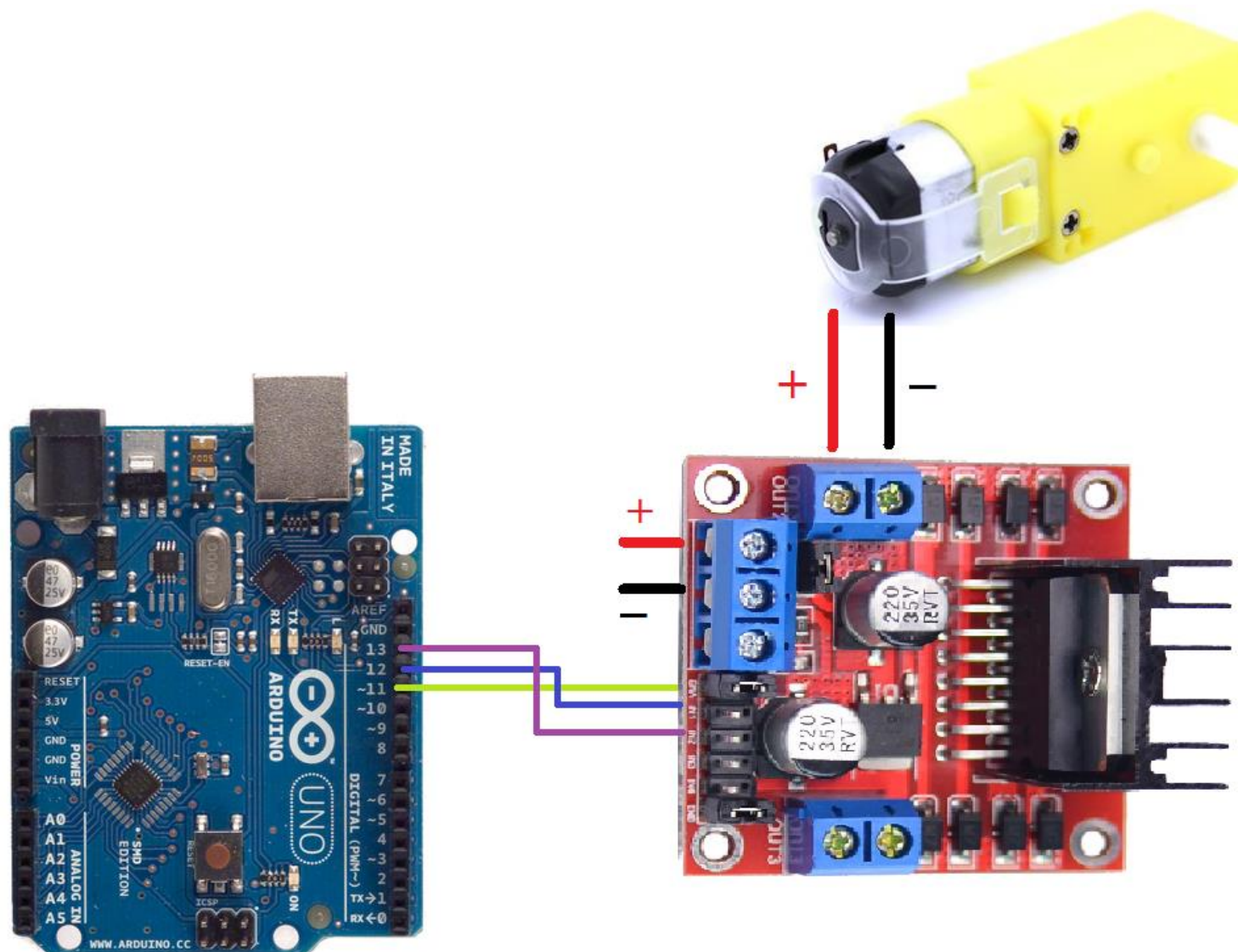
pin: giá trị của digital muốn đọc

Trả về

HIGH hoặc LOW

GIỚI THIỆU

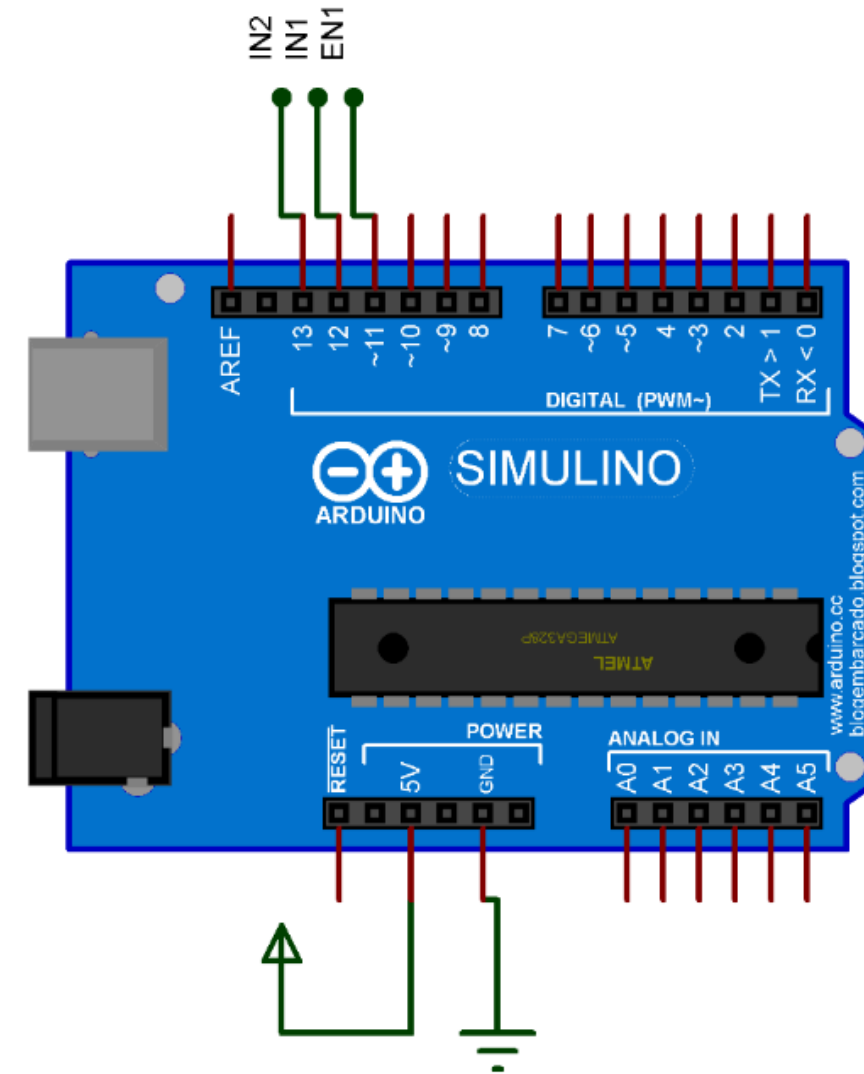
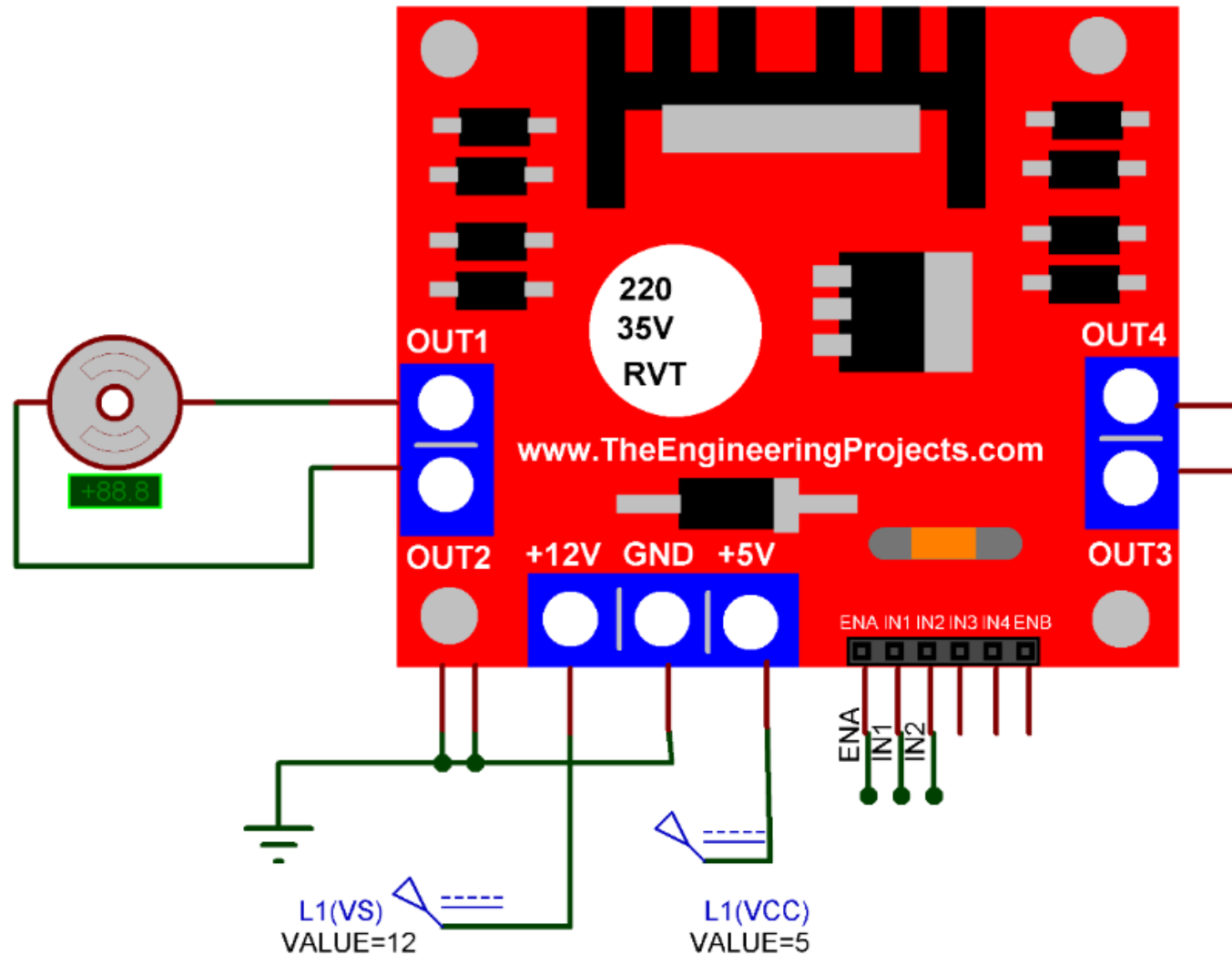
Sơ đồ mạch điện



GIỚI THIỆU

Sơ đồ mạch điện

L1
L298 MOTOR DRIVER



Chương trình điều khiển:

```
#define enA 11 // Chân cho phép
```

```
#define in1 12
```

```
#define in2 13
```

```
void setup() {  
    pinMode(enA, OUTPUT);  
    pinMode(in1, OUTPUT);  
    pinMode(in2, OUTPUT);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
}
```

Chương trình điều khiển:

```
void loop() {  
    // Động cơ Chạy Thuận  
    digitalWrite(enA, HIGH);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    delay(5000);  
    // Động cơ OFF  
    digitalWrite(enA, LOW);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
    delay(3000);
```

```
// Động cơ chạy Ngược  
    digitalWrite(enA, HIGH);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    delay(5000);  
  
    // Động cơ OFF  
    digitalWrite(enA, LOW);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
    delay(3000);}
```

Ví dụ 2: Điều khiển tốc độ động cơ

VÍ DỤ ỨNG DỤNG

Nhập xuất Analog (Analog I/O)

analogRead(): đọc giá trị điện áp từ một chân Analog (ADC). Trả về 1 số nguyên nằm trong khoảng từ 0 đến 1023 tương ứng với thang điện áp (mặc định) từ 0 đến 5V.

Cú pháp

```
analogRead([pin]);
```

Thông số

pin: Chân đọc điện áp

VÍ DỤ ỨNG DỤNG

Nhập xuất Analog (Analog I/O)

analogWrite(): là lệnh xuất ra từ một chân trên mạch Arduino một mức tín hiệu analog (phát xung PWM).

Cú pháp

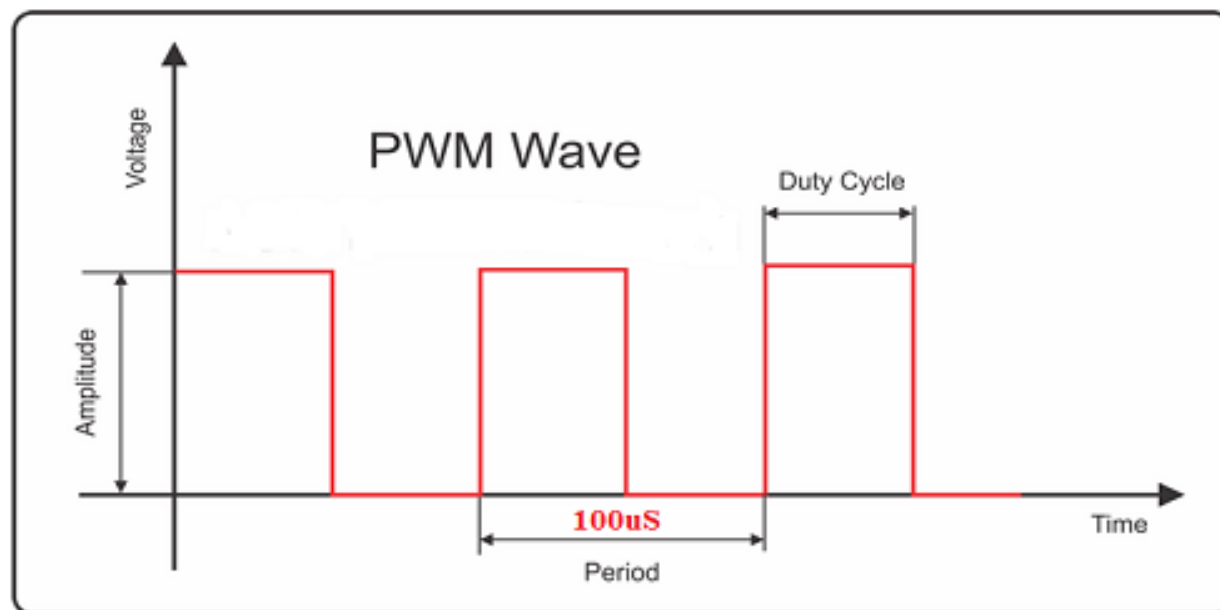
`analogWrite([chân phát xung PWM], [giá trị xung PWM]);`

Giá trị mức xung PWM nằm trong khoảng từ 0 đến 255, tương ứng với mức duty cycle từ 0% đến 100%

VÍ DỤ ỨNG DỤNG

Xung PWM

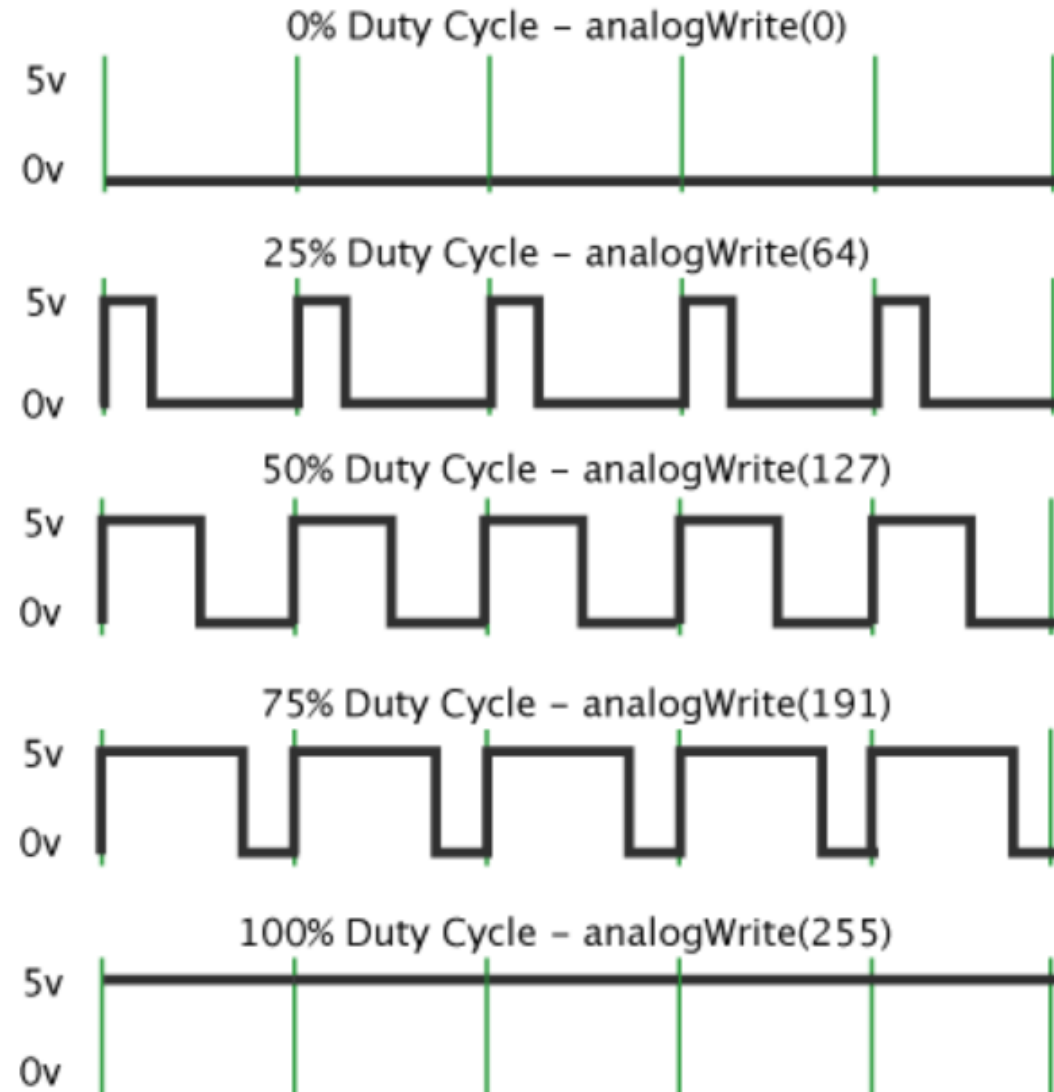
Xung là các trạng thái cao / thấp (HIGH/LOW) về mức điện áp được lặp đi lặp lại.



Đặc trưng cho 1 xung PWM (Pulse Width Modulation) bao gồm tần số (frequency) và chu kỳ xung (duty cycle).

VÍ DỤ ỨNG DỤNG

Xung PWM



Xung khi sử dụng với hàm `analogWrite` trong Arduino

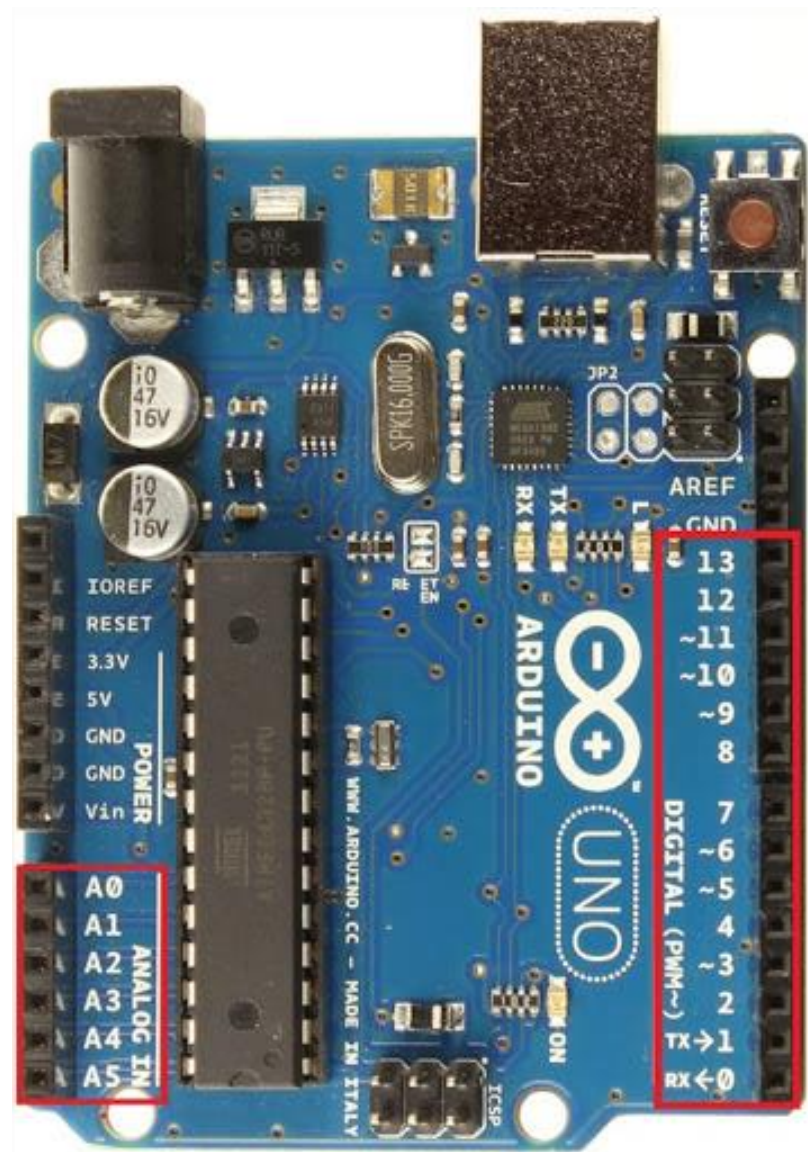
VÍ DỤ ỨNG DỤNG

Xung PWM

analogWrite	tỉ lệ	chu kì xung
analogWrite(0)	0/255	0%
analogWrite(64)	64/255	25%
analogWrite(127)	127/255	50%
analogWrite(191)	191/255	75%
analogWrite(255)	255/255	100%

VÍ DỤ ỨNG DỤNG

Xung PWM



Hàm map

map() là hàm dùng để chuyển một giá trị từ thang đo này sang một giá trị ở thang đo khác. Giá trị trả về của hàm luôn là một số nguyên.

Cú pháp:

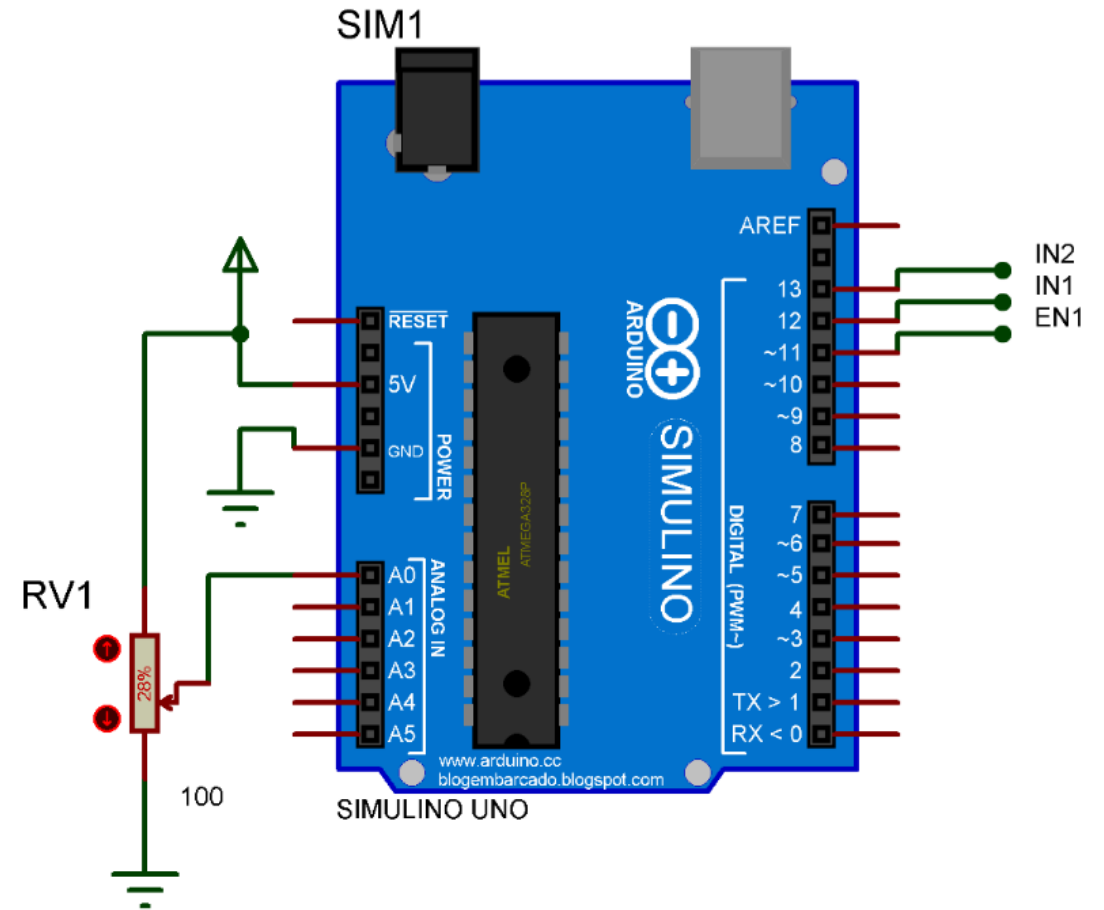
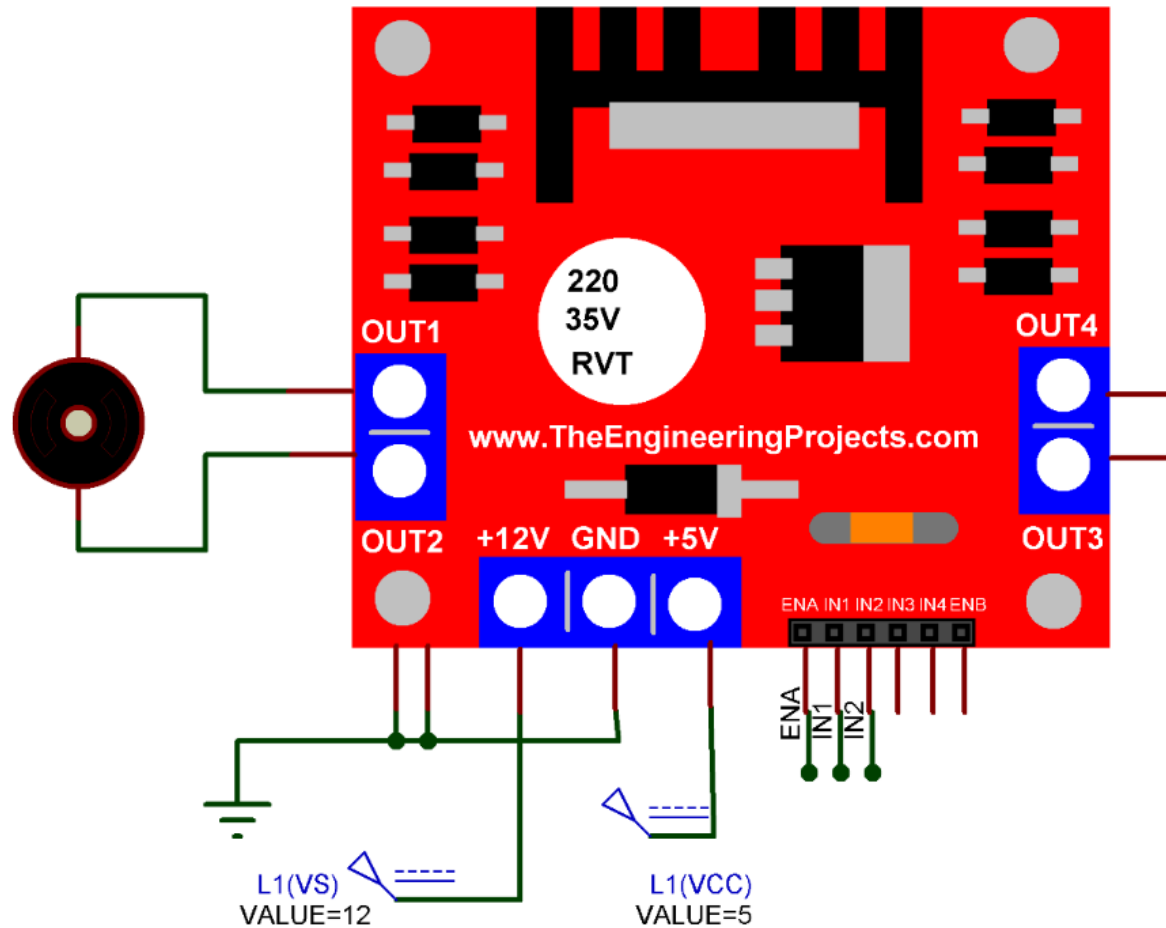
`map(val,A1,A2,B1,B2);`

Trong đó:

- `val` là giá trị cần chuyển đổi
- `A1, A2` là giới hạn trên và dưới của thang đo hiện tại
- `B1,B2` là giới hạn trên và dưới của thang đo cần chuyển tới

GIỚI THIỆU

L1
L298 MOTOR DRIVER



Chương trình điều khiển:

```
#define enA 11 // Chân cho phép
```

```
#define in1 12
```

```
#define in2 13
```

```
void setup() {  
    pinMode(enA, OUTPUT);  
    pinMode(in1, OUTPUT);  
    pinMode(in2, OUTPUT);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
}
```


Chương trình điều khiển:

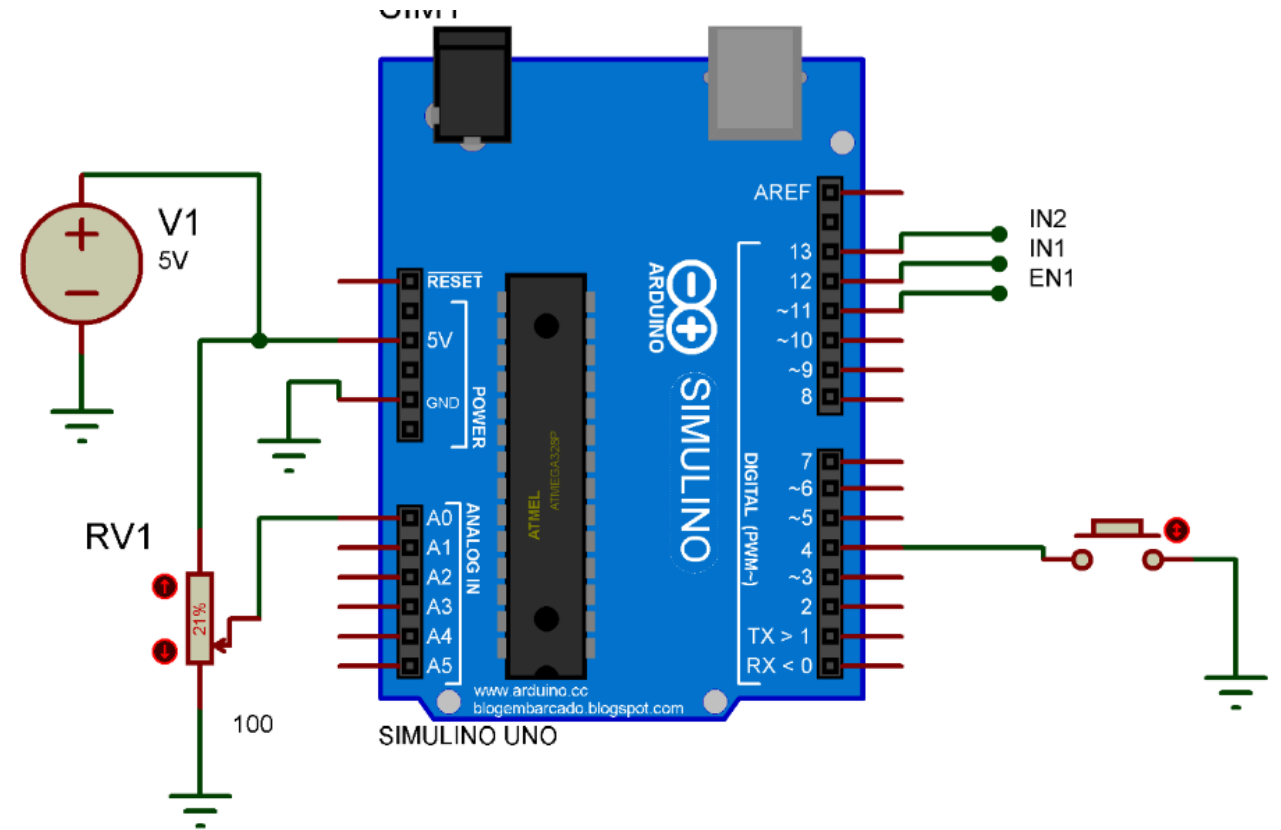
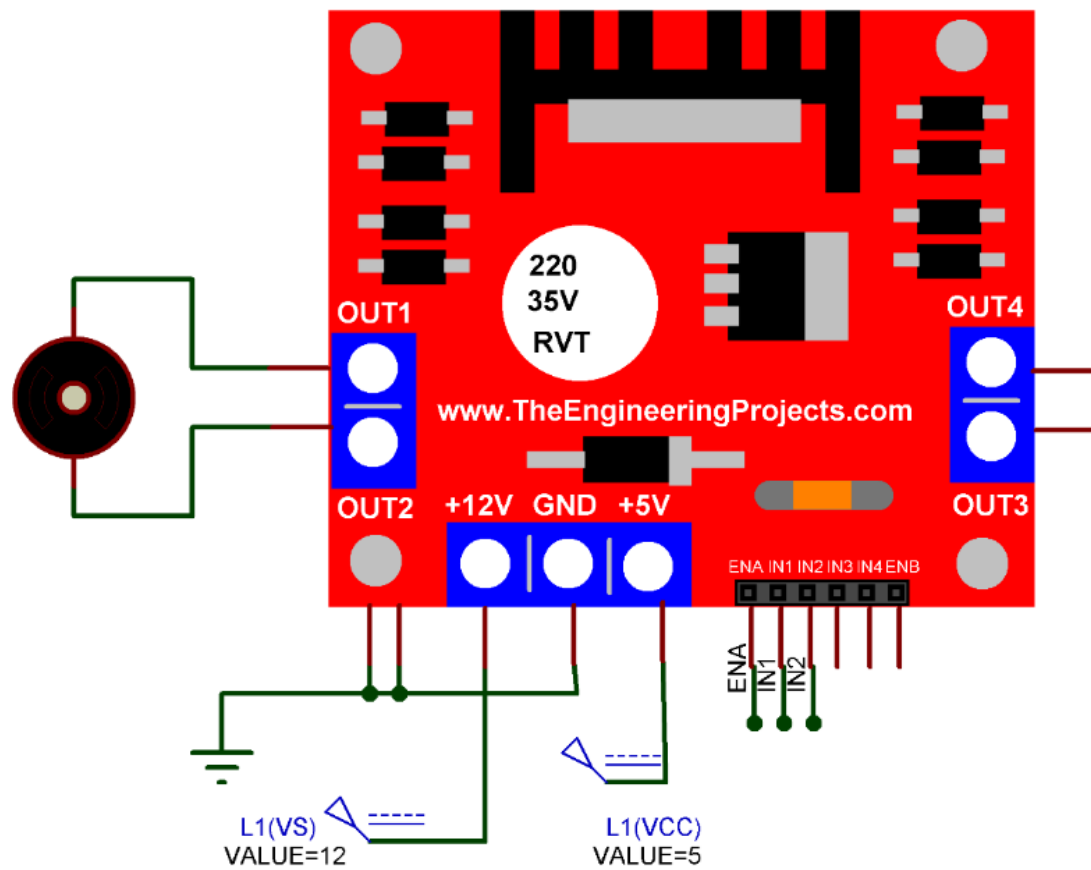
```
void loop() {  
  // Điều chế PWM  
  int potValue = analogRead(A0); // Đọc giá trị từ biến trở  
  int pwmOutput = map(potValue, 0, 1023, 0, 255);  
  analogWrite(enA, pwmOutput); // Xuất xung PWM  
  // Động cơ Chạy Thuận  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, HIGH);  
}
```

Chương trình điều khiển:

```
void loop() {  
  // Điều chế PWM  
  int potValue = analogRead(A0); // Đọc giá trị từ biến trở  
  int pwmOutput = map(potValue, 0, 1023, 0, 255);  
  analogWrite(enA, pwmOutput); // Xuất xung PWM  
  // Động cơ Chạy Thuận  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, HIGH);  
}
```

Ví dụ 3: Điều khiển tốc độ của động cơ bằng biến trở và thay đổi hướng quay bằng nút ấn

GIỚI THIỆU



Chương trình điều khiển:

```
#define enA 11
```

```
#define in1 12
```

```
#define in2 13
```

```
#define button 4
```

```
int rotDirection = 0;
```

```
int pressed = false;
```

```
void setup() {
```

```
    pinMode(enA, OUTPUT);
```

```
    pinMode(in1, OUTPUT);
```

```
    pinMode(in2, OUTPUT);
```

```
    pinMode(button, INPUT);
```

```
    // Set initial rotation direction
```

```
    digitalWrite(in1, LOW);
```

```
    digitalWrite(in2, HIGH);
```

```
}
```

Chương trình điều khiển:

```
void loop() {  
    int potValue = analogRead(A0); // Đọc giá trị từ biến trở  
    int pwmOutput = map(potValue, 0, 1023, 0, 255); // Chuyển đổi giá trị  
    analogWrite(enA, pwmOutput); // Xuất xung PWM tới chân ENA  
  
    // Đọc nút nhấn  
    if (digitalRead(button) == 0) {  
        pressed = !pressed;  
    }  
    while (digitalRead(button) == 0);  
    delay(20);  
}
```

Chương trình điều khiển:

// Nếu nút được nhấn thì thay đổi chiều quay

```
if (pressed == true & rotDirection == 0) {  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    rotDirection = 1;  
    delay(20);  
}  
if (pressed == false & rotDirection == 1) {  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    rotDirection = 0;  
    delay(20);  
}  
}
```

Chương trình điều khiển:

// Nếu nút được nhấn thì thay đổi chiều quay

```
if (pressed == true & rotDirection == 0) {  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    rotDirection = 1;  
    delay(20);  
}
```


*Thank
you*

